Master of Science Thesis in Electrical Engineering
Department of Electrical Engineering, Linköping University, 2023

Gunshot Detection and Direction of Arrival Estimation Using Machine Learning and Received Signal Power

David Grahn & Timothy Cooper

Master of Science Thesis in Electrical Engineering

Gunshot Detection and Direction of Arrival Estimation Using Machine Learning and Received Signal Power

David Grahn & Timothy Cooper

LiTH-ISY-EX–23/5594–SE

Supervisor: **Gustav Zetterqvist**
ISY, Linköping University

Examiner: **Fredrik Gustafsson**
ISY, Linköping University

Division of Automatic Control
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Abstract

Poaching is a persistent issue that threatens many of earth's species including the rhino. The methods used by poachers are varied, but many use guns to carry out their illegal activities. Gunfire is extremely loud and can be heard for kilometres. This thesis investigates whether it is possible to aid anti-poaching efforts in Kenya with a gunshot detection and estimation device using an array of microphones. If successful, the device could be placed around the savannah or any exposed area and warn if poaching is taking place in the nearby. If a shot is fired within the audible range of the device's microphones, a trained machine learning algorithm detects the shot on the edge using a microprocessor. The detection runs in real time and achieved an accuracy of 93% on an unbalanced data set, where the majority class was the one without gunshots. Once a detection has been made, the received signal power to each microphone is used to produce a direction of arrival estimate. The estimate can produce an angle estimate with a standard deviation of 66.78° for a gunshot, and with a standard deviation of 7.65° when testing the model with white noise. Future implementations could use several devices that detected the same event, and fuse their estimates to locate the shooter's position. All of this information, as well as the sound file, can be used to alert and assist local wildlife services. The challenges of this project have been centred around making a system run in real time with only a microprocessor on the edge, while also prioritizing low cost components for future deployment.

Acknowledgments

We want to extend our thanks to Gustav Zetterqvist who served as a supervisor for this project, he has been instrumental with his guidance during the project.

We also want to thank Fredrik Gustafsson who has been the examiner for the project. We want to thank him for the opportunity to partake in project Ngulia it has been a fun challenge and a nice experience.

Carlos Vidal is also a person we want to thank. He has helped us a lot with the hardware aspects of this thesis, without him the hardware would not have been developed to the point it was.

A person who's aid has been instrumental is major Fredrik Perlaky who is the deputy head of education at Markstridsskolan Kvarn. He arranged for us to attend a field exercise with our equipment, allowing us to record many shots being fired. The data he allowed us to gathered served as all the positive data in our training set without which the project would have suffered.

Another person deserving of our thanks is reserve lieutenant Benny Larsson at the marine base in Karlskrona. He allowed us to attend one of his training sessions with our equipment. The data he facilitated served as all the positive data in our testing set as well as directional estimate to real gunshots.

Linköping, June 2023
Timothy Cooper and David Grahn

Contents

Notation

Notations

Notation	Meaning
y_i	Microphone signal for microphone i
s_i	Received signal for microphone i
w_i	Microphone measurement noise for microphone i
l	Measurement time
L	Number of measurements
P_i	Power of signal for microphone i
e_i	Measurement noise power for microphone i
ψ	Angle to sound source
α	Absolute received power
g_i	Gain for microphone i
$h(\psi, \boldsymbol{\theta}_i)$	Microphone directional sensitivity of microphone i
$\boldsymbol{\theta}_i$	Fourier series model parameters for microphone i
K	Number of observed directions

Abbreviations

Abbreviation	Explaination
BIC	Bayesian information criterion
DEU	Detection-estimation unit
DOA	Direction of arrival
DSP	Digital signal processing
DT	Decision Tree
FS	Fourier Series
KNN	K-Nearest Neighbour
LS	Least squares
NB	Naive Bayes
NLS	Non-linear least squares
RSS	Received signal strength
SFS	Sequential forward selection
SVM	Support Vector Machine
STFT	Short Time Fourier Transform
TDOA	Time Delay of Arrival

1 Introduction

This thesis which is part of the larger project Ngulia which aims to ensure the future of wildlife preservation in Africa. This thesis will give aid by detecting and estimating the direction to poachers. This is done by detecting shots fired from poachers though machine learning, and then calculate the direction the sound came from. An alert with the information can be sent to local wildlife protection services, who hopefully can catch the poachers and perhaps even save the animal. This thesis work consists of two large parts: detection and direction of arrival (DOA) estimation. Each part has its own chapter that describes relevant theory and results, with an additional chapter discussing the hardware.

1.1 Background

Poaching has long been a problem for both African elephants and rhinos. African forest elephants and black rhinos are classified as critically endangered [6, 9], while the savannah elephant is classified as endangered [10]. White rhinos have made a major population recovery since the end of the 19th century but are still classified as vulnerable [7]. According to the International Union for Conservation of Nature (IUCN), poaching is the major cause for individual deaths and population decline for all species. While sources disagree on the exact numbers, it is likely that between 10,000 and 35,000 African elephants are killed every year through poaching [1, 8, 25]. The number of rhinos killed annually is around 500 to 1,000 [2].

Poaching of large animals is most often carried out with firearms. Kalashnikov-pattern rifles are commonly used since they are cheap and abundant but are generally not as effective as hunting rifles made for large game. Some poachers will also employ sound suppressors or even forgo firearms entirely to avoid detection.

Tranquilliser darts and poison-tipped arrows have been used where the likelihood of detection by park rangers is high [26].

Downrange from a fired shot, one or two distinct sounds are produced depending on whether the bullet is subsonic (slower than the speed of sound) or supersonic (faster than the speed of sound). The first is the crack of the shock wave if the bullet is supersonic, and second is the the muzzle blast that is created when the bullet leaves the barrel. If the bullet is subsonic, no shock wave is produced. Additionally, if the direction of fire is away from the measuring device the muzzle blast is the only sound that can be observed regardless of the speed of the bullet. A third type of sound is also produced by mechanical workings and residual gas releases from the gun, however these are of much lower amplitude than the previous sounds and are therefore irrelevant to the project. It is important to note that a shock wave and muzzle blast sound may appear to come from slightly different directions [13]. An example of this occurring can be seen illustrated in Figure 1.1.

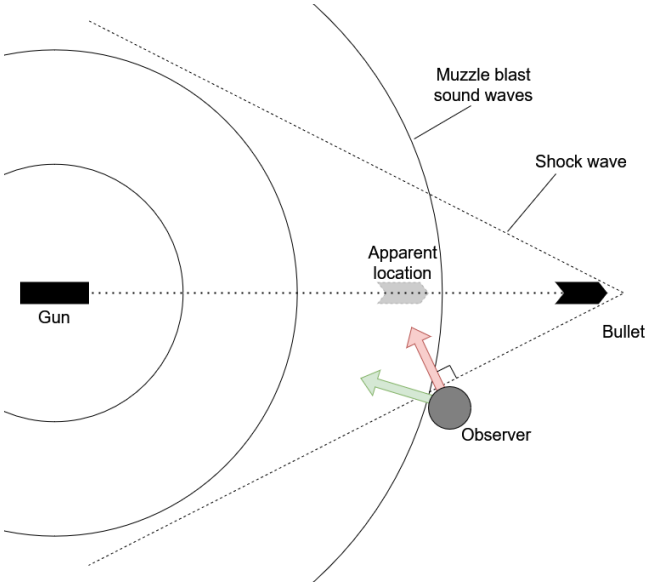


Figure 1.1: *How a shock wave and muzzle blast may appear to come from different direction to an observer downrange. The red arrow is the apparent direction from the shock wave and the green arrow is the direction based on the sound from the muzzle blast.*

A system that can detect and locate users of firearms can allow park rangers to respond quickly to poachers. Depending on the sensitivity and spacing of the individual sensors, it is likely that even air or CO2 powered tranquilizer guns could be located within seconds of a shot being fired.

1.2 Aim

This work aims to develop a novel method of detecting gunshots and estimating the location of the shooter. To perform this task a device that we call *Detection-Estimation Unit* (DEU) has been constructed. The implementation of this device will help to ensure that future poaching is difficult to the point of cessation. The DEU is a simple device consisting of a microcontroller, a microphone array and a solar power source had been created to perform the detection and estimation. The array contains 4 microphones, where currently one is used for detection and all are used for estimation. A DOA estimate is made by utilizing a new method that only relies on the received power to the microphones [31].

1.3 Delimitations

The finished DEU was made using a commercially available microcontroller and microphones to simplify development. The external chassis was made using a 3D printer to fit requirements specific to the project. The DEU is powered by a solar panel with accompanying 18650-batteries and a charge controller.

The data gathered for this project does not contain any shock waves. To observe a shock wave the observer needs to be downrange from a gun firing live ammo. For the shock wave to be distinct from the muzzle blast sound the observer also has to be at a distance from the shooter. Observing a shock blast requires a lot of safety procedures which were not able to be performed for this thesis.

Only shots from guns fired without a suppressor will be attempted to be detected in this project. As such no investigation into the efficacy of detecting with sound suppressors has been made. Gathering data with suppressors would add a difficult step to the data collection process, which is why it is excluded. If a device that detects and estimates non suppressed gun shot sounds could be made the same principals should be able to make a device that can do the same for suppressed gun shot sounds.

1.4 Component overview

During the project a series of prototypes were used. These varied in their layout, mostly based on current knowledge and limitations at the time when they where created. At the start of the thesis the plan was eight microphones in the array. However, this was quickly lowered to six to save on the limited number of microphones available and then to four when it was realized that the microcontroller only supported up to that number of microphones. Details about the prototypes and their development is found in Chapter 2.1. The final deployed DEU can be seen in Figure 1.2

Figure 1.2: *The DEU deployed in Ngulia.*

1.5 Operational overview

Since the shots are likely to be infrequent and it is not feasible to send a continuous stream of audio, the detection itself must be performed 'on the edge', i.e. on the DEU itself. The device can operate in one of two modes, continuous and non-continuous, with some variations in recording method. An explanation of these modes can be found in Section 2.3.1. Had networking been operational the device would then send the recording in a HTTP request over LTE together with the time of detection and the identity of the device. At the receiving server, a DOA estimate would be made and, if there were several detections made within a short period of time, a fused estimate to find the location of the shot could also be made. An approximate location could then be given as a latitude and longitude on a map. All the relevant information and the sound file could then be sent as an alert to wildlife protection services. The sound file is included so that a ranger could listen to the file to see if it is plausible that a shot has been detected and is not a false alarm.

1.6 Research questions

1. How can a gunshot be detected?
2. How can the location of a gunshot be estimated using the hardware setup?
3. How accurate is the detection/estimation?
4. How far away can a gunshot be detected?
5. What needs to be done so that the hardware can withstand the environment in the savannah?

1.7 Contributions

The project provided a quite obvious division of labour based on the expertise of the individuals involved. Timothy studying a master in data driven analysis and machine intelligence took responsibility of the detection part of the project. David studies a master in mechatronics and took responsibility for the estimation part of the project. The remaining work was divided to get hardware working, where Timothy did most of the work on the final DEU and David did most of the work on the second prototype as well as the Arduino code running on the microcontroller.

1.8 Related research

Another project that also is working on gunshot detection with machine learning is Safe Reaction [3]. Their model does not only aim to detect gun shot sounds but are trying to differentiate a lot of different sounds. Their website shows that they have managed to get a gun shot recall of 86% which is impressive and it shows that it is possible to detect gun shots well in some environments.

Way to estimate shooter location have seen research before such as in an article written by David Lindgren, Olof Wilsson, Fredrik Gustafsson and Hans Habberstad that did this using a distributed network of microphones [18]. An interesting discussion about the different sounds that a gun makes is included that may be relevant for future work.

The estimation method used in this thesis was developed by Gustav Zetterqvist, Fredrik Gustafsson and Gustaf Hendeby [31]. They had the idea of using received signal strength to estimate direction of arrival. This thesis implementation also tested how well it would work in practise.

Also related to novel ways of making DOA estimates is a method that uses a Taylor series expansion of the received signal [12]. The paper that describes this estimation method was written by a team consisting of Fredrik Gustafsson, Gustaf Hendeby, David Lindgren, George Mathai, and Hans Habberstad. This method also has benefits similar to the received signal power method such as it can have an arbitrary array configuration and use a much lower sample rate than traditional *time delay of arrival* methods.

1.9 Outline

This thesis consists of four large parts. The first part, Chapter 2, discusses the hardware used in the thesis. The chapter details the configuration and development of the prototypes and the components that make up the DEU are listed along with their purpose.

The second part is the detection which is discussed in Chapter 3. The chapter explores what sounds are produced when a gun is fired along with how to detect them using machine learning. The data that was gathered and made available to the models are also presented along with how the data was gathered. Also included in the chapter are what features were constructed and how they were chosen. How to evaluate a machine learning model along with an evaluation of the different models is also part of the chapter. It ends with an analysis of the detection algorithms performances as well as everything related to detection.

Chapter 4 presents the DOA estimation. The chapter describes how to the signal model is constructed and trained to preform the estimation using the received signal power method. Results are presented for two of the prototypes as well as from an array equipped with high quality microphones that was used to validate the capabilities of the code and provide comparison.

The final part, Chapter 5, presents our conclusions. A final evaluation of the different parts of the project is made here. A large portion is dedicated towards what should be done in the future. There are parts of the project that we have left unfinished and that should not be left so.

2

Hardware and software implementation

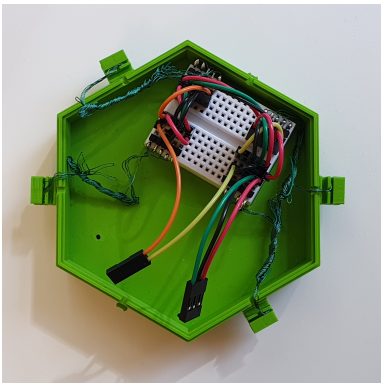
This chapter will discuss the hardware and software implementation that was used in the thesis work.

2.1 Prototype development

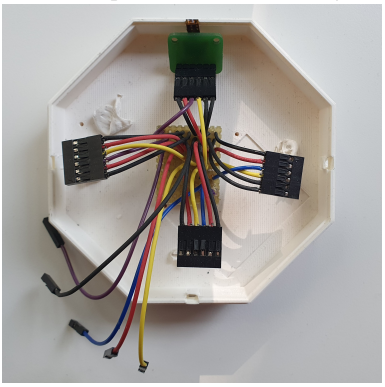
Before a built prototype was finished, some testing of the estimation algorithm was done on a "ReSpeaker USB Mic Array" equipped with four microphones. However this device had its microphones on the top and with an omnidirectional pickup-pattern which made it unsuitable for final implementation since the method relies on directional sensitivity of the microphones to be different. The first prototype, henceforth referred to as prototype 1, was a green hexagonal box that used four unevenly placed microphones at 0°, 120°, 180° and 300°. The first prototype was built when the availability of microphones was thought to be limited and had room for six in the array. However when the microcontroller arrived only four microphones where used as this was the limit.

The second prototype, henceforth referred to as prototype 1, was built with the knowledge from the first and had four evenly spaced microphones in the octagonal shell and were connected with the same PCB:s that where later used on the DEU. The microphones were placed at 0°, 90°, 180° and 270°. Both of these first two prototypes did not have internal room for the microcontroller or a power source and where instead powered with an external battery. The final DEU was built to also house its battery and solar power manager internally which created additional constraints on space. This was also the only one of the devices that could be fully sealed. Pictures of all the prototypes can be found in Figure 2.1.

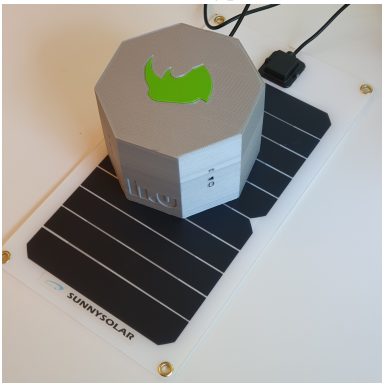
(a) ReSpeaker USB Mic Array



(b) Prototype 1



(c) Prototype 2. Unfortunately only one microphone is present in the picture.



(d) Final DEU with solar panel

Figure 2.1: *The devices used for testing up to final implementation.*

2.2 DEU hardware

The hardware in the final implementation of the DEU consisted of the following components:

- 1 Sony Spresense microcontroller with LTE extention board
- 4 Knowles Cornell II Digital MEMS microphones
- 1 SD-card 32GB
- 1 DFRobot Solar Power Manager 5V

- 1 Sunnysolar Semi Flexible Monocrystalline Solar Panel 5V
- 2 18650 Lithium-ion batteries
- 1 3D-printed shell with inserts for microphones
- Various cables and connectors

Some of the items on the list merit further examination and are discussed in the following sections.

2.2.1 Sony Spresense

The Sony Spresense is an unusually powerful microcontroller with 6 ARM Cortex-M4F cores and 1.5MB of RAM capable of running various embedded software solutions including Arduino. With an LTE extension board there is room for 4 digital microphones, an SD-card reader and a SIM-card holder.

There are some oddities and issues that have arisen from using the microcontroller. One of the strangest things was the fact that the Spresense would not operate if the board was exposed to direct sunlight. However rather than crashing outright, which would require a manual restart, the board seemed to just stop as if it had no power when in sunlight and resume working when back in the shade. Exactly why this happened was not investigated further, but did result in some lost recordings early in the project.

Other issues stem from what has been omitted from the Spresense. A WiFi module would simplify testing of networking components significantly easier even if the remote deployment location would instead use LTE. A battery monitor would also allow for finer control for power management including telling the device to sleep when battery levels are low. Another less urgent feature would be the ability to provide over the air updates to the firmware in order to test new models and provide software support remotely without having to connect the Spresense to a computer.

2.2.2 Shell

The device was left out during a night with rainfall and survived without damage to any of the internal components. For future implementations some care should be taken to make sure the microphones are well protected from the elements as they are the most valuable and exposed components.

2.2.3 Connections

The connections where rushed and the soldering made without proper knowledge of better methods which resulted in a system that risks being disabled by small disturbances.

2.3 Spresense software

The software running on the Spresense was written in Arduino in order to utilize the built in libraries exported from the Spresense SDK. This provided easy to use libraries for audio recording, file management and LTE connectivity. The model and its parameters were exported from the Edge Impulse project page and imported as an Arduino library. A code example was modified to run the inference process.

2.3.1 Operational modes for the detection model

The DEU can operate in one of two detection modes, each with its own advantages and disadvantages.

The more basic option is non-continuous mode where a recording is made into an inference buffer until the a full window of data has been collected, then a classification is made on the data and a prediction is produced. The buffer is then emptied and the process is repeated. During the time that the DSP and classification are being performed, no audio is collected, resulting in gaps where events may be missed. However, since the entire buffer is stored it can easily be saved to a file if desired. This makes it simple to send the actual audio in the case of a detection. However, this only saves one of the channels for reasons that will be discussed later.

Continuous mode uses smaller sample buffers, called slices, that are some fraction of the size of the full window. The slices are placed in an inference buffer as a FIFO sequence that corresponding to the window size. When a new slice is ready the oldest is removed from the back of the inference buffer and the new one is added. After each iteration, the inference process is performed on the current contents of the inference buffer. For example, a model with a model window size of 2 seconds and 4 slices would have 500ms slices and would run the inference process on each 4 times. This mean that even if a shot occurs on the edge between two slices it will not be missed. For the actual implementation a double buffering method is used. Two slice buffers are created, one is used for the audio sampling process and the other is used for the inference process. When the system is started, one of the buffers is filled with audio samples, while the inference process waits for this buffer to be full. When the buffer is full the inference process takes over and the sampling process starts sending the data to the other buffer. Each time a buffer is full, the buffers are swapped to ensure that there is always a place to collect the audio samples. Depending on the hardware and

model setup, the parameters for the number of slices and sample buffer size must be tuned to ensure smooth operation, otherwise crashes are likely to occur.

There are however other drawbacks to this approach. The entire window sized buffer is handled and stored internally by Edge Impulse and is seemingly inaccessible. Instead, a recording is instead started after a detection is made while the inference process is suspended. Storing the data separately in its own window length buffer is both inefficient from a data management perspective and may not be possible due to limited RAM.

A shared problem in both approaches is that only one of the microphones can send data to the model. In the case of non-continuous mode this also means that only one channel is recovered when saving the buffer. The first issue may be solved by sending each microphone to a separate subcore for processing. This will however require an inference buffer for each microphone, or two if running in continuous mode. If storing the whole window at the same time in some circular buffer is desired, there simply is not enough memory. Some things can be done to mitigate this issue: a shorter window, lower sampling rate and more slices can all be used to reduce memory usage. More slices will however lead to more time spent on processing and classifying the data.

2.3.2 Networking

The Spresense only officially supports Truphone and Soracom as LTE operators in a limited number of countries. These are operators that specialise in IoT applications. There was limited documentation as to how to connect with other operators including the one used in Kenya, Airtel. Despite using the same settings as other LTE-enabled devices the Spresense was unable to establish a connection to the cellular network. Whether this was due to an unknown incorrect setting, weak signal strength, compatibility error or some other reason, the source of the issue could not be found. As a result the project proceeded without networking capabilities.

3

Detection

When a firearm is discharged it produces an extremely loud sound that can travel several kilometers. Microphones can be used to record this sound and a detection algorithm can attempt to determine whether or not a recording contains a gunshot. This section will describe how to construct a detection algorithm that can determine if a recording contains a gunshot and what the results were.

As mentioned previously in Section 1.1, when a gun shot is fired two sounds are produced that could be of interest when designing a detection algorithm. Those two sounds are the muzzle blast and the shock wave from the bullet traveling past the observer. For this thesis the shock wave is ignored as it was not able to be observed. The muzzle blast contains more power and can always be heard when a bullet is fired, rather than just in a cone in the direction of the bullet. All the methods later described are therefore always only trying to detect the muzzle blast. A spectrogram of a shot can be viewed in Figure 3.1.

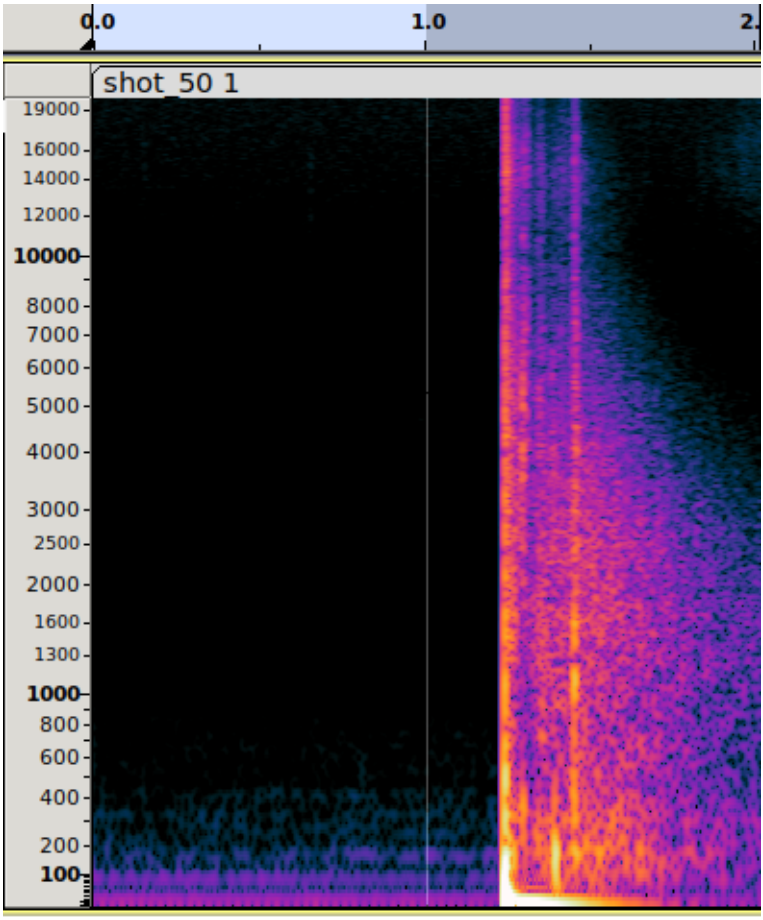


Figure 3.1: *A two second spectrogram of a gunshot.*

The figure shows the spectrogram of a gunshot that was recorded in Karlskrona, Sweden. As mentioned it only contains the sound of the muzzle blast. In the figure it is clear when the sound of the gunshot starts. The sound then dissipates, but where exactly the sounds is determined to end is a difficult question as the rumbling of a bullet can last for a few seconds. Most of the energy is present at the first few milliseconds after which some echos that contain a bit less energy and the rumbling that slowly dissipates.

Two second recordings were chosen with the idea of having one second overlap, with which there would be at least an uninterrupted second of any shot. If the shot started toward the very end of one recording that might not be enough to make a detection. Without overlap the next sound segment might not either be able to make a detection as most of the energy would not be present in it. With overlap the next recording would have the start of the sound and the following

second, which ought to be plenty to make a detection.

The other reason two second recording were chosen is that when a detection is made the recording could be sent to the rangers allowing them to listen to the sound and determine if it was a false alarm or if it was a shot. If the recording is too short it can be hard for a human to determine what is present in the audio.

3.1 Data

There are three types of data in the context of machine learning: training, validation and test data. Training data is used to train the model, validation data is then used to see how well the model performs when evaluating different methods or making feature selections. When the model is fully complete, it is evaluated against the test data to determine its overall performance.

The best kind of test data is made up entirely new data, with no dependence on the data in the training set. It is therefor important to not change the model after evaluating it on the test data, doing so could introduce a bias in the models toward the test data and the data is no longer independent. The purpose of the test data is to show how the models would perform when deployed where it will process new data. To this end, the test data in this thesis was gathered at a different occasion from the rest of the data to avoid any dependence to the rest of the data.

The files were manually labelled to indicate whether they contained a shot or not. If a shot could be discerned in the recording, a two second segment was extracted and labelled as a shot. Since the recording equipment always had four channels, each microphone was listened to and labelled separately. This also meant that there could be a scenario where one a shot could be heard on one microphone and was labelled as such, while another microphone at the same time could not discern the same shot. Since these sound segment cannot receive the same label it quadrupled the amount of labeling work.

3.1.1 Data collection

The first proper data set was collected during a military exercise at MSS Kvarn using Prototype 1 and an unfinished Prototype 2. Since the exercise involved the combatants hiding and moving through the forest there was little to no metadata, but the distance varied approximately from 0.1km to 1.5km as the audio collection equipment was moved further away from the training area. Also due to the nature of the exercise, sometimes more than shots were fired at the same time or in rapid succession.
The method to collecting the data without knowing exactly when a shot would be fired was to place the recording equipment in one spot and wait for the combatants to find each other, then moving some distance further away when shooting

ceased and repeating until the road ran out. This gave a data set with a wide range of distances to gunshots.

It was also at Kvarn that it was realised that the Sony Spresense would stop working when exposed to light as discussed in Section 2.2.1. Since neither prototype had a lid at the time to cover the micro controller, special care was taken to ensure that the device was kept in the shade, but this was not always possible when moving and as such some data was lost. The data from this occasion was chosen to be training and validation data with 67% of the points randomly determined to be training data.

Another set of data was collected from a lone rifleman on a training field in Karlskrona. Both Prototype 1 and Prototype 2 were used to collect the data. As it was possible to communicate without interfering with an exercise it was requested that the shooter fire a groups of 5-7 shots, then to wait a minute for us to move further way. This meant that the shots could be more clearly distinguished. Only distances up to 850 meters are present in this data set, beyond which shots could not be distinguished as the place was hilly and forested. The shooter used a Swedish *Automat Karbin 4* (AK4) which fired 7.62 x 51 mm NATO ammunition. This data set was selected as test data.

There are no shock waves present in the data set as they could not be observed safely. At Kvarn they were not firing live ammo, so a shock wave was not created since no bullet left the barrel. At Karlskrona live ammo was used but it was not deemed safe to be in front of the gun. Furthermore the shooting range was quite short, this means that even if the recording equipment could have been placed down range from the shooter it is certain that a shock wave would could be distinguished from the muzzle blast. The device being placed close to the gun means that the two sounds would have been so close together that the sound of the muzzle blast would probably have over powered the sound of the shock wave. All the negative data was gathered in Kenya at two occasions. One occasion was in the Ngulia rhino sanctuary with two DEU devices. The data from one device was made to be training and validation data, with the same split as the positive data. The other devices data was used as testing data. The other occasion was on the Kenyan savannah but not in the Ngulia rhino sanctuary, as the data from the sanctuary better depicts conditions in the rhino park this data was not suitable as testing data and was used as training and validation data.

3.2 Method

Two different approaches were explored to produce different detection algorithms. The first approach was a simple machine learning solution and the other was a neural network solution. To detect whether a shot has been fired, a supervised approach has been used in both cases. The supervised approach allows labelled data could be produced and fed through the algorithm, and can exclude sounds

that might sound similar to gunshots. An overview of the detection algorithm when using simple machine learning can be viewed in Figure 3.2.

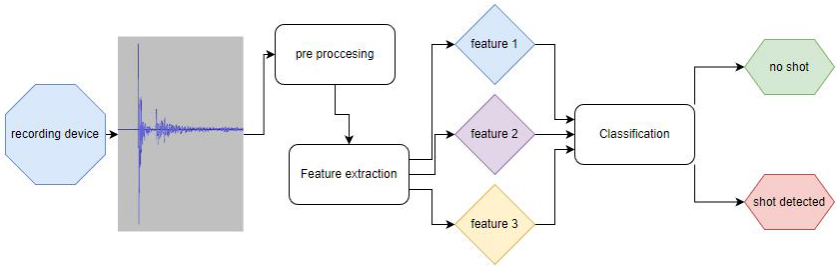


Figure 3.2: *Overview of the detection system.*

3.2.1 Feature based methods

The algorithms that has been explored for the feature based method are: Decision Trees, Naive Bayes Random forest, K-nearest neighbour and Support Vector Machines. More information on Decision Trees, Naive Bayes and Support Vector Machines can be viewed in [16].

Decision trees, DT, are models that sort the data using queries of the features. The order in which they query the features are arranged in a tree like structure, hence the name. Each node in a DT represents a feature and the branches from the node represent different values the node can assume. A classification is made starting at the root node where the first feature is queried and then its branches followed, once a leaf has been reached it will have a result and a classification for the file can be made. An example of a DT can be seen in Figure 3.3.

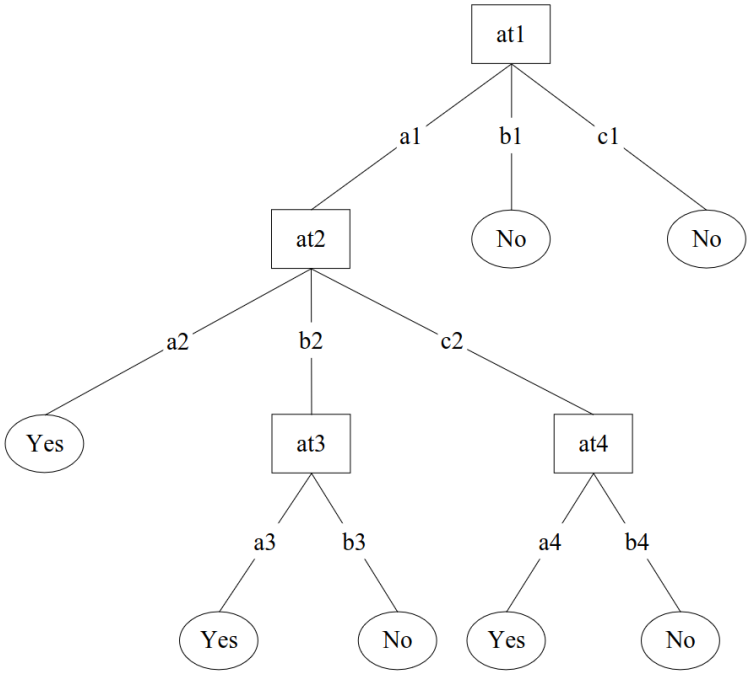


Figure 3.3: *A Decision tree*

The image is taken from [16], in the figure there are four nodes with different attributes or feature and from them there are branches with different alternatives depending on the value of the feature. In this classification a yes or a no classification is made.

Naive Bayes, NB, models work with a very simple equation $R = \frac{P(i|X)}{P(j|X)}$, where i and j are different classifications and X are the features. If R is greater than 1 it predicts class i otherwise class j. NB has short computational time since it is so simple.

Random forest models are an ensemble learning method [5]. During training several decision trees are constructed. When classifying, all the DT are tested and the class that most of the DT:s predicted is chosen. Random forests are not as prone to over fitting but can require more computing power than a decision tree since it needs to train multiple.

The K-nearest neighbour, KNN, method relies on comparing new data points position in the feature space to the position of the points in the training set [14]. The K points that are closest to the new point will decide the classification. The majority class of these K points will be the class for the new data point. In the event of a tie there are many different approaches to make a classification like increasing or decreasing k, randomizing and many more.

Support vector machines, SVM, relies on separating the classes using a vector or a hyperplane. The optimal plane should separate the classes with as large a margin as possible, all the training points should be far from the plane and the classes should be maximally separated. When a new data point is to be classified its position in the feature space is compared to the hyper plane that separates the classes. SVM is a very robust method that is not prone to over fitting due to the fact that the hyper plane cannot bend.

Feature generation

The part that most determines how well these kinds of algorithms perform is what features they have access to. The features are what the algorithms use to make their predictions. The raw data is the received signal from a microphone, in a two second window there are a lot of samples. To make the data more workable, a pre-processing step is performed. To reduce dimensionality, feature extraction is performed on the two second sample window. The features take the samples in the window into some mapping to make a new value, this results in a lower dimensionality. Features has been constructed using the principles described in [20].

There are two kinds of features explored in this thesis. First kind is the feature that takes some common mapping of the raw data and create their features from this mapping. This mapping usually highlights some aspect of the data aiding feature generation. The other kind of features are generated from the raw data directly and tries to describe some aspect or characteristic of the data.

Described in this first list are the features that are generated from some common mapping.

- Standard Deviation [17]. The feature is the standard deviation over the entire window. The idea is that the feature can estimate how noisy the window is.
- Mel Frequency Cepstral Coefficients, MFCC, [22]. MFCC measures how the sound changes over time. These features should have similar values for similar sounds making it possible to distinguish gun shot sounds.
- Spectrogram. A lot of different features are generated from the spectrogram. Some are which frequencies has the most energy and the highest energy peak. There is also a features that describes if the low frequencies had their energy peak at the same time, the idea is that a shot has energy in all frequencies and if they occur at the same time it might indicate a shot. These are also features describing how intense the energy peak is compared to its surroundings.
- Short time Fourier transform, STFT. The STFT has a lot of similarities to the spectrogram as such the same features are extracted from it.

- Zero crossing and zero crossing rate [29]. These features show how often the signal changes sign. The idea with these features are that gun shots ought to have similar values.

- Principal Component Analysis [4]. This measures what is most important in the signal. Similar sounds, such as gunshots, should have the similar values.

In this second list are features that are generated from the raw data

- Max difference. These features takes the differences between all points in the window that are separated by some number of steps. Different features have a different number of steps. The largest difference for each step size becomes the feature. The idea behind this feature is that when a shot occurs there will be a large and sudden change.

- Piecewise standard deviation. The window is divided into parts and the standard deviation is calculated for each of these parts. The resulting values are then used as input to calculate the standard deviation, the mean and the maximum.

- Piecewise mean. These features are similar to Piecewise standard deviation but for each of the smaller parts the mean is calculated instead of the standard deviation.

- Peak time. These features takes the points with the larges value and checks the surrounding points if they also have a high value. The idea here is that shot sounds are intense and dissipate quickly.

In total 62 features were generated from each window. The new feature space is significantly smaller than the raw input from the microphone but calculating all these features takes more than two seconds. This means that, if the system should run in real time, not all of these features could be used. Further dimensionality reduction was therefore necessary.

Feature selection

Once the new features have been generated, a sequential forward selection (SFS) scheme is used to determine which features were the best [15]. SFS is used because it is assumed that the best feature space does not contain a lot of features, which allows SFS to perform well. An illustration of how SFS works is shown in Figure 3.4. How well suited a combination of features in a model are for shot classification is determined by SFS in value called F1 score, that is described in section 3.3.

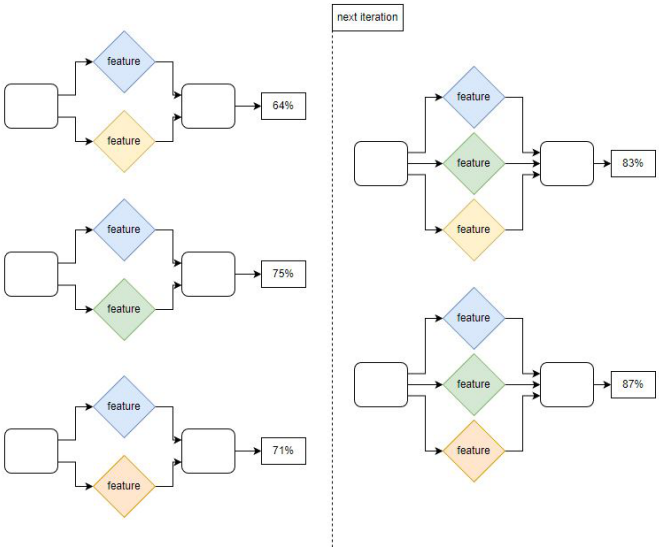


Figure 3.4: *Example of how SFS iterates, the accuracy of the features can be viewed in the rightmost box.*

To determine which feature based model structure is best for the task, all of them are tested. For each model, SFS is performed to determine the features to each model. It was decided to perform seven iterations of the SFS for each model.

3.2.2 Neural Net

The neural net was built using an online resource called Edge Impulse. It can build a neural net when provided with data. Once the model is built code can be generated, it can convert the code to an Arduino library code, which is more compatible to run on a micro processor [30].

Neural networks are more capable of handling large feature spaces. when the spectrogram feature was used for the regular machine learning models it had to be further processed to reduce dimensionality. The neural net model takes the entire spectrogram as its input. This is possible since the layers in the neural network reduce the dimensionality automatically. Since the dimensionality does not have to be reduced no information is lost, as a result all the information the neural network needs is in the spectrogram and it does not require any further input [23].

The spectrogram is treated like an image, therefore an image classification network is constructed. The implementation does not use standard image classification methods such as residual layers or batch normalization, because the final model can not use a lot of processing power as it has to run on a micro processor. Without a large network many standard ways of increasing model performance

cannot be implemented as they require to many layers or they are not necessary since there are not so many layers.

3.3 Evaluating machine learning models

When evaluating machine learning models the confusion matrix is very useful. A confusion matrix shows the truth and what the model predicted [27]. Each column shows what the model predicted and each row shows what the true label was. With two classes there is often one that of interest while the class is just everything else, like in this thesis where gun shot sounds are are being separated from all other sound. The class that is being separated is usually called the positive class and the other the negative class. The positive class is usually shown first in each row and column. The structure of a confusion matrix will be like in table 3.1.

	Predicted class 1	Predicted class 2
True class 1	True Positive (TP)	False Negative (FN)
True class 2	False Positive (FP)	True Negative (TN)

Table 3.1: *Confusion matrix*

When the positive class is labeled correctly it is called true positive and when labeled incorrectly it is called false negative, same for the negative class.

From the confusion matrix some useful matrices can be calculated [24]. These are shown in the following list.

- Accuracy, shows how well the model is at separating the classes. Accuracy is calculated as $\frac{TP+TN}{TP+FN+FP+TN}$.
- Precision, tells how correct the model is when predicting a specific class. To calculate precision the following equation is used $\frac{TP}{TP+FP}$.
- Recall, shows how many data points of a class the model predicted correctly of how many were expected. The following equation is used to calculate recall $\frac{TP}{TP+FN}$.
- F-score or F1-score, is an attempt to combine precision and recall into one metric. F-score is calculated by $2 \cdot \frac{precision \cdot recall}{precision+recall}$.

An example of a confusion matrix can be viewed in table 3.2.

	Predicted class 1	Predicted class 2
True class 1	16	4
True class 2	1	9

Table 3.2: *Example of confusion matrix*

Row one shows the true labels of class 1, the positive class. There are a total of $16 + 4 = 20$ points in class 1. Of the 20 points in class 1, 16 were labeled as class 1 which is correct, i.e. TP. Similarly class 2 contains 10 points, of which 9 were labeled correctly, i.e. TN. The metrics for this class can be viewed in the table 3.3.

	Precision	Recall	F-score	Accuracy
Class 1	$\frac{16}{16+1} = 0.94$	$\frac{16}{16+4} = 0.8$	$2\frac{0.94 \cdot 0.8}{0.94+0.8} = 0.86$	
Class 2	$\frac{9}{9+4} = 0.69$	$\frac{9}{9+1} = 0.9$	$2\frac{0.69 \cdot 0.9}{0.69+0.9} = 0.78$	
				$\frac{16+9}{16+4+1+9} = 0.833$

Table 3.3: *example of model quality*

3.3.1 Unbalanced data sets effect on model precision

An unbalanced data set has the effect that the majority class precision increases while the minority class precision decreases. To illustrate this imagine a model with 90% accuracy for each class. Giving the model an even testing data set containing 10 class A points and 10 class B points results in the following confusion matrix.

	Predicted class A	Predicted class B
True class A	9	1
True class B	1	9

Table 3.4: *Example of a confusion matrix on a balanced data set*

This results in 90% precision and recall for both classes. If the testing data set given to the model is instead 10 class A points and 100 class B points the following matrix would emerge:

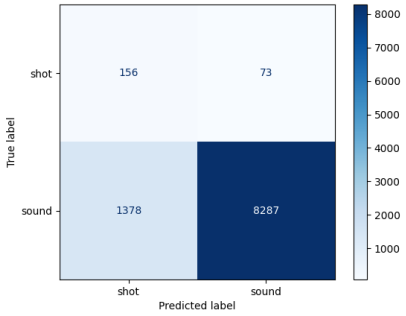
	Predicted class A	Predicted class B
True class A	9	1
True class B	10	90

Table 3.5: *Example of a confusion matrix on an unbalanced data set*

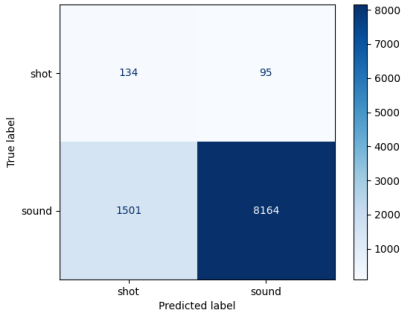
Now, the same model has achieved a class A precision of 47% and a class B precision of 99%. The minority class precision has been decreased and the majority class precision has increased even though the model has not changed. Note that both still have a recall of 90%.

3.4 Results

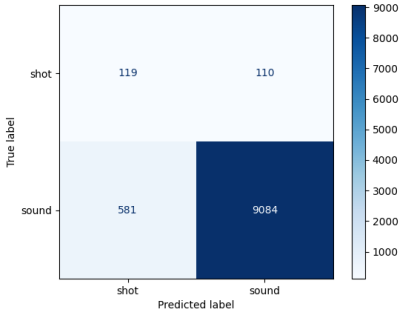
The training and validation data available to the models were roughly two hours split equally between shot and non-shot sounds. For the testing data set more than five hours were available but only 7 minutes were shot sounds. The confusion matrices for the different model can be viewed in Figure 3.5.



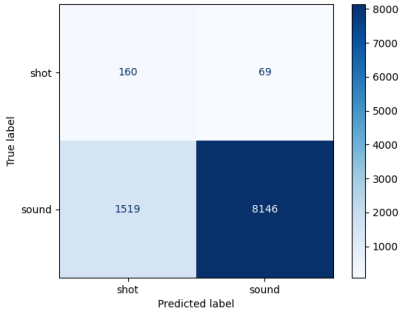
(a) Confusion matrix for the decision tree model



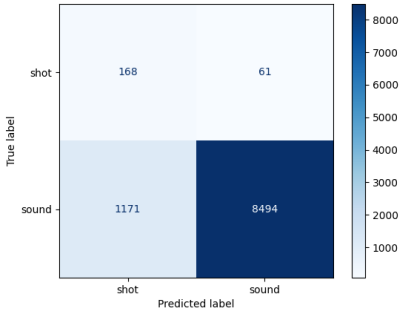
(b) Confusion matrix for the K-nearest neighbour model



(c) Confusion matrix for the naive Bayes model



(d) Confusion matrix for the random forest model



(e) Confusion matrix for the support vector machine model

Figure 3.5: *Confusion matrices of the feature based models.*

The neural network was also trained on the same data and achieved the results seen in 3.6.

	Predicted shot	Predicted sound
True shot	130	99
True sound	1434	8231

Table 3.6: *Confusion matrix for the neural network model*

All the performance metrics are tabulated in Table 3.7.

		Precision	Recall	F-score	Accuracy
Decision Tree	Gun shot	10.2%	68.1%	17.7%	
	Sound	99.1%	85.7%	91.9%	
					85.3%
K-Nearest Neighbour	Gun shot	8.2%	58.5%	14.4%	
	Sound	98.8%	84.5%	91.1%	
					83.9%
Naive Bayes	Gun shot	17.0%	52.0%	25.6%	
	Sound	98.8%	93.9%	96.3%	
					93.0%
Random Forest	Gun shot	9.5%	69.9%	16.7%	
	Sound	99.1%	84.3%	91.1%	
					83.9%
Support Vector Machine	Gun shot	12.5%	73.4%	21.4%	
	Sound	99.3%	87.9%	93.3%	
					87.5%
Neural Net	Gun shot	8.3%	56.8%	14.5%	
	Sound	98.8%	85.2%	91.5%	
					84.5%

Table 3.7: *All the performance matrix for the different models*

The feature based models all had an SFS performed on them. The features that best described the data can be viewed in Table 3.8

Decision Tree	3 MFCC 1 zero crossing 1 picewise mean 1 picewise std 1 peak time
K-Nearest Neigbour	5 MFCC 1 peak time 1 max differance
Naive Bayes	6 MFCC 1 peak time
Random Forest	5 MFCC 1 peak time 1 max difference
Support Vector Machine	6 MFCC 1 picewise std

Table 3.8: *Features chosen by the SFS for the different models*

3.5 Discussion

The testing data set is unbalanced, and this is that to better mimic the real world where background noise is orders of magnitude more common than sound of gunfire. There were 229 files containing shots and 9665 that did not, in other words there is a lot more negative data than positive in the test set. Even this uneven distribution is rather balanced compared to real world data. The distribution is such that there is 42 shots for every hour of background noise which is a lot more then there would be in reality, where one shooting a year would be a lot. The impact of the unbalanced data set is that all gunshot precision is lower and sound precision is higher, but recall is unaffected. The unbalanced data set also has the effect of making sound recall and accuracy highly correlated.

The naive Bayes model was the model that had the best overall accuracy. It had the best precision on shots, because it had a high sound recall, and it had the worst recall for shots. Since the test data set is very unbalanced this means that to achieve a good overall accuracy the recall on the negative class is the most important thing. However, of the models tested it is still believed to be the best. If a system gives false alarms excessively often it will stop getting used even if it had a 100% recall for positive data. A system that only has 50% recall for positive data but with very low false alarm rate would be better, since every warning would be taken seriously and acted on. If one assumes that a poacher fires more than one shot even low recall gets a high chance of detecting at least one. For example with two shots the chance of catching at least one becomes 77% even with just a 52% recall. Even so, the naive Bayes model which had the lowest false alarm rate still would give almost 4000 false alarms each day which is far to many to be useful.

The SVM model stands out from the rest with a very good gun shot precision as well as the highest shot recall and the second highest accuracy. It is surprising that naive Bayes model was the model with the highest accuracy since the model is generally outclassed by the other models. It only achieved this because it heavily favored sound classification at the expense of shot classification, shown by the fact that it has the worst shot recall of any model. Even so, low false positive is beneficial for this task, making naive Bayes the best. With a new set of features or better features it is plausible that SVM would still have among the highest accuracy while naive Bayes would not.

To improve the performance of the system the first step is always more data. For this project there was plenty of negative data but more positive data would improve the models. Another way of improving the performance is to construct better features. This is a hard step as it is difficult to predict what kind of features will be good, as they need to represent some aspect of the data well.

The model with the lowest false positive rate, naive Bayes, still had to many false positives to be useful. When evaluating model performance in the SFS shot F-score was used. To reduce false positives, another metric like shot precision or sound recall might be better, since this would create models that priorities keeping the false positive rate low. This might however have made the models worse in some other metric. The SFS might also be further improved with some tuning. Currently, it performs seven iterations for all models, and increasing this number may improve the quality of the features selected. SFS is performed because there is not enough time to calculate all the features in real time. To improve the SFS the number of iterations performed could depend on the computing demand of the chosen features, with another feature only being chosen if doing so would not exceed the computing power available. Another approach is to only perform another iteration if the previous iteration yielded a large enough improvement, i.e. having a stopping criterion like an optimization algorithm. The final feature environment would need to be verified that it did not require more computing power than were available. A the very least, the number of iterations performed ought not be static.

Another way of reducing false positives is to change the classification threshold, giving the model a larger bias for one class. Giving a model a larger bias for the sound class would make it chose the sound class more often. However, doing this as a rule reduces the overall accuracy and since the shot recall is not great for any of the models it might not be a good trade off. It is probably better to improve the data, change the features or tweak the SFS.

The device is intended to be installed in Kenya to catch poachers, currently all the positive data is gathered in Sweden further more gathered in Swedish forests and dense pine forests are not a feature of the Kenyan savannah. The assumption has been made that the only impact from the forest is to make the shot appear

further away, however this assumption could be false. It could be the case that the sound of a shot traveling through forest significantly alters the characteristic of the shot. Because of this, the device should be tested on positive data from the savannah or savannah like environment prior to deployment.

Something observed while collection data was that at the furthest ranges the human ear can quite clearly hear the gun shots. However, in the data it was hard or sometimes impossible to distinguish the gun shot from the background or even to hear if the sound was present at all. If a microphone with the ability to record lower amplitude sounds, it could increase the range of the detection. Such a microphone was not chosen for this project as it had to be physically small and cheap to allow for large scale deployment.

As stated in section 3.1.1 the furthest ranges the a gun shot was observed was estimated to be 1.5km through forest. For this thesis we have assumed that the only impact is that the sound of a shot traveling through forest makes the shot appear to be further away. If this assumption holds a shot should be able to be identified at longer ranges on the savannah, we would expect at least double. When collecting the data we noticed that the sound level dropped sharply when just a small grove of trees was placed in between the shooter and the device. The caliber of the weapons used will also affect the range at which they can be distinguished. At Kvarn we did not know which type of weapon was fired but a clear difference in sound level could be observed between some on the guns. It is most certainly the case that at the furthest ranges only the most loud guns were audible. The ranges for the less audible guns are probably lower than 1.5km but we have no way of knowing which these guns are or what the ranges are.

Something that is worrying is that at the furthest distances the direction of the incoming sound seemed to matter a lot in regards to how distinguishable the shot was. It could be the case that the microphone in the shots direction could quite clearly hear the shot while the microphone at the other end of the device could not hear the shot at all. In our current implementation the detection algorithm only detects on one microphone to save processing time, this means that if we manage to get a detection range of 3 km it will not be 3 km in all directions. A solution is to run detection on alternating microphones, this will not give a bias toward a direction but it could still mean that the device could miss a fired shot within range. Another solution is to make a weaker detection algorithm that can run faster, enabling two detection algorithms to run in parallel on different microphones. This could result in greater coverage area depending on how much worse the algorithms has to be. Increasing the processing power could also be a solution.

A limitation of our testing set is that it does not contain shots occurring at the same time or very close to one another as the data is all gathered from a single man shooting, and not firing full automatic the shots always appear one at a time. In the training set we have plenty of data where shots are appearing at the same

time or very close to one another making one set of two second contain multiple shots. This fact is not something that increases the accuracy of the models as the files which contain multiple shots in them are easier for the model to detect. The accuracy ought to be a bit better if such data was also present in the testing set.

There was a high presence of wind in the negative testing data that was not present in the rest of the data, this is probably a factor to why the results are not better. The station that produced all the negative data was placed in precarious situation with meant a lot more wind, if more wind would be present in the rest of the sets it could aid detection. The reason for this station being chosen to produce test data and not training data was because it was the single device that had produced the most data, if it had not been so the test set would be significantly smaller. Another way to mitigate this is to place some cover on the microphones, such a cover was planed for but it was not implemented.

Our data has no meta data of distance to shooter. Rough distances in a set are known but any individual shot there is no estimation for. It is probably the case that it is easier to detect shots that are closer as there is more energy present and the sound of the shot is not overpowered by other sounds in the surroundings. We can not know how well the models performs at different distances only how well it performs for all of them. Before deployment some data with good meta data should be gathered to see for which ranges the models can detect and with what performance.

We were not able to collect any data that contained a shock blast as this requires you to be near the bullets path, which we could not do safely. We do not believe that this effected the project much as poachers are close to the animals when they shoot, for a shock wave to be observed the bullet must miss and travel in the direction of the DEU. Even if they are not close to the animal they have to shoot over the DEU for a shock wave to be observed. This fact ought not effect the performance too much as, even when it is present, the shock wave is usually not as audible as the sound of the muzzle blast. Furthermore, a shock wave will not always be present in the practice either and a detection model should be able to detect a shot without the shock wave being present.

Whenever a shot detection is made an alert along with the sound file the was flagged as containing gun shot should be sent to the rangers. The reason to send the sound file is so that they can listen to it and determine if it is a false alarm or if some action should be taken. For a human it is easier to determine if it is a real gunshot if the file is a bit longer so that they can hear what is going on. The actual sound of the shot happens quickly and as such a short file could be better for the detection, however too short will hinder human verfication. It is possible that a shorter file might be easier for the detection algorithm to classify, this would need to be investigated. If the optimal file length is to short for a human to interpret, more audio around the detection could also be sent. There was not enough time in this thesis to investigate this.

When inspecting Table 3.8 we can see that some of the MFCC features are always chosen and the none of the spectrogram or STFT features are chosen. The three methods are similar so it makes sense that only the best would be chosen, but it is interesting that the MFCC was chosen so often while the other were not. It might be the case that the feature extracted from the spectrogram or the STFT are poorly designed and needs to be looked over. Furthermore we can see that a peak time feature was often chosen, which is understandable as it describes the largest peak in a file, which should be very distinct for a gunshot. If the loudest sound in a file is not a shot, a peak time feature could help a model identify that.

4

DOA Estimation

Once a gunshot has been detected the direction of arrival needs to be calculated. Conventionally, DOA estimation is achieved by measuring the slight difference in arrival time of a signal to an array of measurement devices. These are called *time-difference of arrival* (TDOA) measurements and give an estimate by solving a *non-linear least squares problem* (NLS) problem. Beamforming methods can also be used for DOA since the phase of the incoming signal means that frequency domain analysis is also possible, assuming a coherent wave[11, 31]. These methods require a precise and relatively large spacing in the array to create differentiable signals. For a TDOA method the distance between the sensors need to be at least half the wave length and are therefore unsuited for a small device that needs to detect a broad range of frequencies. In this thesis a new method of DOA estimation is explored that only needs the received power to make an estimate. The chapter will first explain the new method used for DOA estimation, the results for a number of arrays and a discussion of the results.

4.1 Method

Each device performs its own DOA estimation using received power of the microphones with a method developed by Gustav Zetterqvist, Fredrik Gustafsson and Gustaf Hendeby[31]. The method consists of a training phase to compensate for the different gains of the microphones, frequency dependency and model the directional sensitivity. From this directional sensitivity an estimate can be performed. Note that while this method also relies on signal strength it is not to be confused with more typical *received signal strength* (RSS) estimations that measure distance.

4.1.1 Signal model

The measured signal from microphone i can be expressed as

$$y_i(t) = s_i(t) + w_i(t) \tag{4.1}$$

where y_i is the measured signal from microphone $i = 1, 2, \dots S$ that consists of the received signal $s_i(t)$ and measurement noise $w_i(t)$ that is assumed to be normally distributed $\mathcal{N}(0, \sigma_i^2)$. The power of the signal from microphone i is measured at a discrete time denoted l, and can be calculated as

$$P_i = \frac{1}{L}\sum_{l=1}^{L} y_i(l)^2, \tag{4.2}$$

where L is the number of samples. Inserting (4.1) into (4.2) means the expression can be rewritten with three terms

$$P_i = \underbrace{\frac{1}{L}\sum_{l=1}^{L} s_i^2(l)}_{P_i^s} + \underbrace{\frac{1}{L}\sum_{l=1}^{L} 2s_i(l)w_i(l)}_{P_i^{sw}} + \underbrace{\frac{1}{L}\sum_{l=1}^{L} w_i^2(l)}_{e_i} \tag{4.3}$$

where P_i^s is power of the received signal, P_i^{sw} is power of the cross-term between signal and noise, and e_i is power of the measurement noise. The number of samples L is assumed large and as such the cross-term P_i^{sw} will approach zero. The normal distribution of $w_i(l)$ means that e_i will have a chi-squared distribution with L degrees of freedom. Again, since the number of samples is large this chi-squared distribution can be approximated by a normal distribution.

$$\frac{L}{\sigma_i^2} e_i \sim \chi_L^2 \xrightarrow{Approx} e_i \sim \mathcal{N}(\sigma_i^2, \frac{2\sigma_i^4}{L}) \tag{4.4}$$

The aim is to use this information to get a DOA estimate ψ, i.e. the angle to the object being tracked. Each microphone is assumed to have a directional sensitivity in its power attenuation, either by design or construction of the array. Since the microphones are placed close together in the array the absolute level of the power received is assumed to be equal for all microphones. The absolute level of power is denoted α and is considered a nuisance parameter to the estimation. P_i can now instead be written as a function dependent on ψ,

$$P_i(\psi) = \alpha g_i h(\psi, \boldsymbol{\theta}_i) + e_i \tag{4.5}$$

where g_i is the microphone gain, $h(\psi, \boldsymbol{\theta_i})$ is the directional sensitivity of the microphone that depends on the angle ψ and parameters $\boldsymbol{\theta_i}$, and e_i is the error described in equation (4.4).

4.1.2 Training

The array is exposed to a signal of wide-band noise from a number of directions. The angle to this sound source and signal received by each microphone is observed. The power measured by each microphone is then calculated using equation (4.2). The parameters to the directional sensitivity $\boldsymbol{\theta}$ and gain g_i is then estimated by solving the following non-linear optimization problem. In this thesis, the parameters to the optimization problem were found using YALMIP [21] with the FMINCON solver.

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & V(\boldsymbol{x}) \\ \text{subject to} \quad & \alpha > 0, \\ & g_i > 0, \\ & h(\psi_i, \boldsymbol{\theta}_i) = 1 \ \ \forall i = 1, 2, \ldots, S, \\ & \sum_{i=1}^{S} g_i^2 = 1 \end{aligned} \tag{4.6}$$

where ψ_i is the angle when the microphone i directly faces the source of the sound and S is the number of microphones. $\boldsymbol{x}$ contains the optimization variables: α, $\{g_1, \ldots, g_S\}$ and $\{\boldsymbol{\theta}_1 \ldots \boldsymbol{\theta}_S\}$. The loss function $V(\boldsymbol{x})$ is defined as

$$V(\boldsymbol{x}) = \sum_{i=1}^{S} \frac{L}{2\sigma_i^4} \sum_{k=1}^{K} (P_i(\psi_k) - (\alpha g_i h(\psi_k, \boldsymbol{\theta}_i) + \sigma_i^2))^2. \tag{4.7}$$

where K is the number of observed directions.

4.1.3 Fourier Series Model

In order to model the directional sensitivity $h(\psi, \boldsymbol{\theta_i})$ for each microphone a Fourier series model

$$h(\psi, \boldsymbol{\theta}_i) = \theta_0^i + \sum_{d=1}^{D} \theta_{d,c}^i \cos d\psi + \theta_{d,s}^i \sin d\psi \tag{4.8}$$

is utilized where D is the order of the FS. This order is determined by using the Bayesian information criterion (BIC) which aims at minimizing the code needed to store data [19].

$$BIC = V(\boldsymbol{x}) \left(1 + (2D + 1) \frac{log(K)}{K}\right) \tag{4.9}$$

4.1.4 Frequency dependency

Empirical testing shows that frequency content of the signal affects the directional sensitivity and the gain of the individual microphones. To compensate, a frequency dependency was introduced to the parameter of the directional sensitivity, $\boldsymbol{\theta}_i(f)$, and to the microphone gain $g(f)$, where f is the frequency content

of the signal. The training phase is altered to include the frequency dependency. The chosen solution is to use adjacent band-pass filters to separate the wide-band noise signal into discrete frequency bands over the whole spectrum. Directional sensitivity for each band-pass and microphone can then be determined by solving the optimization problem (4.6) in that range of frequencies. To account for the increased complexity this introduces the BIC is slightly redefined to also sum over the band-pass filters

$$BIC = \sum_{m=1}^{F} V(x)\left(1 + (2D+1)\frac{log(K)}{K}\right) \tag{4.10}$$

where F is the number of filters.

4.1.5 Estimation

Following the training the signal power model for all the S microphones can be written on vector form as

$$\hat{\boldsymbol{P}}(\psi) = \alpha \boldsymbol{G}(f)\boldsymbol{h}(\psi, \hat{\boldsymbol{\theta}}(f)) + \boldsymbol{e} \tag{4.11}$$

where $h(\cdot)$ contains the directional sensitivity of the microphones, $\hat{\boldsymbol{\theta}}(f)$ contains all of the estimates $\hat{\theta}_i(f)$ and $\boldsymbol{G}(f)$ is a gain matrix with the estimated microphone gains $\hat{g}_i(f)$ on the diagonal. For estimation data the frequency content of the received signal is calculated and used as a gain vector to the band-pass filters of each microphone. Power of the received signal in the frequency range is denoted p_m for filter m, and the range is $[f_m - B/2, f_m + B/2]$ where B is the bandwidth of the filter. The power can then be estimated as

$$\hat{\boldsymbol{P}}(\psi) = \sum_{m=1}^{F} p_m \boldsymbol{G}(f_m)\boldsymbol{h}(\psi, \hat{\boldsymbol{\theta}}(f_m)) + \boldsymbol{e}. \tag{4.12}$$

Least squares can then be used to calculate the DOA estimate of ψ.

$$\hat{\psi} = \arg\min_{\psi} \left\| \frac{\boldsymbol{P}}{\|\boldsymbol{P}\|} - \frac{\hat{\boldsymbol{P}}(\psi)}{\|\hat{\boldsymbol{P}}(\psi)\|} \right\|^2. \tag{4.13}$$

4.2 Results

Continuous testing with the prototypes was done as they became available. Three sets of results where produced from three different array setups and calibration environments.

4.2.1 Known good calibration

To verify the capabilities of the code, calibration data previously collected in an anechoic chamber by our supervisor Gustav was used. This original array used eight microphones and to mimic the layout of the DEU half where removed in the analysis, leaving only every second microphone. The data set contained 34 angle data points with 1° increments between 0 and 5 followed by 5° increments between 5° and 45°, and finally 15° increments for the rest. To determine a good range for the band pass filters the signal was made into spectrogram to determine its frequency contents.

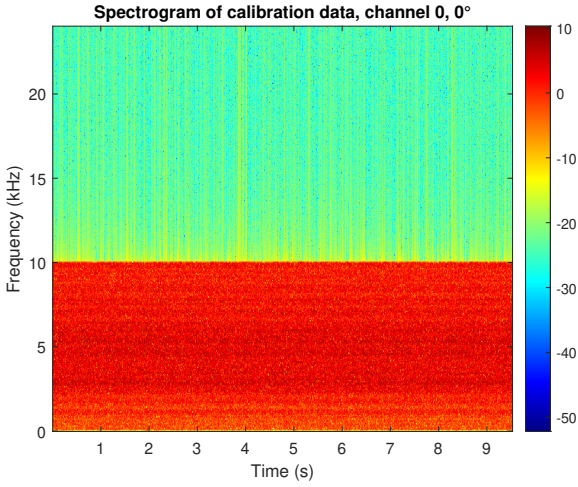
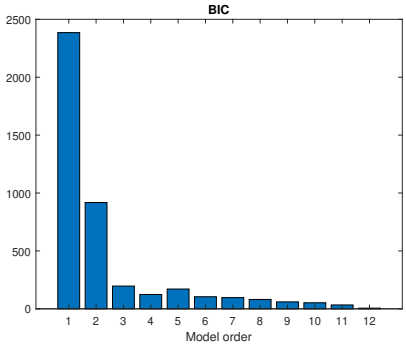
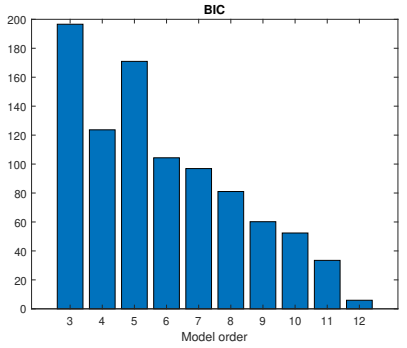


Figure 4.1: *The spectrogram of the signal from the first microphone at* 0° *using the high quality array.*

Figure 4.1 shows that a signal of white noise with a cutoff frequency around 10kHz. Based on this result, the signal was divided into bandpasses 200Hz wide from 200Hz to 8000Hz. The upper limit was chosen so that future calibrations that used lower sampling rates such as 16kHz could be compared. Otherwise this would cause aliasing when a lower sampling frequency is used [28]. To determine the best Fourier model order, the BIC defined in Equation 4.10 was used. Model orders between 1 and 12 were tested as running the YALMIP optimizer on higher orders took increasingly longer time.



(a) A bar plot of the BIC compared to the model order



(b) Zoomed version of the previous figure

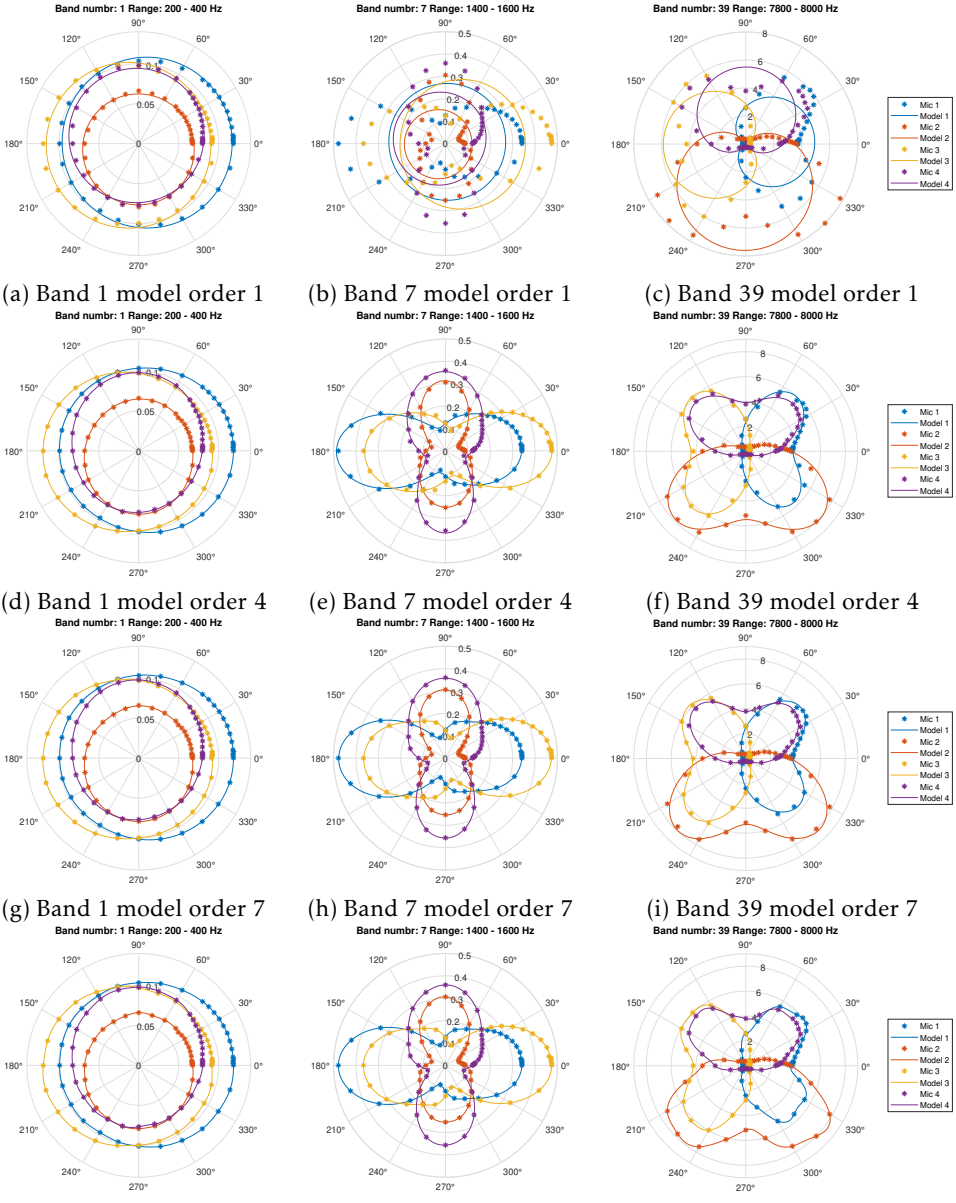
Figure 4.2: *(a) and (b) show the value of the BIC compared to the model order for the high quality array.*

Figure 4.2 shows that the BIC continues to decrease with increased model order. Lower BIC is better but takes a long time to find a solution for. Therefore a model order of 1, 4, 7 and 12 was tested to see if this affected the performance of the estimation.

After model training was complete, the same white noise audio was fed back into the model to assess its ability to estimate known sound sources. The true measurements and the models built with different model order are shown in Figure 4.3. The maximum error and standard deviation are presented in Table 4.1, while the raw angle errors depending on angle can be seen in Figure 4.4.

Model Order	1	4	7	12
Standard deviation [°]	5.4645	1.7425	1.7498	1.7973
Maximum error [°]	10.92000	4.4400	4.8000	5.880
Calculation time [s]	21	46	124	603

Table 4.1: *The performance data of the estimation for the different model orders when using the high quality array.*



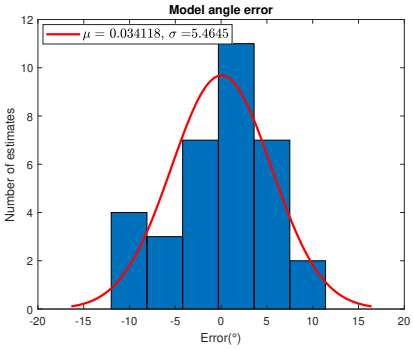
(a) Band 1 model order 1 (b) Band 7 model order 1 (c) Band 39 model order 1

(d) Band 1 model order 4 (e) Band 7 model order 4 (f) Band 39 model order 4

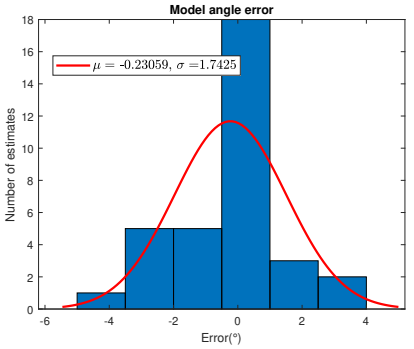
(g) Band 1 model order 7 (h) Band 7 model order 7 (i) Band 39 model order 7

(j) Band 1 model order 12 (k) Band 7 model order 12 (l) Band 39 model order 12

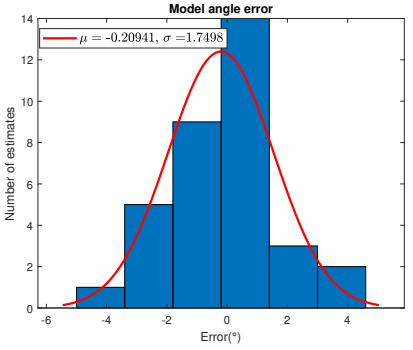
Figure 4.3: *The Fourier series approximation for different model orders and frequency bands when using the high quality array.*



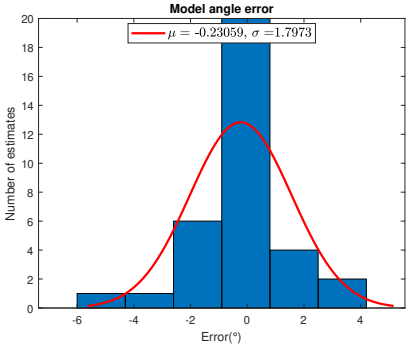
(a) Model order 1



(b) Model order 4



(c) Model order 7



(d) Model order 12

Figure 4.4: *Histogram of the magnitude of the angle errors when using the high quality array. A normal distribution has been fitted to the histogram.*

4.2.2 Prototype 1

This prototype is the same prototype 1 as referred to in Chapter 2, and was calibrated with a Bluetooth speaker inside an empty room. The calibration data set contained 18 data points from angles with 20° increments from 0° to 340°.

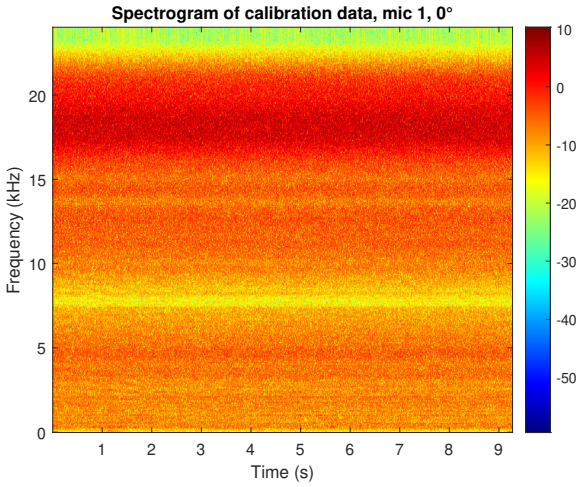
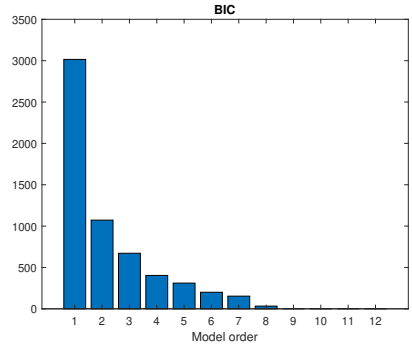


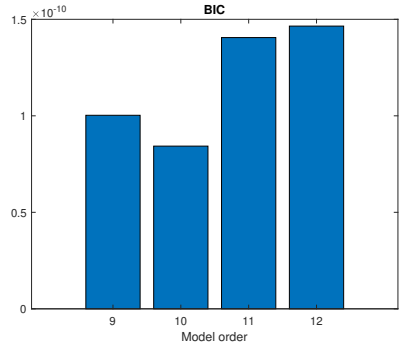
Figure 4.5: *The spectrogram to the signal recorded by the first microphone at* 0° *using prototype 1.*

Since the sound file was different than the one used previously there is no cut off frequency. The received power varies quite significantly based on frequency but the model is made to account for that. To get consistent and comparable results the same bandpass intervals of 200Hz wide and a range of 200Hz to 8000Hz were used.

Again BIC was used to determine the best model order. Results can be seen in Figure 4.6.



(a) A bar plot of the BIC compared to the model order



(b) Zoomed version of the previous figure

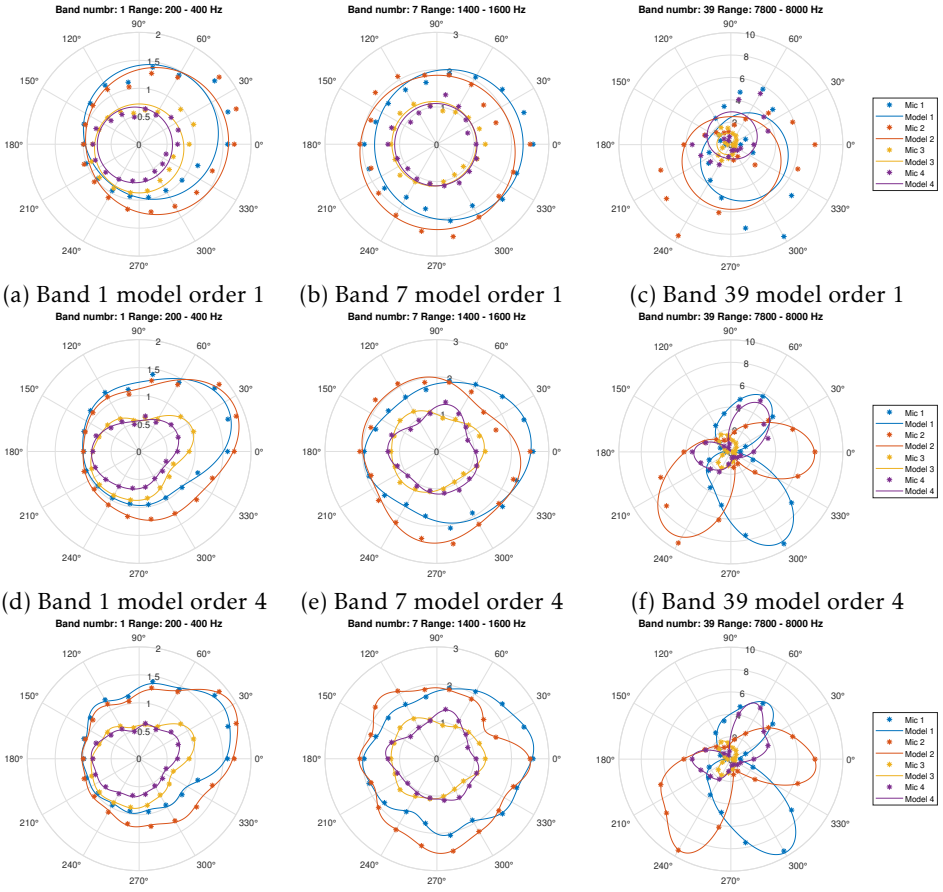
Figure 4.6: *(a) and (b) show the value of the BIC compared to the model order when using prototype 1.*

Doing this analysis quickly revealed a problem with the calibration. Going over a model order of 8 caused the YALMIP solver to assign NaN values to one of the θ factors in the directional sensitivity model. Extensive testing with different optimization parameters could not resolve the problem and the exact cause for the error could not be identified. Therefore the maximum possible of 8 as well 1 and 4 where chosen as model orders in the analysis.

Like before the sound used for training was also used to evaluate the estimation ability. The true measurements and the models built with different model order are shown in Figure 4.7. The maximum error and standard deviation are presented in Table 4.2, while the raw angle errors depending on angle can be seen in Figure 4.8.

Model Order	1	4	8
Standard deviation [°]	23.2460	29.7073	32.2087
Maximum error [°]	41.7200	76.5200	79.4000
Calculation time [s]	15	41	98

Table 4.2: *The performance data of the estimation for the different model orders when using prototype 1.*

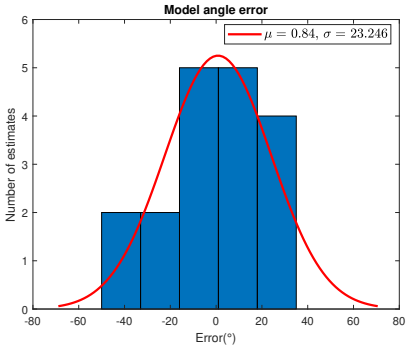


(a) Band 1 model order 1 (b) Band 7 model order 1 (c) Band 39 model order 1

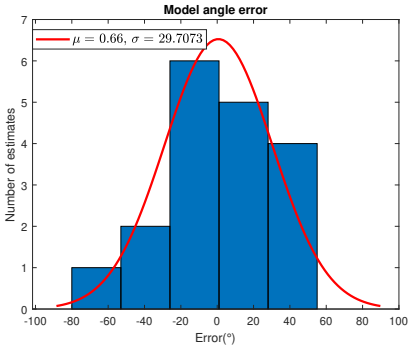
(d) Band 1 model order 4 (e) Band 7 model order 4 (f) Band 39 model order 4

(g) Band 1 model order 8 (h) Band 7 model order 8 (i) Band 39 model order 8

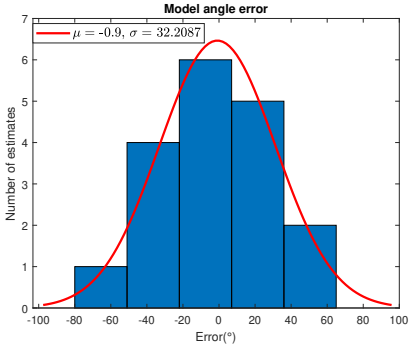
Figure 4.7: *The Fourier series approximation for different model orders and frequency bands when using prototype 1.*



(a) Model order 4



(b) Model order 7



(c) Model order 8

Figure 4.8: *Histogram of the magnitude of the angle errors when using prototype 1. A normal distribution has been fitted to the histogram.*

4.2.3 Prototype 2

Additional time was scheduled to test the next prototype in an anechoic chamber to see if this would improve the results. For this purpose, the new and improved prototype 2 described in 2.1 was used. The calibration data set included 12 data points from angles with 30° increments from 0° to 330°. The same sound file that the high quality array was exposed to was used for the calibration.

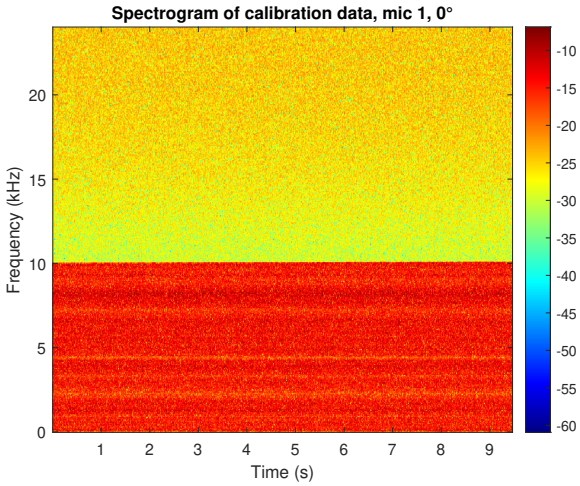
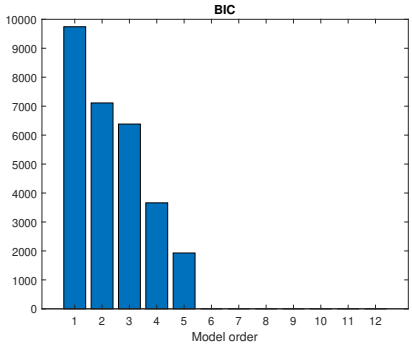


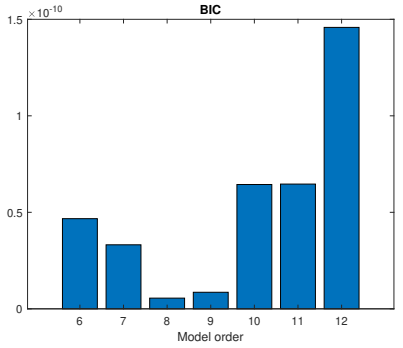
Figure 4.9: *The spectrogram to the signal recorded by the first microphone at 0°.*

Like the calibration data for the high quality array, Figure 4.9 shows a signal of white noise with a cutoff frequency around 10kHz. The signal was again divided into bandpasses 200Hz wide from 200Hz to 8000Hz to get comparable results.

Again BIC was used to determine the best model order. Results can be seen in Figure 4.10.



(a) A bar plot of the BIC compared to the model order



(b) Zoomed version of the previous figure

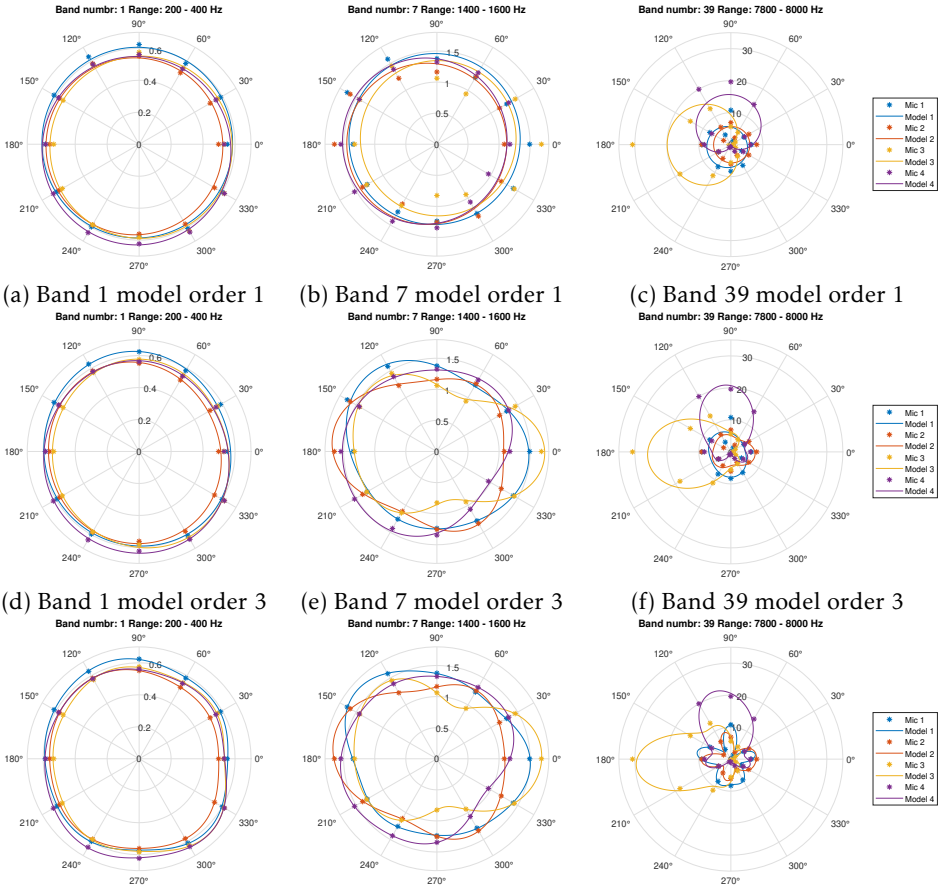
Figure 4.10: *(a) and (b) show the value of the BIC compared to the model order*

This analysis also had problems when using a high model order for the calibration. Going over a model order of 5 again caused the YALMIP solver to assign NaN values in the directional sensitivity model. Performance testing was done with model order 1, 3, and the maximum possible 5.

As with the previous results the sound used for training was also used to evaluate the estimation ability. The true measurements and the models built with different model order are shown in Figure 4.11. The maximum error and standard deviation are presented in Table 4.3, while the raw angle errors depending on angle can be seen in Figure 4.12.

Model Order	1	3	5
Standard deviation [°]	22.8470	7.6448	11.7895
Maximum error [°]	41.0400	15.1200	21.6000
Calculation time [s]	12	26	47

Table 4.3: *The performance data of the estimation for the different model orders when using prototype 2.*

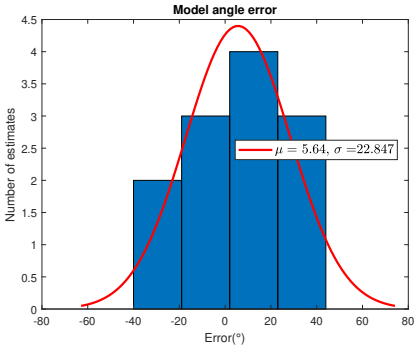


(a) Band 1 model order 1 (b) Band 7 model order 1 (c) Band 39 model order 1

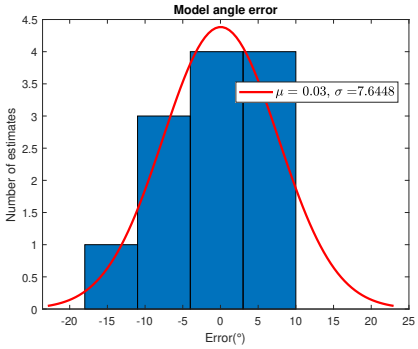
(d) Band 1 model order 3 (e) Band 7 model order 3 (f) Band 39 model order 3

(g) Band 1 model order 5 (h) Band 7 model order 5 (i) Band 39 model order 5

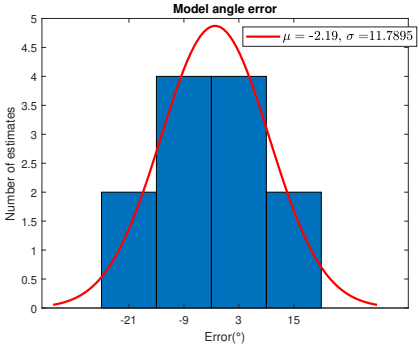
Figure 4.11: *The Fourier series approximation for different model orders and frequency bands when using prototype 2.*



(a) Model order 1



(b) Model order 3



(c) Model order 5

Figure 4.12: *Histogram of the magnitude of the angle errors when using prototype 2. A normal distribution has been fitted to the histogram.*

Some live testing with real gunshot was made to ascertain what real world performance could be expected. The shooter was around 20 meters from the recording device and to get the true angle a protractor attached to the device was used. The angle error histogram can be seen in Figure 4.14 and the performance results can be seen in Table 4.4. A spectrogram of one of the gunshots can be seen in Figure 4.13.

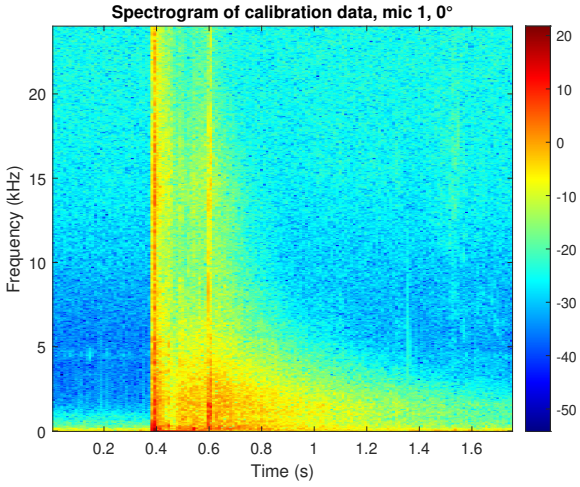
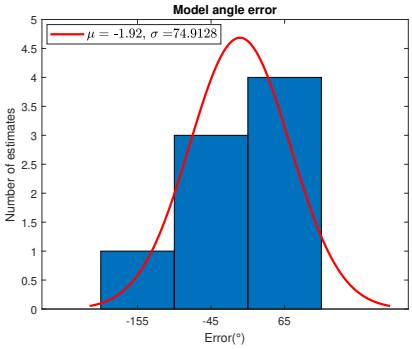


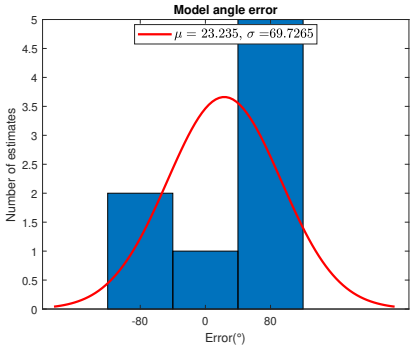
Figure 4.13: *The spectrogram of one of the gunshots used in the estimation. The second peak is an echo, likely from the embankment that catches the bullet.*

Model Order	1	3	5
Standard deviation [°]	74.9128	69.7265	66.7821
Maximum error [°]	144.6000	94.8000	93.6400

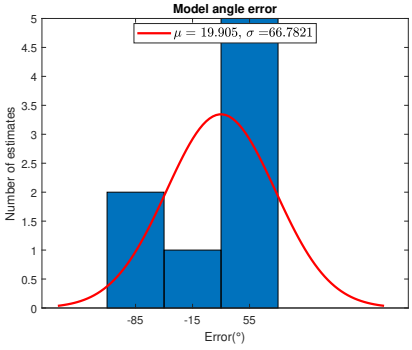
Table 4.4: *The performance data of the estimation to real gunshots for different model orders.*



(a) Model order 1



(b) Model order 3



(c) Model order 5

Figure 4.14: *Histogram of the magnitude of the angle errors to gunshots. A normal distribution has been fitted to the histogram.*

4.3 Discussion

4.3.1 Known good data

The calibration data supplied by our supervisor provided a good testing ground to tune the performance and verify functionality of the code. Further analysis gave some surprising results. While a low BIC indicates that the model is well optimized in terms of number of parameters and accurate to the testing data it has some limitations because of how it is currently implemented. Since then BIC is not tested against the complete summarized model used for the actual estimation but rather on each individual frequency band, it results in a model that is only optimized to follow these individual frequency bands. Overall the results show that a low BIC does not always correspond to a more accurate model when the bands are summed up and used for estimation.

4.3.2 Prototype 1

Calibration of prototype 1 proved difficult, both because of the environment where calibration data collection took place and because of the uneven behavior of the microphones. The recording was made in an empty and echoing room which caused the sound to bounce a lot and possibly interfere with itself. This is likely the explanation to the "double bubble" look to the directional sensitivity, i.e. specific echos that the device pick up way more sound from unexpected directions. The uneven placement of the microphones around the array may also have affected performance of the estimation since it left some directions with less overlap. Two of the microphones had a consistent intrinsic lower received power despite all microphones being of the same make and model as well as being set to the same gain. Completely removing the weakest microphone from the analysis, i.e. microphone number 3, would barely affect the estimate in most situations except for a few specific angles. This indicates that for the model to work optimally, the microphones gain should be roughly equal and that echos will adversely affect the estimation performance.

4.3.3 Prototype 2

Calibrating prototype 2 in the anechoic chamber made the model significantly more accurate, even if the uneven behavior of the microphones leaves a lot to be desired. With the best model order of 3 and a sound source 1000m away it would mean a standard deviation of 133m and a worst result of 263m. If this kind of performance can be replicated in real world conditions it would likely be sufficient in the purpose of finding a poacher on the savannah. Unfortunately, when estimating the direction to real gunshots the performance was not as good. With the best model order for these sounds of 5 and a gunshot 1000m away the standard deviation would be 1101 a worst result of 1459m. This result would not give sufficient accuracy to merit implementation on an anti-poaching device. However, since a DOA estimation in it self does not give a range estimate this inaccuracy may be less important if several estimates are fused. While testing

indicates that the method does indeed work to give estimates in the general direction of a gunshot, it might be just as accurate as guessing a direction solely based on what microphone had the highest received power. Gauging the direction to a gunshot with echoes and other noises that interfere with the signal is inherently difficult. It is also possible that, while the device was not disassembled between the gathering of calibration data and the gathering of gunshot data, some of the microphones may have shifted in their position, thus making the estimate worse. It should be noted that the gunshot data was collected few days before the calibration, making microphone performance drift an unlikely factor. The true angle to the shooter is also much more difficult to gauge with only a protractor with a radius of about 20 cm, something that may have lead to inaccurate angles. The spectrogram of the gunshot shows that the sound impulse is very short and has a very wide spectrum, with the highest intensity at low frequencies. Lower frequencies also travel further in the atmosphere, which makes the prospect of making estimations at longer distances good. However it is also at lower frequencies that the directional sensitivity is the least prominent.

There is a potential simple explanation to the poor performance however. When the shot was fired that that close to the device the sound is too loud for the microphones to measure, resulting in a "clipped" signal. This also means that the power is calculated incorrectly, resulting in the bad measurements. Of course, it is difficult to ascertain if this was indeed what caused the estimate to be inaccurate and to what degree.

4.3.4 Solver error

Exactly why the solver fails when the model order is too high for the two prototypes is unknown. However, a likely culprit may be because of the larger spacing between the calibration points that causes the solver to find some otherwise impossible Fourier series that nonetheless give a lower loss function value than a more reasonable and smooth model. What makes this more likely is that a higher increment in angle between the calibration points caused the calibration to fail earlier.

5

Conclusion

The estimation method can, in a controlled environment and with a good array, give excellent angle estimates. The step to translating this for use in practise where microphones need to be cheap and outside conditions cannot be controlled will need further work.

Shots can be differentiated from other sound. We found many models that managed to do so all better than a coin flip, and as such if more shots are fired there is a very good chance that at least one of the shots could be detected. Since the models were better than a coin flip the features managed to describe the data but some tuning of the features is necessary for good results. The largest problem with the solutions is the false positive rate and is what really hinders deployment of any of the detection models.

The microphones were limited to an audible range of about 1.5km when used in Swedish forests, even though the human ear could clearly hear gunshots from that range. An open plain would have the sound travel further.

The device was solid enough to withstand one night of heavy rainfall. However the microphones are a weak point where dust and water could potentially leak in if left outside for a long time. A method to shield these from the elements without compromising on detection and estimation performance is desirable.

5.1 Future work

A lot of the time dedicated to this project has be collecting data of guns being fired, this does not only include the physical gathering but an equally time con-

suming part has been talking to potential people that could allow us to be present while they shoot. We have found that military personnel has by far been the most helpful with this and if more data is to be collected these are the people to turn to first.

5.1.1 Detection

To improve the detection in the future the first step is always to improve the data available to the models. Specifically more positive data would be required. There is plenty of negative data so initially more positive data would be required but different negative data would also serve the project. The best kind of data to gather would be sounds of shot gathered in Kenya specifically on the savannah, as it stands all the data is gathered in Swedish forests or Swedish hilly forest. One intermediate step is to gather gun shot data in Sweden but that at least is on a field or at the very least where there is a line of sight to the shooter. Another way to gather different gun shots would be to gather data where the shock wave from the bullet is present. To see if it effects the detection. If it is not possible to do any of these more data from Swedish forests would still improve the performance.

There is a need to verify that shots sounds traveling through forest do not significantly alter the characteristic of a shot. If this is the case, all the positive data in the data set would need to be replaced. The easiest way to test is to gather some shot data from a field in Sweden or better yet on the savannah in Kenya, then send this data to a model trained on the existing data set and analyze the performance. If the performance is significantly worse some it could indicate that some important characteristic of the sound is attenuated when going through forest. Another approach would be to do some kind of principal component analysis and check if the different shot files are similar.

Even though this thesis finds that the neural net solution performed worse than feature based methods we still believe it has the most potential. A neural net solution is easier to scale with more layers, and then do residual layers or batch normalization. The field of neural net image classification is wast and constantly growing providing room to improve the model, especially if more processing power is available. Feature based solutions require the construction of better features to be improved. Identifying and implementing better features is time consuming and require a lot of knowledge about the specific problem to be solved. If this route is taken it is expected that naive Bayes would not remain the best model, but that it would be overtaken by SVM. Although testing all approaches to feature based methods is probably wise.

If for what ever reason a neural network solution is not desired the current models could be improved with some parameter tuning. Very little tuning has been performed during this project as having better data and better features will improve the models a lot more and therefore time was placed there. The models performance could probably be improved if the features were tuned or if hyper

parameter optimization was performed. These things would not improve the model greatly but some improvement could probably be gained.

5.1.2 Estimation

While the training phase of the calibration is fairly quick, collecting data takes a long time. A standardized testing protocol would speed up the calibration data collection process and allow for more precise models and perhaps models tuned to different frequency spectrums. Such a protocol should contain information on how the device should be mounted, how the angles are accurately measured to the sound source, what sounds should be played and how the recordings are saved.

In future implementations the microphone array should be constructed so that the microphones do not need to be moved from their mounting position other that to replace them. This is because the calibration relies on the properties of the microphones being constant, and even slight movements to them in relation to each other will have an adverse effect on the DOA estimate.

With how the code works at the moment, it only gives a DOA estimate and no other information. Calculating several estimates gives a general idea of how accurate the model is in general, but it would be desirable to have the DOA estimate to be given along with a confidence interval, i.e. how sure the model is of the estimate. For example, let's say a sound is coming from a true angle of 40° and the model gives an estimate of 42°. While this is a good estimate from what is known about the true angle, this gives no information on how confident the model is of the estimate. A confidence interval of 3° in either direction let the user know that the estimate is likely to be accurate, whereas a confidence interval of 30° may mean that the estimate isn't as useful. Even if, as in this example, the ground truth angle was not far from the estimate. What the confidence interval is for any given sound is likely to vary depending on the true angle from which the sound came and the strength of the sound.

Currently the usage of higher model orders is rather inefficient. As show in the results in Section 4 higher model order is not necessary for representing lower frequencies. A dynamic way to assign lower model order to simpler frequency bands, and higher model order for more complex ones is an excellent way to save on resources, both for optimisation time and number of parameters. For this BIC could be used, i.e. each band is tested individually with Equation 4.9 for the best model order and then saves this to a new vector.

5.1.3 Localization

Localization by fusing multiple DOA estimates can give the exact location of a fired shot. It is also theoretically possible to use the *received signal strength* (RSS) to get a distance estimate together with the DOA estimate which would allow for

location estimation with only a single detection. This future expansion would make the finding of the poacher much easier and the theory for doing this.

Triangulation

If the devices are close enough, so that multiple devices can hear the sounds, several DOA estimations can be combined to locate the source. This can be done with triangulation, an illustration which can be seen in Figure 5.1. Methods for fusing DOA estimates can be found in books such as *Statistical Sensor Fusion* by Fredrik Gustafsson [11].

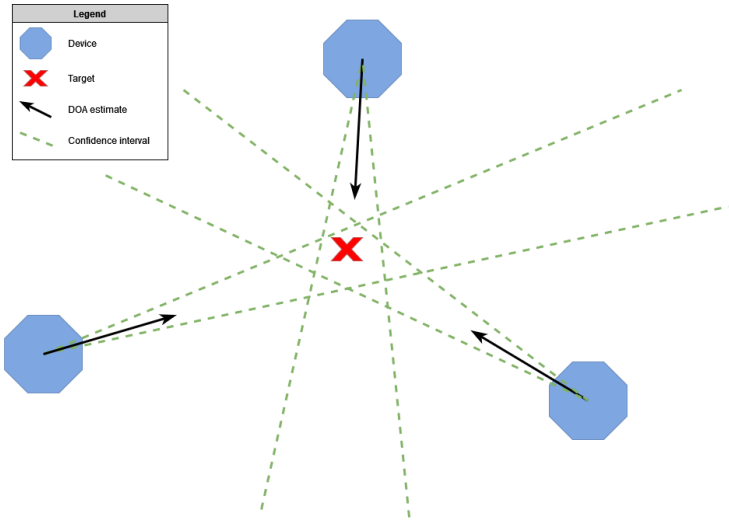


Figure 5.1: *How the stations can work together to find the location of the shooter.*

Distance estimation

If no more than one detection is made then triangulation cannot be performed. In this case, a distance estimate could be made with an RSS method. Since all sounds have a predictable falloff in intensity and together with a rough estimate of how loud the gun was, received signal strength can be calculated. That together with the DOA estimate will provide a coarse estimate to the position of the gunman. An illustration can be seen in Figure 5.2.

Some models such as ISO 9613-2 have been created to predict loudness of a sound a from a distance based upon strength at emission, atmospheric conditions and

obstacles. This could then used to create a loss function to estimate the distance. However, doing so would require good knowledge of the current conditions as well as the type of rifle being discharged. The former can be solved with weather predictions or weather stations feeding data to the model, while the latter will either have to be a complete guess or attempting to classify the type of weapon used during the detection step.

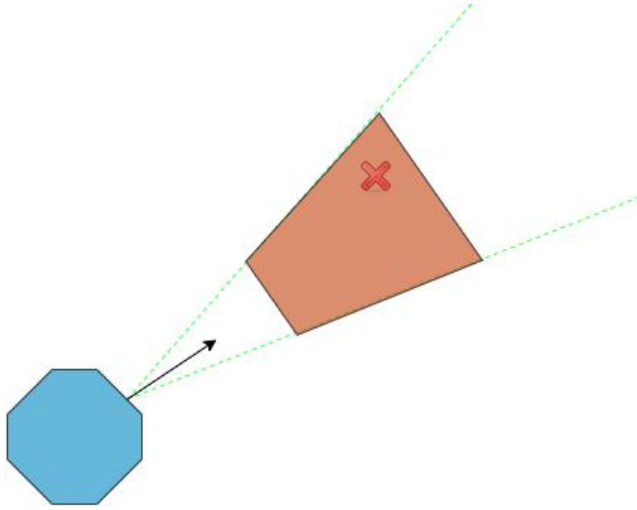


Figure 5.2: *How a single station can estimate the position of the shooter*

Bibliography

[1] Africa's poaching crisis. URL https://campaign.awf.org/poaching-infographic/.

[2] Poaching numbers | conservation | save the rhino international. URL https://www.savetherhino.org/rhino-info/poaching-stats/.

[3] Safereaction. URL https://safereaction.com/.

[4] Hervé Abdi and Lynne J. Williams. Principal component analysis. *WIREs Computational Statistics*, 2(4):433–459, 2010. doi: https://doi.org/10.1002/wics.101. URL https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.101.

[5] L. Breiman. Random forests. *Mach Learning*, pages 5–32, 2001.

[6] R. Emslie. Diceros bicornis, Black Rhino. *The IUCN Red List of Threatened Species*, 2020. doi: https://dx.doi.org/10.2305/IUCN.UK.2020-1.RLTS.T6557A152728945.en.

[7] R. Emslie. Ceratotherium simum, White Rhino. *The IUCN Red List of Threatened Species*, 2020. doi: https://dx.doi.org/10.2305/IUCN.UK.2020-1.RLTS.T4185A45813880.en.

[8] Julian Blanc Iain Douglas-Hamilton Patrick Omondif Kenneth P. Burnham George Wittemyer, Joseph M. Northrupa. Illegal killing for ivory drives global decline in african elephants. *Proceedings of the National Academy of Sciences*, 2014. doi: 10.1073/pnas.1403984111.

[9] Edwards C.T.T Maisels F. Wittemyer G. Balfour-D. Taylor R.D. Gobush, K.S. Loxodonta cyclotis, African Forest Elephant. *The IUCN Red List of Threatened Species*, 2021. doi: https://dx.doi.org/10.2305/IUCN.UK.2021-1.RLTS.T181007989A204404464.en.

[10] Edwards C.T.T Maisels F. Wittemyer G. Balfour-D. Taylor R.D. Gobush, K.S. Loxodonta africana, African Savanna Elephant. *The IUCN Red List*

of Threatened Species, 2021. doi: https://dx.doi.org/10.2305/IUCN.UK.2022-2.RLTS.T181008073A223031019.en.

[11] Fredrik Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur AB, third edition. ISBN 978-91-44-12724-8.

[12] Fredrik Gustafsson, Gustaf Hendeby, David Lindgren, George Mathai, and Hans Habberstad. Direction of arrival estimation in sensor arrays using local series expansion of the received signal. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 761–766, 2015.

[13] Michael G. Haag and Lucien C. Haag. Chapter 17 - the sound levels of gunshots, supersonic bullets, and other impulse sounds. In Michael G. Haag and Lucien C. Haag, editors, *Shooting Incident Reconstruction (Third Edition)*, pages 407–444. Academic Press, San Diego, third edition edition, 2021. ISBN 978-0-12-819397-6. doi: https://doi.org/10.1016/B978-0-12-819397-6.00017-9. URL `https://www.sciencedirect.com/science/article/pii/B9780128193976000179`.

[14] P Hart. The condensed nearest neighbour rule. *IEEE Transactions on Information Theory*, 14:515–516, 1968.

[15] Huan Liu Hiroshi Motoda. Feature selection extraction and construction. *Towards the Foundation of Data Mining Workshop, Sixth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2002), Taipei, Taiwan, pp. 67–72*, 2002.

[16] S. B. Kotsiantis. Supervised machine learning: A review of classification techniques. *Informatica 31 (2007) 249-268*, 2007.

[17] Lee Sangseok Lee Dong Kyu, In Junyong. Standard deviation and standard error of the mean. *kja*, 68(3):220–223, 2015. doi: 10.4097/kjae.2015.68.3.220. URL `http://www.e-sciencecentral.org/articles/?scid=1156109`.

[18] David Lindgren, Olof Wilsson, Fredrik Gustafsson, and Hans Habberstad. Shooter localization in wireless microphone networks. *EURASIP Journal on Advances in Signal Processing*, 2010(1):1 – 11, 2010. ISSN 1687-6180. URL `https://login.e.bibl.liu.se/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,uid&db=edssjs&AN=edssjs.5FC20B75&lang=sv&site=eds-live&scope=site`.

[19] Lennart Ljung, Torkel Glad, and Anders Hansson. *Modeling and Identification of Dynamic Systems*. Studentlitteratur AB, second edition. ISBN 978-91-44-15345-2.

[20] T.Deepa L.Ladha. Feature selection methods and algorithms. *International Journal on Computer Science and Engineering (IJCSE), vol.3(5), pp. 1787-1797*, 2011.

[21] J. Löfberg. Yalmip : A toolbox for modeling and optimization in matlab. In *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[22] Beth Logan. Mel frequency cepstral coefficients for music modeling. *In Proceedings of International Symposium on Music Information Retrieval (ISMIR)*, 2000.

[23] Abiodun Esther Omolara Kemi Victoria Dada Nachaat AbdElatif Mohamed Humaira Arshad Oludare Isaac Abiodun, Aman Jantan. State-of-the-art in artificial neural network applications: A survey. *Heliyon, volume 4, issue 11*, 2018.

[24] Radu Mariescu-Istodor Pasi Fränti. Soft precision and recall. *Elsevier B.V*, 2023.

[25] Julian Blanc Carsten F. Dormann Colin M. Beale Severin Hauenstein, Mrigesh Kshatriya. African elephant poaching rates correlate with local poverty, national corruption and global ivory price. *Nature Communications 10 (1)*, 2019. doi: 10.1038/s41467-019-09993-2.

[26] Small Arms Survey. *Small Arms Survey 2015: Weapons and the World*. Cambridge University Press, 2015. ISBN 9781107323636.

[27] K.M. Ting. Confusion matrix. *Encyclopedia of Machine Learning*, 2011.

[28] Lennart Ljung Torkel Glad. *Reglerteori - Flervariabla och olinjära metoder*. Studentlitteratur AB, second edition. ISBN 978-91-44-03003-6.

[29] R.W. Wall. Simple methods for detecting zero crossing. 2003.

[30] Jan Jongboom Zach Shelby. Edge impulse, 2019. URL `https://docs.edgeimpulse.comhttps://docs.edgeimpulse.com/docs/`.

[31] Gustav Zetterqvist, Fredrik Gustafsson, and Gustaf Hendeby. Using received power in microphone arrays to estimate direction of arrival. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10097197.