A Sound Processing Pipeline for Robust Feature Extraction to Detect Elephant Rumbles

M. B. C. K. SILVA
Index No: 13001132

Supervisor: Dr. C. I. Keppitiyagama
Co-supervisor: Mr. A. Sayakkara

December 2018

Submitted in partial fulfillment of the requirements of the
B.Sc in Computer Science Final Year Project (SCS4124)

Declaration

I certify that this dissertation does not incorporate, without acknowledgment, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, be made available for photocopying and for inter-library loans, and for the title and abstract to be made available to outside organizations.

Candidate Name: Mr. M. B. C. K. SILVA

Signature of Candidate Date:

This is to certify that this dissertation is based on the work of
Mr. M. B. C. K. SILVA
under my supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

Supervisor Name: Dr. C.I. Keppitiyagama

Signature of Supervisor Date:

Abstract

A significant number of human and elephant lives have been lost due to the human-elephant conflict in Sri Lanka. To save lives of humans and elephants, it is therefore essential to minimize encounters between them. An early warning system, which detects and localize the presence of elephants through their infrasonic emissions is a viable solution to mitigate such conflicts. The high cost of infrasonic detectors is an important challenge to the real-world deployment of such localization systems, in particular in developing countries where the human-elephant conflict occurs. ElOC is a system developed as a part of inventing a low-cost automatic elephant detection and localization system. Which is capable of localizing the infrasonic source within a ten-meter accuracy.

In this dissertation, a novel approach is proposed to extend the ELOC to identify the elephant infrasound automatically. The novelty of this approach is the capability of distinguishing the infrasonic emissions from the elephant on top of the low-cost, resource-limited hardware platform of the ELOC. The approach first applies a sequence of operations to reduce the effect of noise contained in the infrasonic signal captured by the ELOC node. Then the spectral features of the infrasonic signal are extracted with wavelet-based signal reconstruction to analyze the signal more precisely. Finally, the extracted features are feed to the pre-trained classifier to distinguish the infrasound emissions from the elephants.

This study is able to classify elephant rumbles with an accuracy of 82%. Thereby the proposed approach exhibits promising results in elephant detection and capable of operating on the resource-limited hardware platform of the ELOC. This study also contributes to the domain of digital signal processing since the study is the first attempt of wavelet-based feature extraction in the domain of infrasound elephant rumble detection.

Preface

The project was undertaken at the request of Mr. T. P. Ranathunga with my passion for making a world better place. My research question was formulated together with my supervisor, Dr. Chamath Keppetiyagama. The outcome of this research will help reduce the human-elephant conflicts in the world and it will be an excellent course for both humans and elephants.

Some of the concepts identified in the literature review has been used in the context and those are justified and referenced accordingly. Sample data was collected from the Smithsonian conservation biology institute. I carried out the analytical calculation in the results and evaluation chapter under the supervision of my supervisor.

Acknowledgement

This thesis is the result of not only my hard work during the fourth year of my degree but the result of four long and hard years. I would be lying if I said I had no help in completing this research and I take this chance to thank them from the bottom of my heart.

I would like to give my thanks to my supervisor Dr.Chamath Keppitiyagama for being the mentor and the person who made me interested in research. your patience and kindness is much appreciated and valued.

My special thanks go to Mr. Asanka Sayakkara, Dr. Prabhash Kumarasinghe Mr. Namal Jayasooriya, and Mr. Charith Madushanka for their continuous supports and advise throughout the research period. I kindly thank them for the support and valuable insights given for many aspects of the research. I am also thankful for the SCORE lab for letting me to conduct experiments.

Contents

List of Figures

List of Tables

Acronyms

AWGN Additive White Gaussian Noise.

CWT Continuous Wavelet Transform.

DWT Discrete Wavelet Transform.

FT Fourier Transform.

GPS Global Positioning System.

HEC Human Elephant Conflict.

IDWT Inverse Discrete Wavelet Transformation.

MFC Mel Frequency Cepstrum.

MFCC Mel Frequency Cepstral Coefficients.

RMS Root Mean Square.

SNR Signal to Noise Ratio.

STFT Short Time Fourier Transform.

SVM Support Vector Machine.

TDOA Time Difference of Arrival.

WGN White Gaussian Noise.

WPT Wavelet Packet Transform.

WT Wavelet Transform.

Chapter 1

Introduction

Statistics have shown that the world elephant population has dramatically declined over the last few decades. At the commencement of the 20th century there were an estimated 200 000 Asian elephants, but at present, there are probably no more than 35 000 to 40 000 elephants left in the wild [1]. Human Elephant Conflict (HEC) is one of the major threats to wild elephants, which has been caused by habitat loss and fragmentation. In Asia, most of the areas that form elephant habitat are in proximity to human settlements. Therefore, some conflicts often emerge between animals foraging for food and the local human population [2]. Each year, over 50 people in Sri Lanka are killed by elephants and over 100 elephants are killed by farmers.

Several traditional approaches have been taken to overcome the HEC such as lighting fires, firecrackers, and electric fence systems. The most common approach to reducing the HEC is the electric fence system, which is an expensive and potentially life-threatening solution. Recently, technological advancements have led to the use of radio and Global Positioning System (GPS) collars to detect the presence of elephants [3]. However, these techniques are expensive and suffer from various drawbacks such as non-availability of resources and operational difficulties. Therefore 3rd world countries like Sri Lanka and India are suffering from the inadequacy of cost-effective and efficient solutions.

Thus, the ultimate purpose of this research is to implement a sustainable adaptive sensor-actuator system for elephant detection and localization using infrasound. The successful application of this research aims to mitigate the human-elephant conflict in developing countries like Sri Lanka, with a set of low-cost devices.

1.1 Background to the Research

1.1.1 Elephant Call Types

Elephants use vocalizations for both long-range and short-range communication. They produce a broad range of sounds from very low frequency to higher frequency. Moreover, these sounds can be classified into different call type based on their physical properties [4]. Mainly into four basic types such as Trumpet, Roar, Chirp, and Rumble (Asian elephants) [5, 6]. Spectral and temporal characteristics of these four call types are similar in Asian and African elephants. The call type and its frequency range (including harmonics) used by elephants is given in Table 1.1.

Table 1.1: Different call types and frequency range

Call Type	Frequency range
Trumpet	405-5879
Roar	305-6150
Chirp (Squeak/Squeal)	313-3370
Rumble	10-173

Elephants use these calls in the context of play, disturbances, presence of other species or vehicles and interactions within herds. To communicate between family herds over long distances, they produce low-frequency vocalizations known as rumbles [5].

A typical African male elephant rumble fluctuates around a minimum of 12 Hz, a females rumble at around 13 Hz [7]. In Asian elephants, these values fluctuate between 14 Hz to 24 Hz within 1 to 25 seconds [5] due to their smaller vocal cords compared to African elephants. This research is focused on detecting elephants through their rumbles.

1.1.2 Infrasound

Sound is a pressure wave which is created by a vibrating object. These vibrations set particles in the surrounding medium (typically air) in vibrational motion, thus transporting energy through the medium. The frequency of a sound is the number of times per second, that sound wave cycles from positive to negative and positive to again (measured by Hz). Humans have a range of hearing from 20 Hz (low) to 20,000 Hz (high). Any Frequencies beyond this range are inaudible to humans. Sound waves below the frequencies of audible sound refer as the infrasound and nominally includes anything under 20 Hz. According to [7] and [5], elephant rumbles can be considered as infrasound.

When sound propagates through a medium, sound is lost by heating the medium through which it propagates. Acoustic attenuation is a measure of the power loss when sound propagates through the media, and on most media this attenuation is frequency dependent. Figure 1.1 represent the sound absorption coefficient per atmosphere for air at 20 degree Celsius according to relative humidity per atmosphere [8].

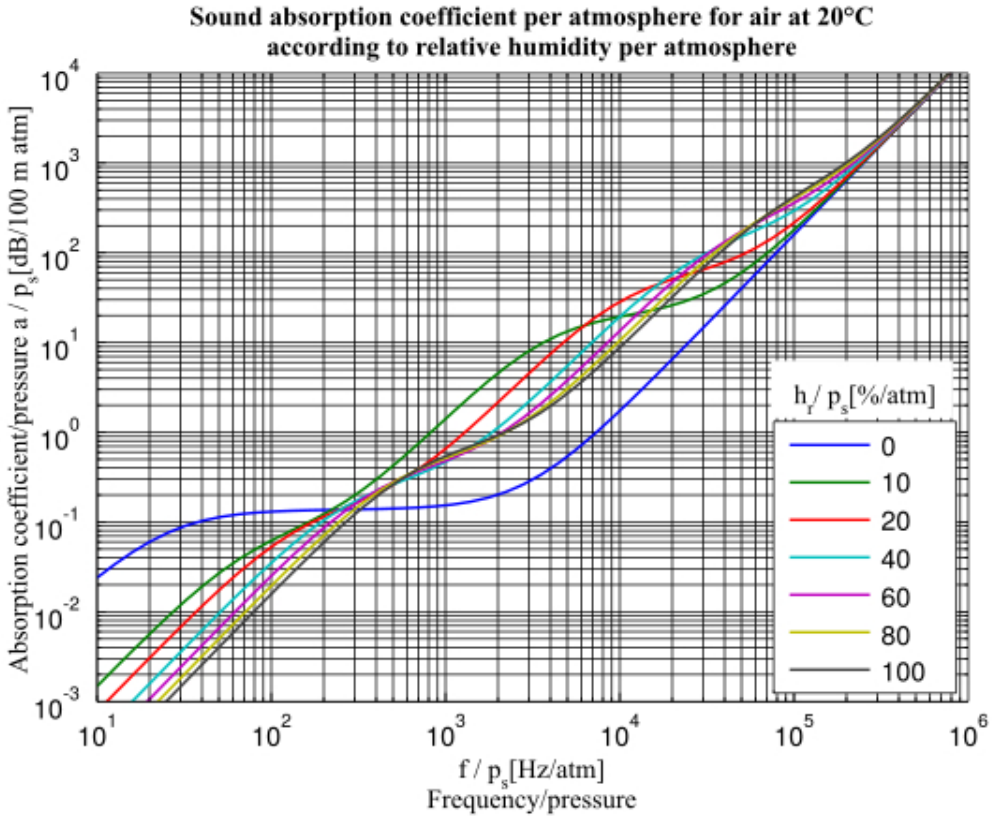

Figure 1.1: Absorption coefficient per different frequencies

In [9] it shows that infrasound has low attenuation coefficient than the high-frequency sounds. Therefore, infrasound can travel a greater distance from the source of the sound. The infrasound used by elephants for long-range communication can travel a distance that exceeds 1 km [10]. That is the primary reason for selecting infrasonic elephant rumble to detect the presence of the elephant from long distance. (Which will be further emphasized in Section 2.2).

1.1.3 Infrasound Detectors

Most of the commercially available infrasound detectors are made for detecting geographical events such as earthquakes and volcanic eruptions. Early works have shown [11] that these devices are capable of capturing infrasound elephant rumbles. However, these devices are very costly, and Infitec Infra 20 can be considered as the relatively low-cost device

compared to the other existing devices, and this also costs around US$345 [12].

ELOC node is a low-cost infrasonic detector proposed by Sayakkara et al. [13] as a part of a complete low-cost system for elephant detection and localization. The device consists of two microphones and a single board computer. Moreover, it is more than 90% cheaper than the currently available infrasonic detectors in the market. Therefore, it is suitable for a developing country like Sri Lanka where the lower cost for deployment and maintenance of a system is the key to long-term sustainability.

1.2 Research Problem and Research Question

1.2.1 Research Problem

Although ELOC node is economically sustainable infrasonic detector. Since it has been developed with the low-cost hardware components, it has limited computation power and less sensitivity than the commercially available infrasonic detectors. Ranathunga et al. [14] proposed a method to extend the ELOC node as an automatic elephant detection system. However, the proposed method, is not efficient to run on the ELOC node (Further emphasized in Section 2.2.1).

1.2.2 Research question

How to extend the ELOC node as an automatic elephant detection system?

1.3 Research Aim and Objectives

The main aim of the research is to propose an approach to detecting elephant rumbles captured by the ELOC node as a part of long-term attempt to minimize human-elephant conflicts by early detection of elephants near human habitats.

The objective of the research are as follows

- Analyze the elephant detection method proposed by Ranathunga et al. [14] and determine whether it can be enhanced to work on top of the ELOC node.
- Determine the computation power and sensitivity sufficient enough to extend ELOC node as an automatic elephant detection system.

- Identify the possible mechanism to mitigate the adverse effect of noise to the elephant detection.
- Formulate a mechanism to extend ELOC node as an automatic elephant detection system.

1.4 Justification for the Research

As described in Section 1.2 the research is part of a complete system for elephant detection. The final goal is a system with various inputs including automatic elephant detection and localization. The localization system has already been completed by Sayakkara et al. [13]. Thus, this research focuses on proposing a methodology for automatic elephant detection which is an essential part to achieving complete elephant detection system. "Another subsystem is an elephant behavior model which can estimate where the elephants would be. Elephants tend to have a regular timetable in which they move from the forest to the watering hole and back. A map of the terrain could provide information about the forest areas and the availability of lakes inside, e.g., a national park. The idea is to take all these inputs and then use a particle filter to fuse the inputs into a system that presents elephant locations together with estimates of the probability that there is an elephant at the given location." [13]

Furthermore, this research is not only focused on automatic detection of the elephant but also focuses on mitigating the effect of noise and attempts to introduce a computationally efficient methodology which will be able to run on the resource-limited hardware platform which is ELOC node.

1.5 Methodology

The proposed research methodology incorporates 4 main modules: Pre-processing module, Feature extraction module, Feature selection module, and Classifier. Figure 1.2 represents the high-level diagram of the proposed methodology.
the datasets mentioned in Figure 1.2 are defined as follows

- **Dataset-1** : Replayed elephant rumbles and negative dataset captured using ELOC node.
- **Dataset-2** : A collection of elephant rumbles from Smithsonian conservation biology

institute and negative dataset (infrasound recordings) taken from the internet.

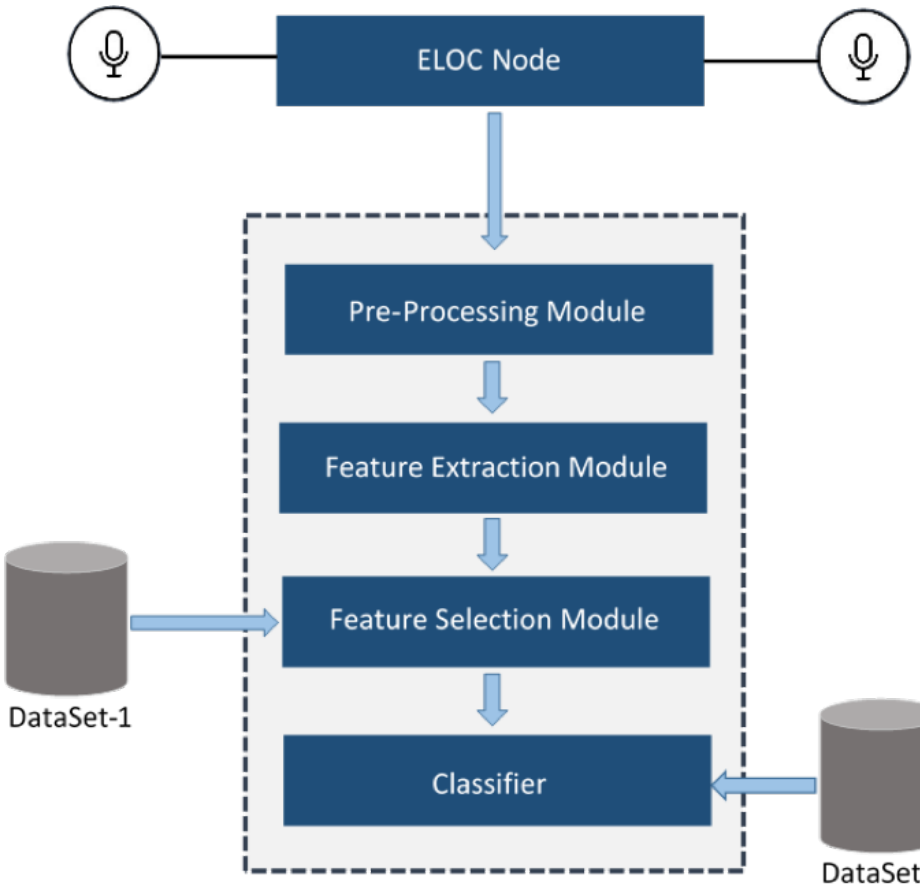

Figure 1.2: Proposed research methodology

Initially, an infrasonic sound signal is captured by the two microphones in the ELOC. After capturing the 2-second length audio signal, it will pass it to the pre-processing module. This module consists of 3 sub-modules; Butterworth band-pass filter mitigates the influence of the low-frequency noise and high-frequency noise that exists beyond the typical range of elephant rumbles. Wavelet-based denoising further denoise the input signal and beamforming based stereo to mono conversion, convert the stereo infrasound recoding into the signal channel while improves the strength of the dominant sound source and reducing the strength of noise coming from the other directions.

Secondly, the processed signal passes into the feature extraction module. And it consists of 3 sub-modules, and it extracts the 672 features from the given signal as well as wavelet-based reconstructed variations of that signal to analyse the signal more precisely. After that, extracted feature vector passes into the feature reduction module which consists of two pre-trained sub-modules. First, sub-module brings the feature vectors into normalized scale while minimizing the effects of outliers. Normalized feature vector then passes into second

sub-module, and it returns the pre-defined most robust and relevant subset of features from the given feature vector.

Finally, reduced feature vector passes into the pre-trained classifier. The classifier will determine whether given signal segment consists of infrasonic elephant rumble or not.

1.6 Outline of the Dissertation

The dissertation is organized as follows. Chapter 2 examines the existing approaches related to the domain of infrasound elephant detection and present the theories and methods associated with the problem area. Chapter 3 describes the proposed research design and methodology. Chapter 4 describes the implementation details of the proposed methodology. Chapter 5 presents the evaluation results of the proposed approach and justified assumption made during the proposed research design. The closing chapter, Chapter 6 expresses the conclusion of the thesis and outlines the future work.

1.7 Delimitation of Scope

Since this research mainly focuses on detection of Asian elephant calls, an elephant call recording dataset by Smithsonian Biology Research group will be used for training and evaluation of the module. The dataset consists of different elephant call recordings captured in Sri Lanka and this dataset has been annotated by a biology expert.

The accuracy of the proposed system will be evaluated by replacing the elephant rumbles by a subwoofer. Earlier work has shown [11] that subwoofers can replay elephant sounds that include fundamental frequency components in the infrasonic range with sufficient output power to emulate a real elephant. The influence of the different noise type will be tested by mixing these noises programmatically under different Signal to Noise Ratio (SNR).

The proposed approach is targeting on developing automatic elephant detection system on top of the ELOC node. Therefore, proposed approach will not consider the other available infrasonic detectors.

1.8 Conclusion

This chapter placed the foundations for the dissertation. It introduced the research problem, research questions, and aim and objectives. Then the research was justified, the

methodology was summarized and justified, the dissertation was outlined, and the limitations were given. On these grounds, the dissertation can proceed with a comprehensive description of the research.

Chapter 2

Literature Review

2.1 Introduction

This section, provides a review of work that relates to this study. Section 2.2 discusses the previous research and results that lay the foundation for the work in this research. Section 2.3 presents the essential concepts related to the proposed research approach.

2.2 Related works

In recent times, there have been many types of research that addresses the detection of the presence of elephants using the sounds (calls) of the elephants.

Payne et al. [10] discovered that elephants use infrasound to communicate. Further work with William Langbauer et al. have shown that the elephants were indeed making infrasonic calls. The large vocal cords of the elephants can produce low-frequency sound signals known as rumbles, which can travel up to 6 kilometers in distances. Moreover, they discovered that elephant rumbles are audible to humans because of harmonic waves created from the infrasonic fundamentals. These findings can be considered as the basis for all further research in this particular area.

Dissanayake et al. [15] proposed a method to localize the elephant using a sensor network and quantitatively analyze the effects of various environmental parameters on the positioning accuracy of the system. They found that the accuracy of elephant localization and detection significantly depends on the behavioral sound propagation factors in the air. Accordingly, to the conclusion of the study, if the temperature is uniform over the subjected area, the impact of the temperature is insignificant for the accuracy of source localization. In contrast, the

temperature variation among each source sensor path will cause an effect positioning error significantly. Moreover, SNR significantly impacts the localization accuracy.

Mayilvaganan et al. [16] proposed an approach for elephant localization and direction of arrival estimation using acoustic sensor network based on the error rectification method. They use two microphone sensor array networks, each array with 0.5-meter distances. In the results section, they have mentioned that this system can detect the arrival of the elephant with 96% accuracy with real-time data.

Limitation of this research is that they have not mentioned how to identify elephant calls among other sound sources. They have also concluded that factors such as wind and temperature affect the accuracy of the elephant localization while SNR plays a significant role in localization accuracy.

Zeppelzauer et al. [17] proposed a method for automated detection of elephant vocalizations which is targeted towards a future automated early warning system for elephants. They introduced a novel technique for signal enhancement to improve the robustness of the detector in noisy situations. Even though this novel approach enhances the performance of the detector in a noisy environment, they did not mention what type of sensor that they have used for the study. They have considered a broad range of low-frequency elephants sounds from 0 Hz to 500 Hz, which includes infrasound. Even though they were able to detect elephant vocalizations, detecting them at longer distances where higher harmonics are highly attenuated is problematic. According to the evaluation result, for the application of the proposed method as an early warning system is not reliable since it exhibits high false positive rate. Moreover, they have not mentioned the computation cost and real-time capability of the proposed approach.

Venter et al. [18] proposed another method for automated detection of infrasound elephant call. However, the accuracy of this method also depends on the availability of higher harmonics.

Although different research exists with relevance to the acoustic detection of elephants using infrasound waves, most of them are done for African elephants. There is no considerable research in existence for the Asian elephant. Moreover, there is no significant attempt towards an economically feasible solution for detecting elephants through the use of infrasound calls, especially in a noisy environment.

2.2.1 Works towards Low-Cost Elephant Detection and Localization System

Dabare et al. [11] have done a feasibility study for detecting infrasonic elephant rumbles using a low-cost INFRA-20 device. The cost of the device is approximately US$ 345.00 which is comparatively cheaper than the other infrasonic detectors available in the market for the same purpose. They tested the strength of proposed device by playing low-frequency sound using a subwoofer and measure the power of captured signal using the proposed device from a different distance. According to experiment results, they have inferred the ability to detect infrasonic waves with a low-cost hardware platform.

Sayakkara et al. [13] have proposed a novel approach to locating wild elephants using Low-cost Infrasonic Detectors. They introduced a new hardware node call ELOC node which consists of two Panasonic WM-61A omnidirectional back electret condenser microphone and single board computer. This node cost around US$ 73.00 which is significantly cheaper than the other infrasonic detectors available in the market. This node is more sensitive than the INFRA-20 device in infrasonic range. Furthermore, proposed device is sensitive enough to capture higher frequencies above the infrasonic range.

As the second part of the research, they introduced an infrasound localization mechanism. The proposed approach first calculates sound arrival time difference between two microphones by cross-correlating the audio signal captured by two microphones. Then use Time Difference of Arrival (TDOA) based technique to locate infrasonic sources. Based on test experiment carried out, they have concluded that proposed methodology is reliable to locate the source of the sound coming from an angle in the range of 30 to 90 to the device.

By extending the findings of Sayakkara et al. [13], Ranathunga et al. [14] proposed a method to Elephant detection and localization using infrasound. This method consists of two main sections, Acoustic detection of Elephants and sound localization.

2.2.1.1 Acoustic Detection of Elephants

For acoustic detection of elephants, they have used a support vector machine for distinguishing the rumbles and modified Mel Frequency Cepstral Coefficients (MFCC) for feature extraction which is originally designed to process human vocals. According to the evaluation, this approach exhibits notable performance as shown in Table 2.1 they further enhance the model by applying signal enhancement method proposed in [17]. Table 2.2 represents the enhanced performance.

Table 2.1: Detector Performance

Performance measure	Percentage
True negative	80.96
False positive	10.04
True positive	89.41
False negative	10.58

Table 2.2: Detector performance after signal enhancement

Performance measure	Percentage
True negative	95.78
False positive	4.22
True positive	93.65
False negative	4.23

The following observation has been made by critically analyzing the above approach.

1. Conflicting performance enhancement - As shown in Table 2.2, the false positive rate is considerably reduced by applying method proposed by Zeppelzauer et al. [17]. However, in the conclusion of used enhancement method, it is mentioned that it has a too high false positive rate.

2. Testing dataset is not fair - recording sample rate, bit depth, and infrasound capturing capabilities of the recording device of the negative data set is not mentioned.

2.2.1.2 Elephant Localization

They use same sound source localization approach which is proposed by Sayakkara et al. [13] to localize the elephant using ELOC node. To calculate sound arrival time difference between two microphones, slide one sound signal along the time axis until maximizing the cross-correlation between two sound signals captured by ELOC node.

2.2.1.3 Limitation of the approach

1. The maximum sampling frequency which can be captured using ELOC node is 11000Hz. The approach is most effective under the 48000Hz sampling rate and accuracy significantly reduces when the sampling rate is below the 48000Hz (further emphasized in the Section 5).

2. When calculating time difference, the whole sound captured by ELOC node is considered. If there is another significant sound source other than the interested elephant

(another elephant or any other low-frequency sound source), results in the wrong inter-arrival time calculation. Eventually, this will lead to wrong angle calculation.

Due to above limitations, the approach proposed by the Ranathunga et al. [14] cannot be used on top of the ELOC node to achieving the ultimate goal of the low-cost elephant detection and localization system.

2.3 Theories and Concepts

2.3.1 Noise Types

Noise is an unwanted signal that conflicts with the communication or measurement of another signal. Based on nature and origins, noise can be classified into several categories [19] such as

1. Electronic noise such as shot noise and thermal noise.
2. Acoustic noise, originating from moving, vibrating or colliding sources such as revolving machines, moving vehicles, keyboard clicks, the wind, and rain.
3. Electromagnetic noise.
4. Electrostatic noise created by the presence of a voltage.
5. Communication channel distortion, fading quantization noise and lost data packets due to network congestion.

Moreover, based on its spectrum, pulse or time characteristics, a noise process is further classified into several categories such as white noise, band-limited white noise and impulsive noise [19]. Electronic noise, acoustic noise, and electromagnetic noise are the leading type of noise applicable to this study.

2.3.2 Wavelet Analysis

Wavelets analysis is a mathematical method that divides data into different frequency sub-bands, and then analyze each sub-bands with a resolution matched to its scale. Recently, wavelet transforms have been used in various domains of speech processing such as speech recognition, pitch detection, and speech denoising and enhancement [19].

Followings are the some of the main advantages of Wavelet theory [20].

- Wavelet offers a simultaneous localization in time and frequency domain.
- Able to extract the fine details in a signal
- Able to denoise a signal without significant degradation.
- Computationally efficien

Fourier Transform (FT) has a critical drawback which is, the time information is lost when transforming to frequency information. In other words, when looking at an FT of a signal, it is impossible to recognize when a particular event occurs [21]. But, this drawback is not crucial for the stationary signal (signal properties do not change significantly over time) analysis. However, most of the natural infrasound has a short duration and spectral characteristics of the signal changes substantially over their duration (non-stationary signal). Therefore FT is not sufficient to analyze the behavior of these signals.

To overcome the problem of time, Short Time Fourier Transform (STFT) was developed. STFT Is one of the basic forms of time-frequency representations and involves a technique called windowing of the signal by which the FT is modified to analyze only a small section of the signal [22]. STFT can provide some information about when and at what frequencies a signal event occurs, however, the accuracy of these information depends on the size of the time window. The main drawback is that once a particular size for the time window is chosen, that window size is the same for whole time-frequencies analysis. This means that if we choose narrow window sizes, it will give good time resolution but poor frequency resolution while broader window sizes give good frequency resolution but poor time resolution. Most signals require a more flexible approach where we can vary the window size to determine more accurately either time or frequency.

To overcome the above limitation Wavelet Transform (WT) was developed as an alternative to STFT . In contrast to STFT , WT uses short windows at high frequencies and long windows at low frequencies [23].

Wavelet transform can be classified into two main groups: Discrete Wavelet Transform (DWT) and Continuous Wavelet Transform (CWT) . CWT operates over every possible scale and translation whereas DWT uses a specific subset of scale and translation values. DWT is used to reduce the computational burden of CWT [19]. DWT has been widely used for analyzing non-stationary signals and provides a time-frequency representation of

the signals [24]. In DWT, a signal is decomposed into low-frequency band (approximation coefficients) and higher frequency band (detail coefficients). The low-frequency band is used for further decomposition [19].

Farooq et al. [25] proposed a pre-processing stage based on wavelet denoising for extracting robust MFCC features in the presence of additive white Gaussian noise. They found that MFCC features extracted after denoising were less affected by Gaussian noise and improved recognition by 2 to 28 % for SNR in the range 20 to 0 dB.

Fatma Z. G et al. [26] have carried out a study on a classification of asthmatic breath sounds by using Wavelet Transforms and Neural Networks. They carried out Analyses of sound using both DWT and WPT separately. They concluded that Wavelet Transforms with neural network give higher success ratio for classifying asthmatic breath sounds. Moreover, they discovered that DWT has slightly better success than Wavelet Packet Transform (WPT) with respective to their study.

Chapter 3

Design

3.1 Introduction

This chapter explains the proposed solutions to the research problem. Section 3.2 describes design Considerations as a foundation for the research design. Section 3.3 provide comprehensive description of the proposed research design.

3.2 Design Considerations

3.2.1 Audio Sample Rate

Before processing the audio signal, the particular signal should be captured and converted into the digital format. In analog to digital conversion, a continuous signal is converted into to a sequence of samples (a discrete-time signal). The number of points an analog signal is measured (sampled) per second is called the sampling rate.

3.2.1.1 NyquistShannon Sampling Theorem

NyquistShannon sampling theorem is the fundamental theorem which expresses the relationship between analog signals (continuous-time signals) and the digital signals (discrete-time signals). This theory state that, if the highest frequency component of the analog signal is f_{max} then the sampling rate must be at least $2f_{max}$. If the sampling rate is less than the $2f_{max}$, the highest frequency elements in the analog signal will not be accurately represented in the digital output.

As described in the Section 1.1.1, fundamental frequency of Asian elephant rumbles fluctuates around 14 Hz to 24 Hz. With the harmonics, it will fluctuate around 14 Hz to

175 Hz. Therefore, according to NyquistShannon sampling theorem, sampling rate around 360 Hz is necessary for capturing the particular frequency range.

3.2.1.2 Oversampling

Oversampling is the process of sampling a signal with a sampling rate significantly higher than the Nyquist rate ($2f_{max}$). Oversampling increases resolution, reduces noise and helps avoid aliasing and phase distortion. Which means oversampling can improve the performance of applications which depends on information in the waveform shape of the signal.

Even though oversampling has several advantages, it will require more computation power, computation time and memory during the signal processing. Since the proposed approach intends to run on the ELOC node, maximum recording and processing sampling rate limited to 8000 Hz due to limited computation power and memory.

3.2.2 Audio Channels

ELOC node simultaneously captures the audio signal using two infrasonic microphones. Hence, input audio signal comes in the stereo format. Both channels contain similar audio signals but shifted in the time axis.

The feature extraction methodology is only capable of extracting the features from one channel at once. Since both channels contain similar signals, it is reasonable to apply feature extraction process to either one of two channels. However, both microphones can be affected by different background noise coming from the different directions form the primary sound source. This study proposed beamforming base stereo to mono conversion approach which minimizes the effect of such background noise, which will be discussed in Section 3.3.1.2.

3.2.3 Audio Format

Audio formats can categorize into three main categories: Uncompressed Audio Formats, Lossy Compressed Audio Formats, and Lossless Compressed Audio Formats.

Lossless Compressed Audio Formats reduce the file size while keeping all the information in source sound file. But not efficient as lossy compression in term of file size reduction. Lossy compressed audio formats reduce file size dramatically while removing some of the information in the source sound.

Uncompressed audio is the precise sound, a real sound signal that has been captured and converted to digital format without any additional processing. Therefore, uncompressed

audio files contain most accurate audio information and more significant in the file size. These files usually stored in the '.wav.' format on the disk.

For audio editing and audio processing, it is recommended to use uncompressed audio format, since it contains most precise information of the original audio signal. The proposed approach will use the uncompressed '.wav format during entire audio processing processes.

3.2.4 Possible Effect of Noise

As described in Section 2.3.1 electronic noise, acoustic noise and electromagnetic noise are the leading types of noise applicable to this study. Proposed approach attempts to mitigate the effect of such noise by using several techniques during the pre-processing stage of the audio signal, which will be discussed in the Section 3.3.1. Furthermore, proposed approach attempts to figure out a more robust set of features (features that are less affected by noise) which can be used to identify the infrasonic elephant rumbles with the presence of such noise, which will be discussed in the Section 3.3.2.

However, the proposed approach is not fully effective when the significant presence of the noise which has the same frequency range of the infrasonic elephant rumbles. This limitation needs to be addressed in future studies.

3.2.5 Segmentation Duration

Before processing signal, the signal should be segmented into equal length. The length of 2 seconds was selected based on the typical duration of Asian elephant rumbles and processing capabilities of single board computer used in ELOC node.

3.3 Research Design

The proposed research design incorporates four main modules: Pre-processing module, Feature extraction module, Feature Reduction module, and Classifier.

3.3.1 Preprocessing Module

The proposed preprocessing module consists of three main steps: Butterworth band-pass filter, wavelet-based denoising, and beamforming based stereo to mono conversion. Figure 3.1 represents the high-level architecture of the proposed preprocessing module.

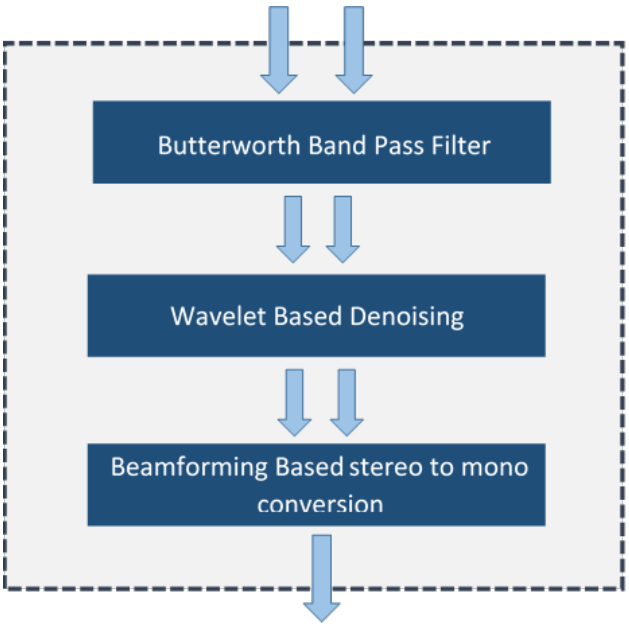

Figure 3.1: High-level architecture of the preprocessing module

3.3.1.1 Butterworth band-pass filter

Even though hardware level low-pass filter in the ELOC node attenuates frequencies above 150 Hz, it will allow pass frequency up to 300 Hz. Since elephant rumbles fluctuate around the range of 14 Hz to 174 Hz, any signal components beyond that range contain unwanted information for elephant rumble detection. Moreover, these unwanted signals can have an adverse effect on the elephant rumble detection. Thus the the Butterworth band-pass filter [27] with the following configuration applied before the further processing of the audio signal..

- Low-cutoff frequency - 10 Hz
- High-cutoff frequency - 150 Hz
- Filter order - 9
- Sample rate 600 Hz

The low-cutoff frequency and high-cutoff frequency are selected based on the typical frequency range of elephant rumbles. Filter order 9 is selected by analyzing the frequency response at several filter orders for the same sampling rate (600 Hz) and cutoff frequencies.

Figure 3.2 and Figure 3.3 represent the frequency spectrum of an elephant rumble recording before and after applying the Butterworth band-pass filter. It is clear that the tuned band-pass filter removes the unwanted frequencies while maintaining uniform sensitivity for the required frequency range. Since the filtering process removes the frequency range from 0

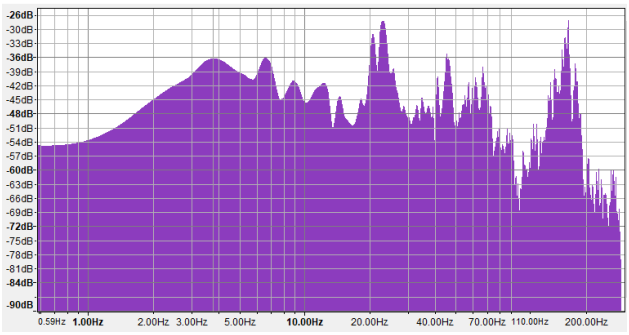

Figure 3.2: Frequency spectrum of an elephant rumble recording before applying the band-pass filter

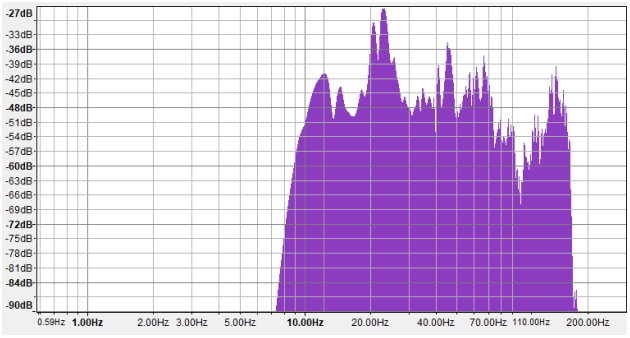

Figure 3.3: Frequency spectrum of an elephant rumble recording after applying the band-pass filter

Hz - 10 Hz and frequencies above 150Hz, it will mitigate the influence of the low-frequency noise, and high-frequency noise that exists beyond the typical range of elephant rumbles.

3.3.1.2 Beamforming Based Stereo to Mono Conversion

As shown in Figure 3.4 two microphones in the ELOC deployment unit, located at a distance of 3m from each other, simultaneously captures the audio signal using two infrasonic microphones. Hence, input audio signals come in the stereo format. Any sound waves coming either from the front or the back, as seen in Figure. 3.4, reach one of the microphones earlier than the other. Thus, both channels contain similar audio signals but are shifted in the time axis. An assumption is made here that the sound waves are parallel to each other from the point of view of the microphones.

However, both microphones can be affected by background noise coming from different directions to the primary sound source. Therefore beamforming based stereo to mono conversion approach use to minimizes the effect of such background noises. By considering the reference signal (the signal which reaches the setup first) and the delayed signal, can calculate the shift between the two signals in term of samples as shown in Equation 3.1.

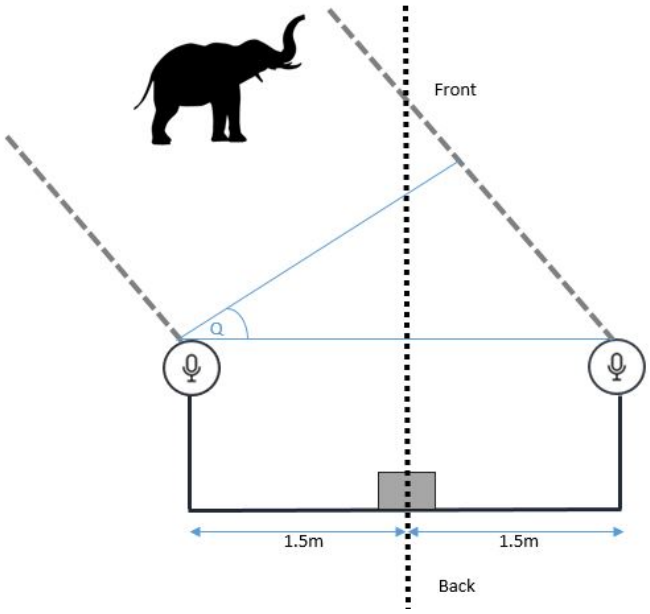

Figure 3.4: Basic setup of the ELOC deployment unit where the pair of microphones (ELOC node) located at a 3m distance from each other and capturing data in a time-synchronized manner

$$argmax \left\{ \sum_{n=0}^{n} \sum_{m=0}^{n} \|delayed\{n\}\ ref\{n+m\}\| \right\} = argmax \left(\sum_{n=0}^{n} \|C_{ref,delayed[m]}\| \right) \tag{3.1}$$

Where n is the number of samples in the given signal segment, m is the number of shifted samples between the two signal and is $C_{ref,delayed[m]}$ the cross-correlation matrix between two signals.

After calculating m, shift the reference signal by adding zero padding in front of the reference signal segment. Then m frames are lifted from the tail of the reference signal segment to maintain the constant number of frames in the signal segment. Finally, merge the shifted reference signal and the delayed signal by taking the average of the corresponding frames in the two signals. This process improves the strength of the dominant sound source while reducing the strength of noise coming from the other directions. A flowchart showing the process of beamforming based stereo to mono conversion is presented in Figure 3.5.

3.3.2 Feature Extraction Module

The proposed feature extraction module consists of three sub-modules: wavelet-based signal decomposition sub-module, wavelet-based signal reconstruction sub-module and fea-

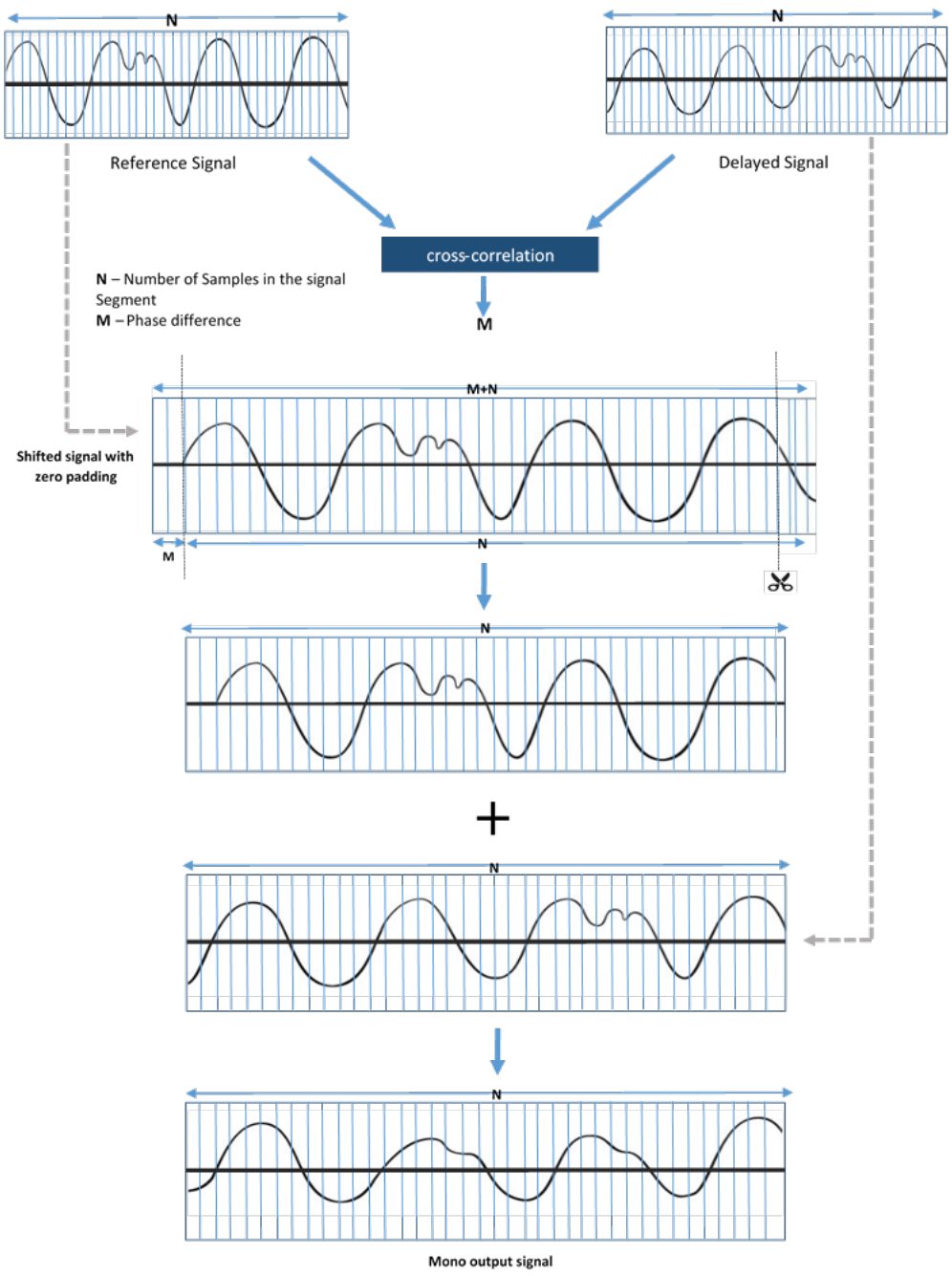

Figure 3.5: The process of beamforming based stereo to mono conversion

ture extraction sub-module. Figure 3.6 represents the high-level architecture of the proposed Feature extraction module. Output signal segment from the pre-processing module is taken as the input for the feature extraction module

3.3.2.1 Wavelet-based decomposition and reconstruction

Firstly, preprocessed sound segment is decomposed into sub-bands using DWT with the following configuration.

- Segment length - 2 seconds (600Hz, 1200 samples)

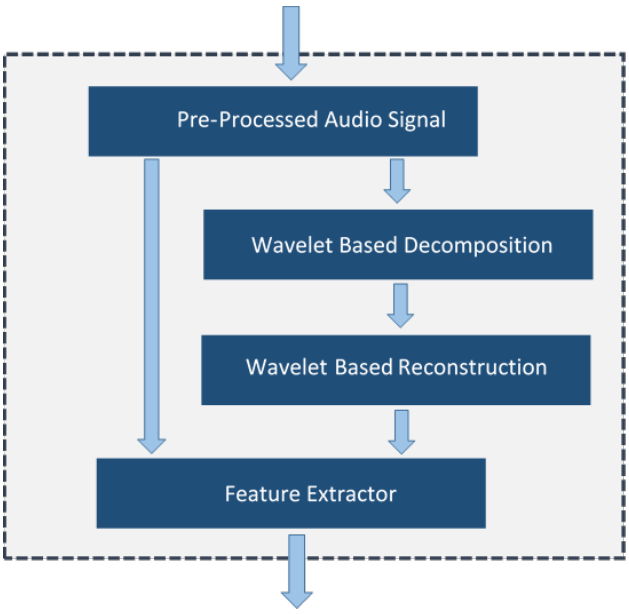

Figure 3.6: High-level architecture of the feature extraction module

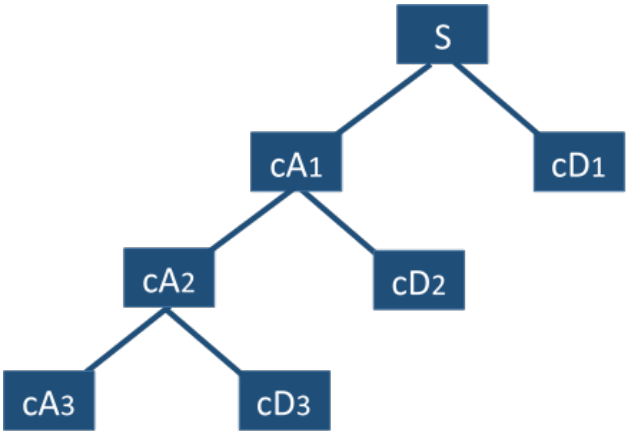

Figure 3.7: DWT Decomposition tree

- Mother wavelet - Daubechies 3 (db3)
- Decomposition level - 3

The accuracy of classification depends on the type of the selected mother wavelet [28]. We evaluated the decomposition capability of the different mother wavelets with different decomposition level, based on the energy-to-Shannon entropy based self-evaluation criterion given in [28] and selected the mother wavelet and decomposition level mentioned above which give the maximum energy-to-Shannon entropy ratio for elephant rumble signal decomposition.

Once DWT is applied to the signal segment with the decomposition level 3, signal segments are divided into cD1-cD3 detail sub-bands and cA3 approximation sub-band. Figure 3.7 represents the DWT decomposition tree and Figure 3.8 is the layout of the coefficients in the output vector.

After obtaining the wavelet transform coefficients, we generate the 7 reconstructed signal

Figure 3.8: Layout of the resulting wavelet-transform vector. S is the input signal, cAn is the approximate coefficients of n^{th} level and cDn is detailed coefficients of n^{th} level.

Table 3.1: Description of reconstructed signals

A3	Reconstructed signal using approximation coefficient sub-bands at level 3 (cA3)
D3	Reconstructed signal using detailed coefficients sub-bands at level 3 (cD3)
D2	Reconstructed signal using detailed coefficients sub-bands at level 2 (cD2)
D1	Reconstructed signal using detailed coefficients sub-bands at level 1 (cD1)
all_recon	Reconstructed signal using cD1-cD3 detailed sub-bands and cA3 approximation sub-band
rm_cA3_cD1	Reconstructed signal using cD1 and cD2 detailed sub-bands.
rm_cA3	Reconstructed signal using cD1-cD3 detailed sub-bands

variation of the original signal by applying the Inverse Discrete Wavelet Transformation (IDWT) to individual wavelet coefficient sub-bands and combinations of wavelet coefficients sub-bands as given in the Table 3.1.

Figures 3.9, 3.10, 3.11 and 3.12 represents the process of generating reconstructed signals.We use these reconstructed signal variations and the original signal for the final feature extraction process.

Since DWT decomposes the original input signal into different frequency sub-bands, reconstructed signals only contain the frequency components in the particular wavelet coefficients which are used to reconstruct the signal. Extracting features from such reconstructed signal variations will allow extracting features from each frequency sub-band separately. That means noise with a particular frequency will only affect one or several of the frequency sub-bands and other sub-bands remain unaffected. Thus, combinations of features extracted

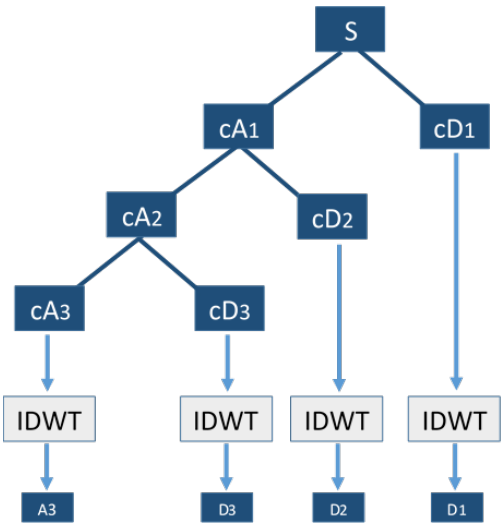

Figure 3.9: Signal reconstruction using individual sub-bands.

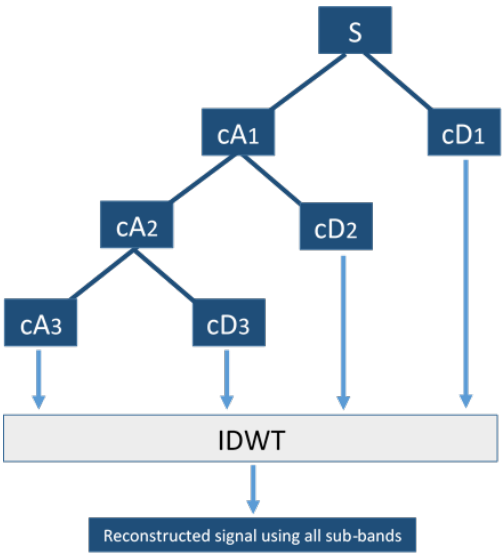

Figure 3.10: Signal reconstruction using all sub-bands.

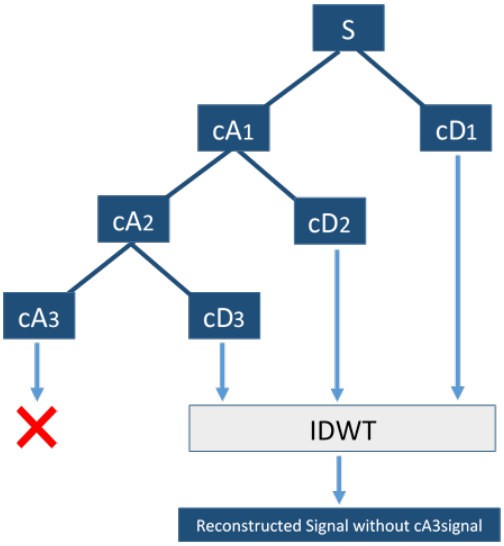

Figure 3.11: Signal reconstruction without cA3.

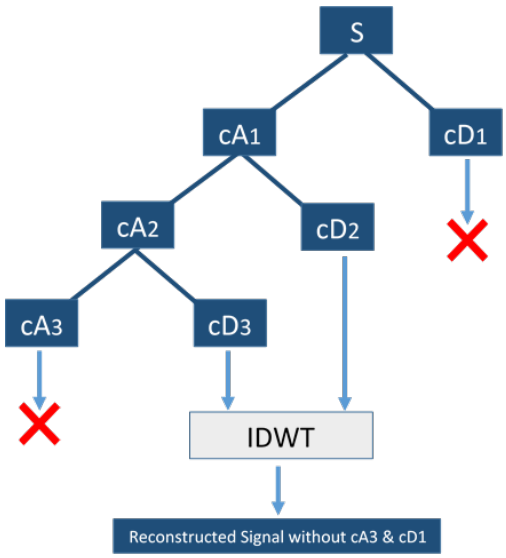

Figure 3.12: Signal reconstruction without cA3 and cD3.

from the different frequency sub-bands are more robust to noise than the features extracted from the entire frequency range.

Furthermore, DWT does not merely divide the given signal into frequency sub-bands. The iterative process of the multi-resolution wavelet decomposition depends on the mother wavelet which is used for decomposition. Mother wavelet 'db3' is empirically selected to match with the properties of elephant rumbles. Therefore, it has a higher tendency to pass sound waves more similar to elephant rumbles while filtering out waves significantly different than the elephant rumbles in each iteration of multi-resolution wavelet decomposition.

3.3.2.2 Feature Extraction

Following are the features extracted in proposed feature extraction module.

3.3.2.3 Chroma Features

In the context of audio processing the Chroma features or the Chroma gram representation of audio in which the entire spectrum is projected onto 12 pitch classes representing the 12 distinct semitones (Chroma)

3.3.2.4 Mel-Frequency Cepstral Coefficient

In audio processing, the Mel Frequency Cepstrum (MFC) is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear Mel scale of frequency. MFCC are coefficients that collectively make up a Mel frequency.

3.3.2.5 Root Mean Square (RMS) Energy

RMS Energy is the square root of the arithmetic mean of the squares of the values or the square of the function that defines the continuous waveform.

3.3.2.6 The spectral centroid

The spectral centroid is a measure used in digital signal processing to characterize a spectrum. It indicates where the "center of mass" of the spectrum is located. Perceptually, it has a robust connection with the impression of "brightness" of a sound.

3.3.2.7 Spectral Contrast Features

Spectral Contrast feature is proposed to represent the spectral characteristics of a music clip. It represented the relative spectral distribution instead of an average spectral envelope.

3.3.2.8 Spectral-Roll-off

The roll-off frequency is defined as the frequency under which some percentage (cutoff) of the total energy of the spectrum is contained.

3.3.2.9 Zero Crossing Rate (ZCR)

The zero-crossing rate is the rate of sign-changes along a signal, the rate at which the signal changes from positive to negative or back.

Even though features mentioned above are extracted from the pre-processed audio signal and re-construed variations of that signal, the entire feature vector is not used for training and prediction purposes. Extracted feature vector is then passed on to the feature selection module.

During the feature extraction process, we extract 84 different spectral features under 9 main feature categories. The main feature categories are given in the Table 3.2. Table 3.3 illustrate the composition of the extracted features.

We extract the above feature composition separately from every reconstructed signal variation and the original signal. Therefore, finally we will get 672 (84 * 8) extracted features from a given signal segment. However, the entire feature vector will not be used for training and prediction purposes. The extracted feature vector is then passed on to the feature selection module.

Table 3.2: Feature categories

Chroma Features
Mel-Frequency Cepstral Coefficient
Root-Mean-Square (RMS) Energy
The spectral centroid
Spectral Contrast
Spectral bandwidth
Spectral-Roll-off
Zero Crossing Rate
Polynomial Features

Table 3.3: Feature categories

Features	Number of values per frame
Chroma features	36
MFCC features	5
RMS Feature	1
Spectral centroid	1
Spectral bandwidth	1
Spectral contrast	6
Spectral roll-off	1
Zero Crossing Rate	1
Melspectrogram features	30
Polynomial features	2
Total number of features	84

3.3.3 Feature Selection Module

Feature selection module assists in creating an accurate predictive model. Proposed feature selection module consists of two sub-modules: Quantile transformer and feature selector. Figure 3.13 represents the high-level architecture of the proposed feature selection module. The output feature vector from the feature extraction module is taken as the input for the feature selection module.

3.3.3.1 Quantile Transformer

Many machine learning algorithms are designed with an assumption that all features vary on comparable scales while each feature takes values near to zero. In contrast, features in the feature vector provided by the feature extraction module has different scales. Moreover, because of the nature of the input signals, it has a higher tendency to contain huge outliers. This is because it is impossible to capture the extact elephant rumble in the natural environment. These two characteristics can decrease the predictive performance of many machine

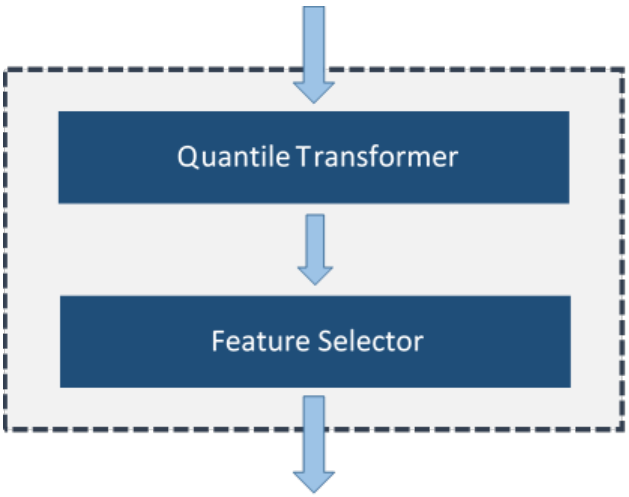

Figure 3.13: High-level architecture of feature selection module

learning algorithms. Also, unscaled data can slow down the training and prediction process of many machine learning algorithms.

The proposed approach uses quantile transformer [29] to bring extracted features into a comparable scale while minimizing the effect of outliers. Quantile transformer, transforms the features to follow a uniform distribution, using quantile information. This means for a given feature, the quantile transformer tends to flatten out the most frequent value. The quantile transformer applies a nonlinear transformation to each feature independently, by using the cumulative density function of a feature. Therefore, quantile transformer should be tuned with the features extracted from the training dataset. As mentioned in the Section 1.5, this approach depends on two training datasets. Combination of features extracted from both datasets have been considered during the quantile transformer tuning process.

3.3.3.2 Feature Selection Sub-Module

For classification with small training samples and high dimensionality, feature selection plays a vital role in avoiding over fitting problems and improving classification performance. Feature selection will remove irrelevant, correlated and redundant features and choose a robust subset of features which has the highest relevance to classification. Proposed feature selection module has been built based on the feature ranking with recursive feature elimination [30] and cross-validated selection of the optimal number of features.

Since this is a wrapper feature selection approach, the feature selection process depends on the given external classifier and the feature vector obtained from the training dataset with corresponding class labels. We evaluated several classifiers with the cross-validation accuracy and selected the Support Vector Machine (SVM) as the optimal classifier since

it provides better accuracy than the other evaluated classifiers. Since we use SVM as a classifier, the feature selection module is also tuned based on recursive feature elimination and cross-validated with the SVM.

In contrast with the quantile transformer tuning process, this study only considers the features extracted from the dataset-1 (which have replayed elephant rumbles). Because even though the proposed approach uses a very high-quality dataset (captured using domain-specific device) during the training process, it should be able to classify the sound signal captured by ELOC node (which is not sensitive as the domain-specific device and signal can be slightly distorted because of low-cost hardware components). The underlying assumption is, any feature subset which is robust enough to classify the replayed version of elephant rumble captured by the ELOC node should be able to classify the direct elephant rumble captured by ELOC node or any device more sensitive than the ELOC node. The validity of this assumption will be evaluated in Section 5.

3.3.4 Classifier

The final module of the proposed approach is a binary classifier. The classifier determines whether the given set of features extracted from infrasound segment, consists of an elephant rumble or not.

SVM selected as the optimal classifier since it provides better accuracy than the other classifiers considered. The classifier is trained using dataset-2. Since this dataset was recorded using a 48000-sample rate, firstly it is converted to the 600-sample rate. All steps mentioned above from preprocessing to feature selection are applied to this dataset excluding the Beamforming based stereo to mono conversion step because these recordings are already in the mono format. Finally, hyper parameters of the classifier have been tuned based on the cross-validation score with different parameters combinations.

3.3.5 Elephant Detection and Localization

Once sound segment was captured by the ELOC node, it will go through above-mentioned pre-process and features extraction models. Then features are transformed into the prede-fined sales using the pre-trained quantile transformer. Transformed features then pass to the pre-trained feature selection module. Output feature vector from the selection module is finally passed to the pre-trained classifier to decide whether the captured sound segment consists with elephant rumbles or not. If it is an elephant rumble, the cross-correlated output

of the pre-processing module is used to localize the location of the elephant.

3.3.6 High-Level Architecture of the Proposed Approach

Figure 3.14 represents the details high-level architecture of the proposed approach.

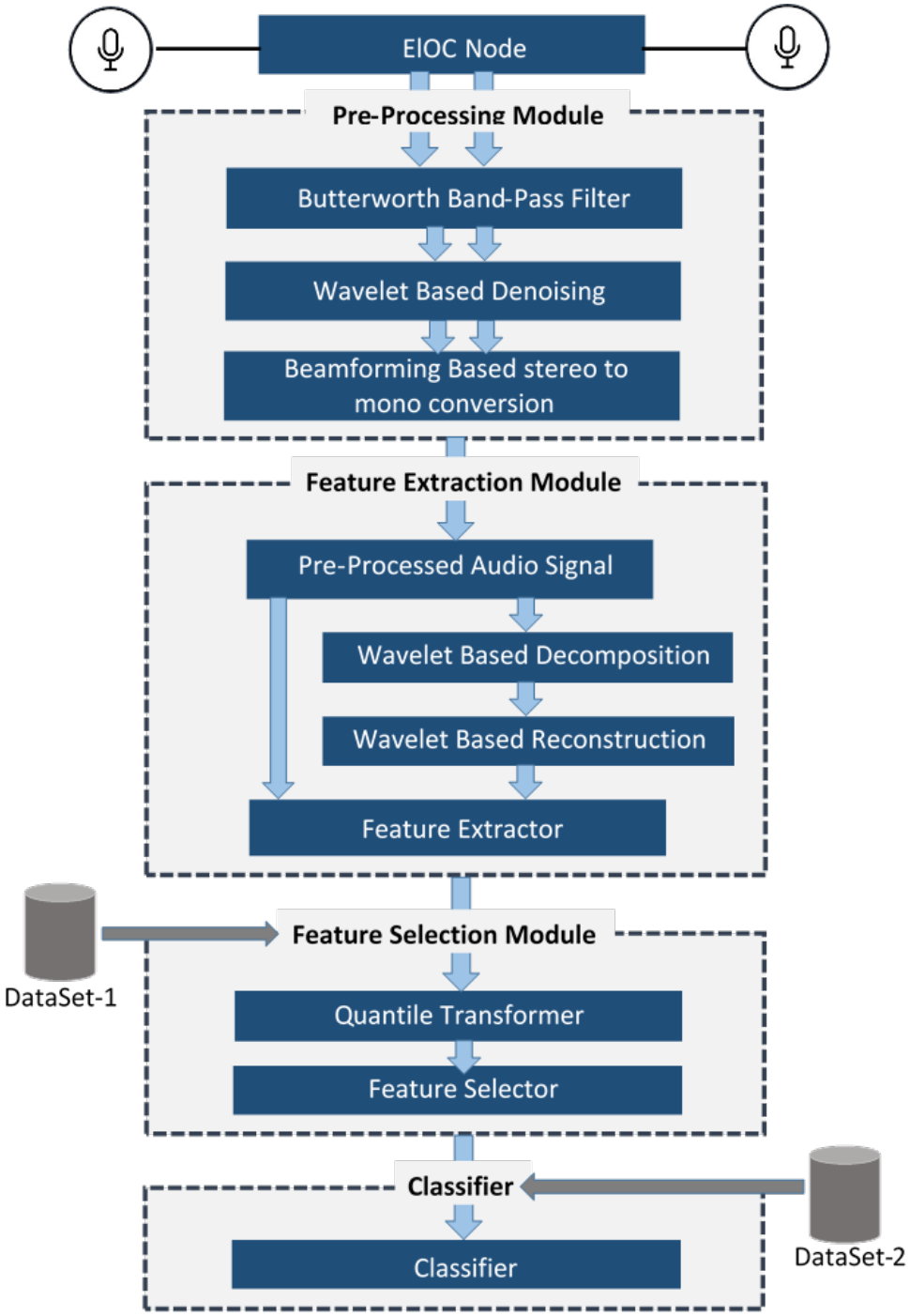

Figure 3.14: High-level architecture of the proposed approach.

Chapter 4

Implementation

4.1 Introduction

This chapter explains the proposed solutions to the research problem. Section 4.2 describes the software tools utilized for implementation. Section 4.3 and 4.4 describes the implementation details of the proposed research design.

4.2 Software Tools

The proposed solutions were implemented using Python 3.5 with the libraries given in the Table 4.1

Table 4.1: Support libraries

LibROSA	python package for music and audio analysis
PyWavelets	Open Source wavelet transform software for the Python programming language
scikit-learn	Open source tools for data mining and data analysis
resampy	python module for efficient time-series resampling
pandas	Open source library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language

Wider availability of the supporting libraries and the ability for the same implementation to run on both the Desktop computer (test environment), and the ELOC node (production environment) are the main reasons to select python as the developing language.

All of the mentioned features in Table 3.2 are extracted using the Python LibROSA [31] library. These feature extraction methods take the audio file, sample rate, frame length,

and hop length (shift between frames) as the input arguments. However, some of the librosa functions require some degree of tuning based on the dataset and specific frequency range that is to be analyzed. These feature extraction methods are configured to analyze the infrasound as follows.

- **chroma_cqt and chroma_cens**

  These two methods require as parameters, the minimum frequency to analyze and the number of octaves required to analyze the above the minimum frequency. Since elephant rumbles fluctuate around 14 Hz to 24 Hz (with harmonics 14 Hz to 175 Hz), 10 Hz is selected as the minimum frequency with 4 octaves. Which will result in the analysis of a frequency range from 10 Hz to 160 Hz (minimum frequency 10 Hz, 1st octave 20Hz, 2nd octave 40Hz, 3rd octave 80Hz and 4th octave 160HZ)

- **melspectrogram and mfcc**

  These two methods require the minimum and maximum frequency range that has to be analyzed. 10 Hz as the minimum frequency and 160 Hz as the maximum frequency are selected to match with the frequency range of elephant rumbles.

- **spectral_contrast**

  This method requires the frequency cut-off for the first frequency sub-band and the number of frequency bands as input. 5 Hz is selected as the minimum frequency cut-off, and six frequency sub-bands are considered. Which will result in the analysis of a frequency range from 0 Hz to 160Hz. ([0Hz - 5 Hz] [5Hz - 10 Hz] [10Hz - 20 Hz] [20 Hz - 40 Hz] [40Hz - 80 Hz] [80 Hz - 160Hz])

Audacity 2.2.1, an open source, cross-platform audio software for multi-track recording and editing, is used for the experiments and for analytical purposes during this study.

4.3 Implementation Details - Preprocessing Module

4.3.1 Butterworth band-pass Filter

As described in Section 3.3.1 preprocessing module starts with the Butterworth band-pass filter. The following functions showcase the implementation of the Butterworth band-pass filter using butter and filtfilt methods in the SciPy library.

```
1       def butter_bandpass_filter(data, lowcut, highcut, fs, order=5):
2               b, a = butter_bandpass(lowcut, highcut, fs, order=order)
3               y = lfilter(b, a, data)
4               return y
```

The phase shift of two signals captured by microphones is critical to the further process. However, data filtered by the lfilter method are subject to the phase shift. Therefore here this implementation used the filtfilt method. The filtfilt method applies a linear digital filter twice, once forward and once backward, which will generate a filtered output with zero phase and filter order twice than the original.

4.3.2 Beamforming Based Stereo to Mono Conversion

Shift between two audio signals is calculated using cross-correlation. The following function 'compute_shift' calculates shift between the two signals in terms of the number of samples. Output value can be negative or positive based on the direction of the shift relative to the reference signal.

```
1       def compute_shift(x, y):
2               assert len(x) == len(y)
3               c = cross_correlation_using_fft(x, y)
4               assert len(c) == len(x)
5               zero_index = int(len(x) / 2) - 1
6               shift = zero_index - np.argmax(c)
7               return shift
```

The following function 'shifter' shifts the signal accordingly by adding zero padding and maintain the constant number of frames by lifting the frames from the tail.

```
1       def shifter(a1, b1):
2               shift = compute_shift(a1, b1)
3               abs_shift = abs(shift)
4
5               print (shift)
6
7               padded = list(np.zeros(abs_shift))
8               if shift < 0:
9               padded.extend(b1)
10              b1 = padded[:-abs_shift]
11
12              elif shift > 0:
13              print (a1)
14              padded.extend(a1)
15              a1 = padded[:-abs_shift]
16
17              return a1, b1
```

Finally, the function 'stereo_to_mono' generate the mono signal by taking the mean of two signals return by the function 'shifter.'

```
1       def stereo_to_mono():
2               x, y = shifter(a, b)
3               Z = np.mean([x, y], axis=0, dtype=np.float64)
4               return Z
```

4.4 Implementation Details - Feature Extraction Module

4.4.1 Wavelet Decomposition and Reconstruction

The following function 'wavelet_reconstructor' exhibits the process of multi-level wavelet decomposition and reconstruction. The function is implemented using wavedec and waverec methods in the PyWavelets library.

Wavedec function takes the audio signal, mother wavelet name and decomposition level as the input arguments. Then it will return the ordered list of wavelet coefficients array as follows.

[cAn,cDn,cDn-1 cD1]

Where 'n' denotes the level of decomposition. The head element cAn is the approximate coefficient array, and subsequent elements (cDn, cDn-1...cD1) are details coefficients arrays.

The waverec function performs the multilevel wavelet reconstruction while taking the wavelet coefficients and mother wavelet name as the input argument. To generate a reconstructed audio signal which is based on the specific wavelet coefficient or the combination of them, the waverec function is used by providing wavelet coefficient list after omitting the irrelevant coefficients.

```
1   def wavelet_reconstructor(signal, wavelet_name, decomposition_level)
        :
2       reconstructed_signal = {}
3
4       # row signal
5       # reconstructed_signal.append(signal)
6       reconstructed_signal['row'] = signal
7
8       coeffs = wavedec(signal, wavelet_name, level=
            decomposition_level)
9       backup = copy.deepcopy(coeffs)
10      # reconstructed signal with all sub-bands
11      # reconstructed_signal.append(pywt.waverec(coeffs,
            wavelet_name))
12
13      reconstructed_signal['all_recon'] = pywt.waverec(coeffs,
            wavelet_name)
14      # with out approximate sub-band
15      coeffs[-4] = np.zeros_like(coeffs[-4])
```

```
16              # reconstructed_signal.append(pywt.waverec(coeffs,
                    wavelet_name))
17              reconstructed_signal['rm_approx'] = pywt.waverec(coeffs,
                    wavelet_name)
18
19              # with out approximate and level - 1 sub-band
20              # coeffs[-4] = np.zeros_like(coeffs[-4])
21              coeffs[-1] = np.zeros_like(coeffs[-1])
22              # reconstructed_signal.append(pywt.waverec(coeffs,
                    wavelet_name))
23              reconstructed_signal['rm_approx_cd1'] = pywt.waverec(coeffs,
                    wavelet_name)
24
25              # single sub-bands
26              # only level 3 approximate coefficient
27              coeffs = copy.deepcopy(backup)
28              coeffs[-1] = np.zeros_like(coeffs[-1])
29              coeffs[-2] = np.zeros_like(coeffs[-2])
30              coeffs[-3] = np.zeros_like(coeffs[-3])
31              # reconstructed_signal.append(pywt.waverec(coeffs,
                    wavelet_name))
32              reconstructed_signal['cA3'] = pywt.waverec(coeffs,
                    wavelet_name)
33
34              # only level 3 detailed coefficient
35              coeffs = copy.deepcopy(backup)
36              coeffs[-1] = np.zeros_like(coeffs[-1])
37              coeffs[-2] = np.zeros_like(coeffs[-2])
38              coeffs[-4] = np.zeros_like(coeffs[-4])
39              # reconstructed_signal.append(pywt.waverec(coeffs,
                    wavelet_name))
40              reconstructed_signal['cd3'] = pywt.waverec(coeffs,
                    wavelet_name)
```

```
41
42              # only level 2 detailed coefficient
43              coeffs = copy.deepcopy(backup)
44              coeffs[-1] = np.zeros_like(coeffs[-1])
45              coeffs[-3] = np.zeros_like(coeffs[-3])
46              coeffs[-4] = np.zeros_like(coeffs[-4])
47              # reconstructed_signal.append(pywt.waverec(coeffs,
                    wavelet_name))
48              reconstructed_signal['cd2'] = pywt.waverec(coeffs,
                    wavelet_name)
49
50              # only level 1 detailed coefficient
51              coeffs = copy.deepcopy(backup)
52              coeffs[-2] = np.zeros_like(coeffs[-2])
53              coeffs[-3] = np.zeros_like(coeffs[-3])
54              coeffs[-4] = np.zeros_like(coeffs[-4])
55              # reconstructed_signal.append(pywt.waverec(coeffs,
                    wavelet_name))
56              reconstructed_signal['cd1'] = pywt.waverec(coeffs,
                    wavelet_name)
57
58              return reconstructed_signal
```

Finally, the function 'wavelet_reconstructor' will return the dictionary of reconstructed audio signals. (Dictionary key represents the wavelet coefficient used to generate the audio signal, and relevant value is reconstructed audio signal)

4.4.2 Feature Extraction

All of the mentioned features in Section 3.3.2.2 are extract using python librosa library. Those feature extraction methods take the audio file, sample rate, frame length, and hop length (shift between frames) as the input arguments.

The function extract_features exhibits the complete feature extraction process. Implementation of the functions is provided in Appendix A

Finally, the function 'extract_features_all_signal_type' extracts all of the features men-

tioned above from the input signal and reconstructed variations of that signal and returns as a single feature vector.

```
1       def extract_features_all_signal_type(signal_array, sample_rate,
           frame_length, hop_length, feature_class):
2               final_feature_vector = []
3
4               no_of_frame_in_row_signal = len(signal_array['row'])
5               for key in signal_array.keys():
6
7               # after the reconnecting it will change the total number of
                   samples by +/- 1 .
8               # This will lead to diffenet number of frames in given
                   signals.
9               # to overcome this all recontacted signal length limit to
                   the same size of row signal
10              signal = signal_array[key][:no_of_frame_in_row_signal]
11
12              extract_features(signal, key, sample_rate, frame_length,
                   hop_length, feature_class, final_feature_vector)
13
14              return final_feature_vector
```

Implementation related to the experiments and evaluation are given in Appendix A

Chapter 5

Results and Evaluation

5.1 Experiment Results

5.1.1 Frequency range filtering with Butterworth band-pass filter

Before applying the Butterworth band-pass filter, the filter should tune to match with the filtering requirement. Low-cutoff frequency, High-cutoff frequency and filter order are the tuning parameters of the Butterworth band-pass filter. Separate tuning needs to be carried out for the different sampling rates.

To tune the Butterworth band-pass filter, we implemented a helper function to plot the frequency response at given filter orders for the same sampling rate and cutoff frequencies. Figure 5.1 represents the frequency response for different filter orders under the 600 Hz sampling rate, 10 Hz low-cutoff frequency, and 150 Hz High-cutoff frequency.

According to the Figure 5.1 filter order 9 is more suitable since it has a maximum frequency response from 10 Hz to 140Hz.

To demonstrate the importance of correct Butterworth band-pass filter tuning, Frequency spectrum and waveform of the elephant rumble filtered with higher filter order - 12 is shown in Figuries 5.4 and 5.5.

As shown in the Figure 5.5 application of un-tuned Butterworth band-pass filter distort the input signal.

5.1.2 De-noising experiment using wavelet transformation

Another experiment was carried out to de-noise infrasonic elephant rumbles using wavelet transformation. First, White Gaussian Noise (WGN) was added to the selected elephant

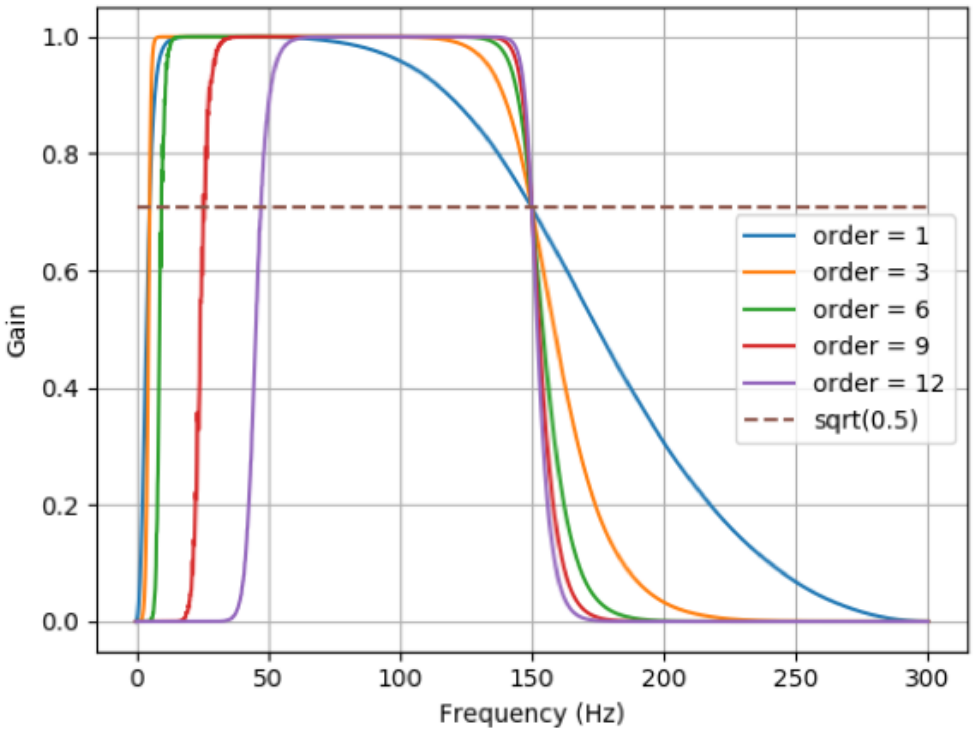

Figure 5.1: The frequency response for different filter orders under the 600 Hz sampling rate, 10 Hz low-cutoff frequency, and 150 Hz High-cutoff frequency

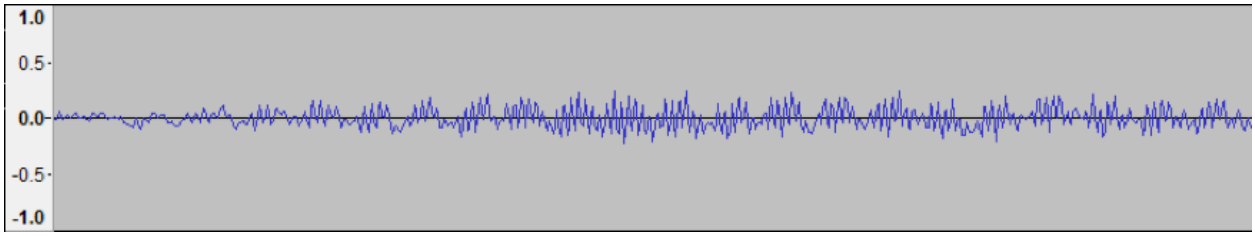

Figure 5.2: The waveform of elephant rumble recording before the applying band-pass filter

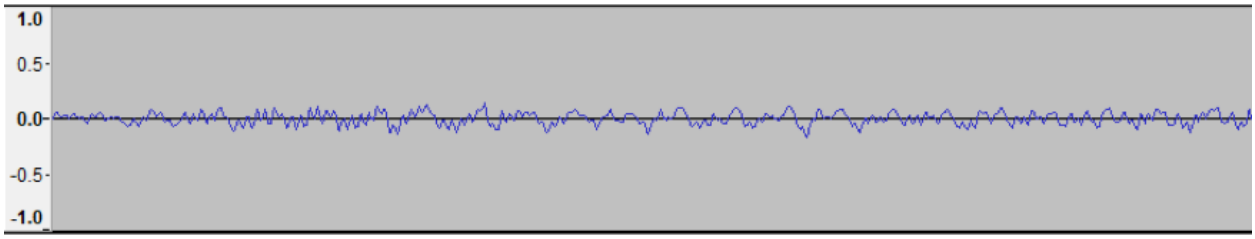

Figure 5.3: The waveform of elephant rumble recording after applying band-pass filter with order 9

rumble signal. Then it was attempted to de-noise the noisy signal using wavelet transformation. Finally, the similarity of the original signal and de-noise signal was measured using cross correlation. De-noising performance with several wavelet families as well as with different decomposition levels were measured and compared the under different SNR.

Figure 5.6 illustrates the de-noising performance with three wavelets families, Coiflet 1

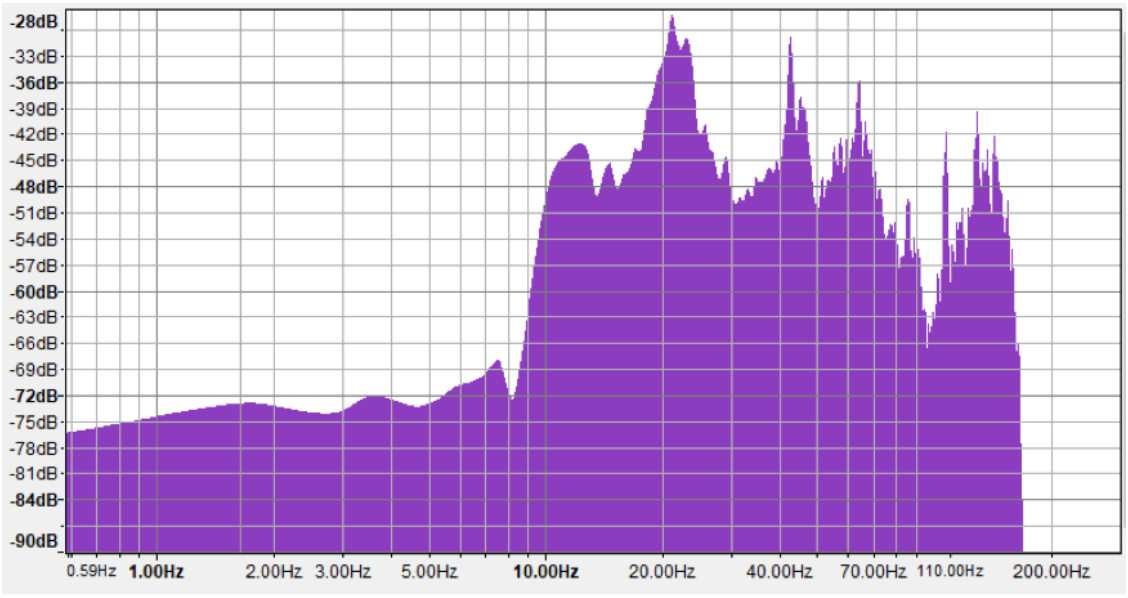

Figure 5.4: Frequency spectrum and waveform of the elephant rumble filtered with higher filter order - 12.

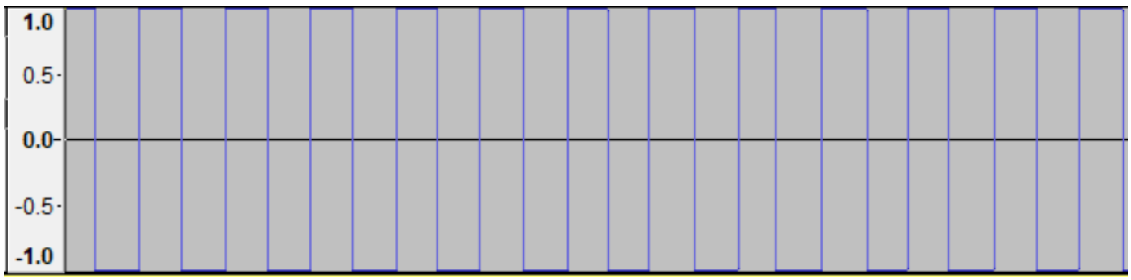

Figure 5.5: Un-tuned Butterworth band-pass filter distort the input signal.

to 5, Daubechies 5 to 9 and Symlets 1 to 5. According to above Figure 5.6, there is no significant performance difference with selected wavelet with relevance to elephant rumble de-noising.

Furthermore, the de-noising performance was compared under different SNR.

Figure 5.7 shows original audio signal, the audio signal with noise (AWGN=10) and the recovered signal. The right one shows the cross correlation of the original signal and the signal de-noise of left figure.

Figure 5.8 shows original audio signal, the audio signal with noise (AWGN=1) and the recovered signal. The right one shows the cross correlation of the original signal and the signal de-noise of left figure.

According to above figures, this de-noising process performs effectively even with SNR = 1 under the Additive White Gaussian Noise (AWGN). Based on experiment results, it was observed that wavelet based de-noising can be effectively use to de-noise infrasonic elephant rumble affected by WGN.

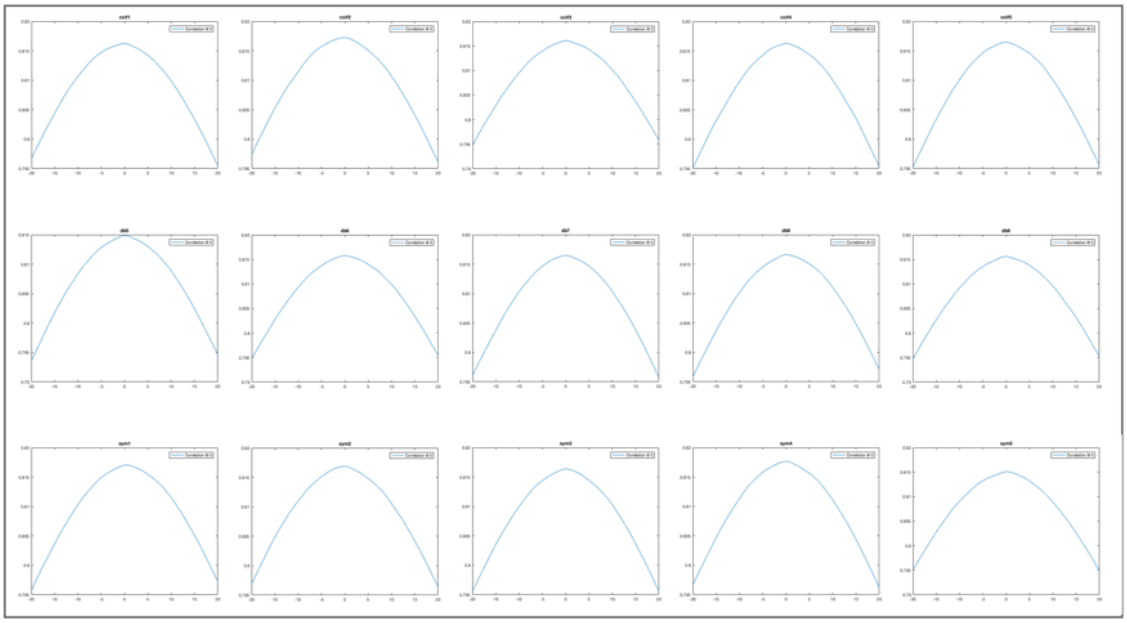

Figure 5.6: De-noising Performance.

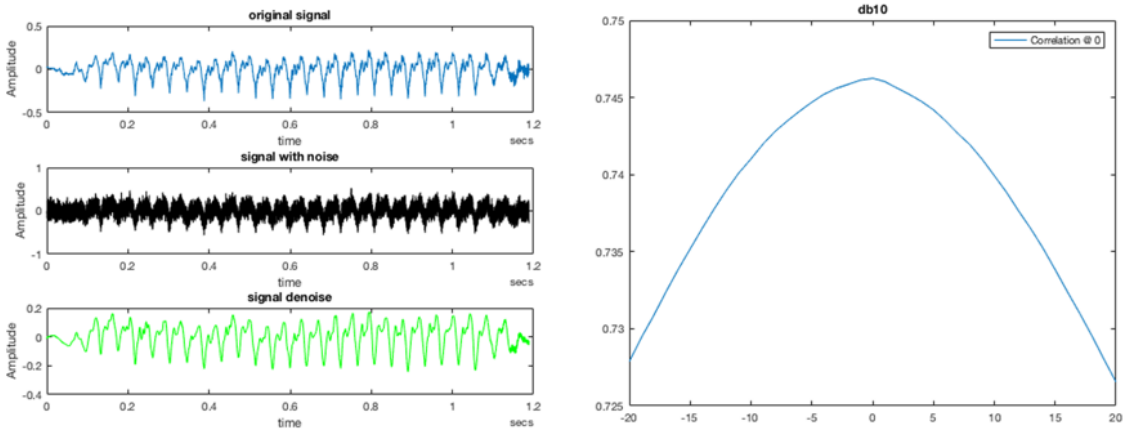

Figure 5.7: De-noising Performance under SNR=10.

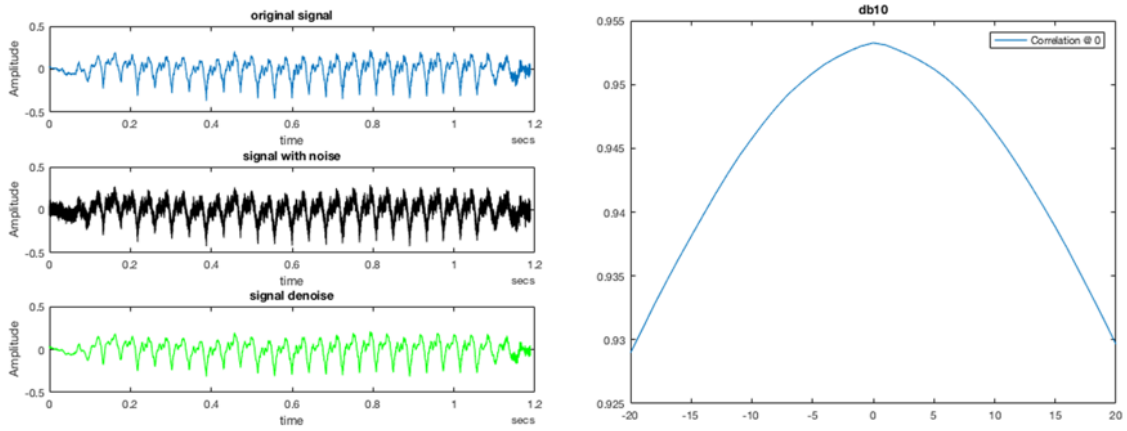

Figure 5.8: De-noising Performance under SNR=1.

Even though this finding has potential to improve the performance of elephant detection as well as the localization process, when we analyzed the effect of acoustic noise such as revolving machines, wind and moving vehicles, they are much different from AWGN.

5.1.3 Feature Selection

5.1.3.1 Analysis of the importance of individual features

To analyze the significance of individual features towards the classification, we analyze the importance of features using a forest of tree method provided in [29]. Figure 5.9 represents the feature importance of top 30 features.

According to Figure 5.9 features extracted from the reconstructed variation of the signal has higher importance than the feature extracted from the row signal prove the validity of the features extracting on top of the DWT. Table in the Appendix B represent feature importance of the top 50 features along with their important value.

5.1.3.2 Feature selection for the classifier training

As described before optimal combinations of features are selected based on the recursive feature elimination score and cross-validation score.

Figure 5.10 represents the cross-validation score variation with the number of features selected for features extracted from dataset-1.

Figure 5.11 represents the cross-validation score variation with the number of features selected for features extracted from dataset-2.

According to the Figure 5.10 and 5.11, it is clear that to classify the elephant rumbles captured by the domain-specific device (dataset-2) only 30 features are required. However, to classify the replayed version of elephant rumbles (dataset-1) 196 features are necessary. Low sensitivity and sound wave distortion due to the low-cost hardware can be assumed as the primary reasons for that.

5.2 Classifier Evaluation

5.2.1 Dataset

To train and evaluate the proposed approach we used the elephant sound dataset from the Smithsonian Conservation Biology Institute. The dataset contains 5592 different elephant sound recordings that were made in Sri Lanka, with 48000 Hz sampling rate and 32-bit depth. The dataset has been manually annotated by a biology expert, and there are 14 types of elephant calls in the data. We only use rumble recordings among these 14 call types. As the negative dataset, we use freely available infrasound recordings taken from the internet.

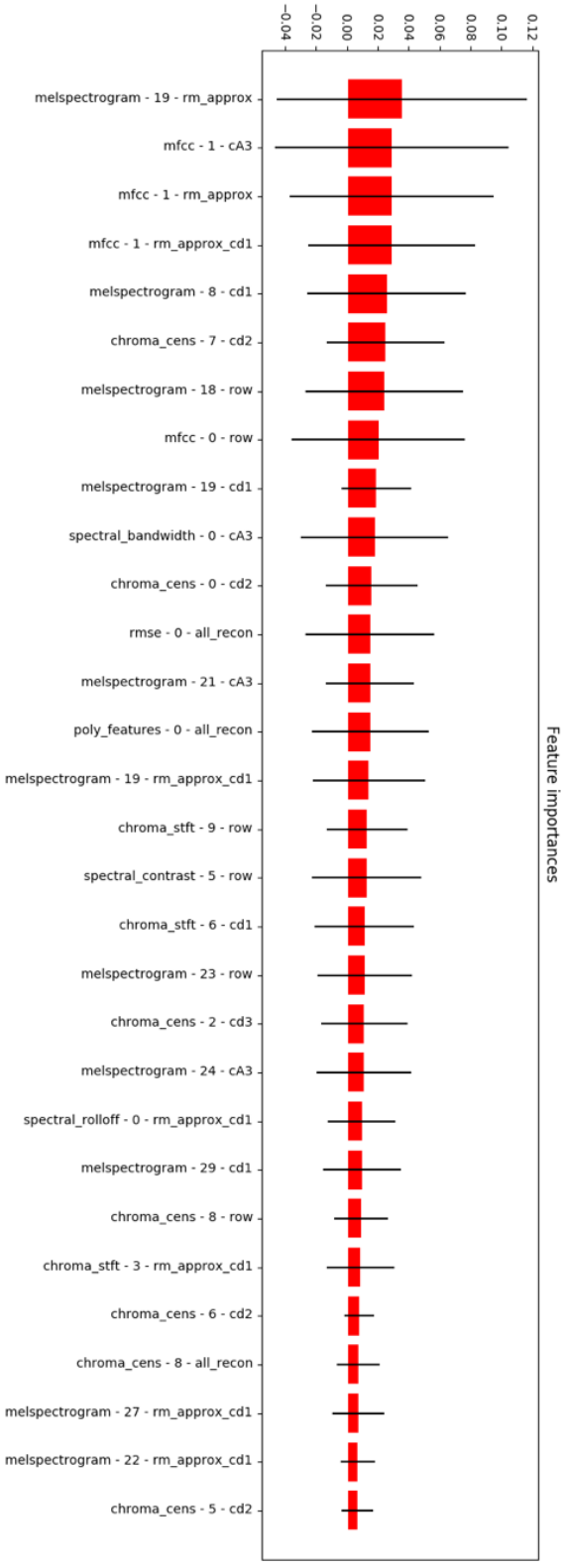

Figure 5.9: The feature importance of top 30 features. The red bar represents feature important of the forest, along with their inter-tree variability.

Since the proposed approach targets the detection of elephant rumbles captured by the ELOC deployment unit, we constructed the synthesized dataset by replaying the above elephant rumble recordings and negative dataset using a subwoofer. Earlier work has shown [11]

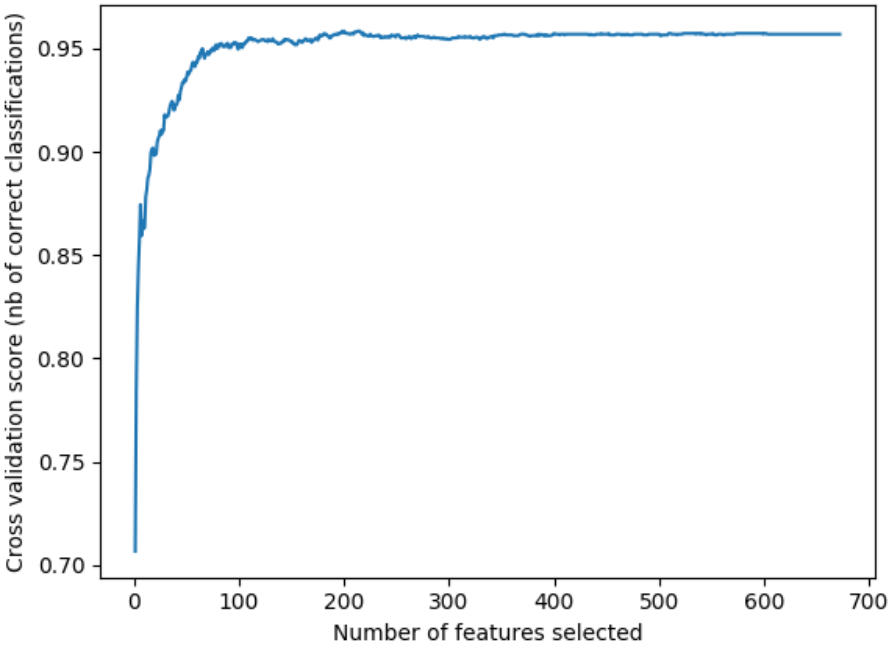

Figure 5.10: The cross-validation score variation with the number of features selected for features extracted from database 1.

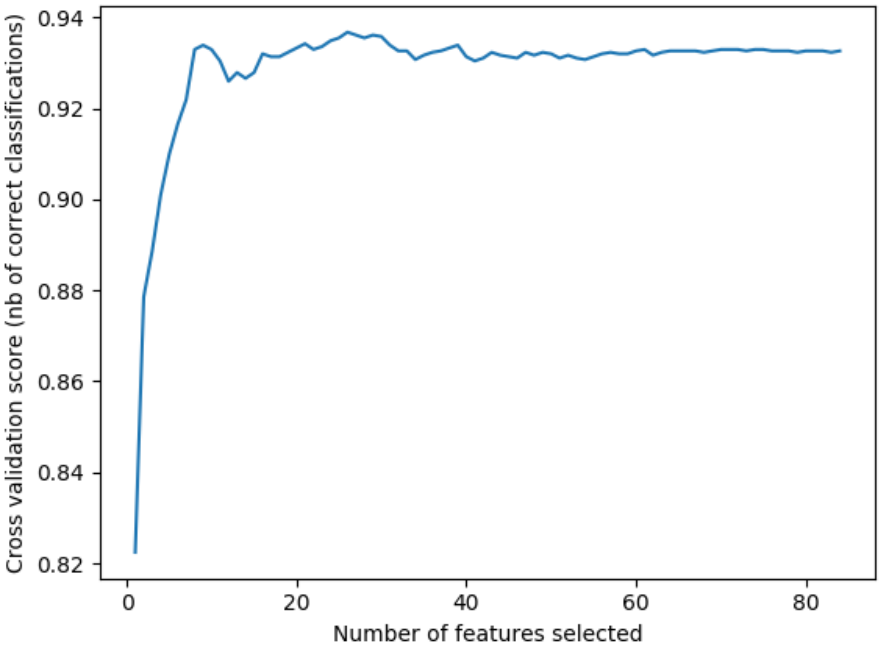

Figure 5.11: The cross-validation score variation with the number of features selected for features extracted from database 2.

that subwoofers can replay elephant sounds that include fundamental frequency components in the infrasonic range with sufficient output power to emulate a real elephant.

Therefore, the two datasets used by this study is as follows:

- Dataset-1:Replayed elephant rumbles and negative dataset captured using ELOC node.
- Dataset-2:A collection of elephant rumbles from Smithsonian conservation biology institute and negative dataset (infrasound recordings) taken from the internet.

5.2.1.1 Training and testing datasets

- Dataset-1 is divided into two parts for classifier evaluation and feature selection module tuning. (75 positive samples and 75 negative sample were used for the training while 25 positive and 25 negative samples used for testing.)(This dataset is not used for the classifier training)
- Dataset-2 is divided into two parts for classifier training and testing. (104 positive samples and 110 negative sample were used for the training while 75 positive and 100 negative samples used for testing)
- Each data is processed with a 1-second window with overlapping. (0.6-second shift between consecutive windows).

5.2.1.2 Testing dataset with noise

First, we normalized the testing samples in dataset-2 to -25dB; then noise samples are normalized to -15dB, -25dB, -35dB, -45dB, -55dB, -65dB, and -75dB. Finally, normalized noise clips and testing sample are mixed programmatically to obtain the desired level of SNR_{dB}; -10dB, 0dB, 10dB, 20dB, 30dB, 40dB and 50dB respectively. SNR_{db} is expressed in equation (5.1) where P is the average power, $P_{signal,dB} = 10\log_{10}(P_{signal})$ and $P_{noise,dB} = 10\log_{10}(P_{noise})$.

$$SNR_{dB} = P_{signal,dB} - P_{noise,dB} \tag{5.1}$$

5.2.2 Evaluation results

We evaluate the classification accuracy with both datasets. Table 5.1 and Table 5.2 represent the classification accuracy with regard to dataset-1 and dataset-2 respectively. It is visible that, the proposed approach achieves 93% classification accuracy with dataset-2 and 82% classification accuracy with dataset-1. Thus, it could be stated that it performs well (82%) in detecting replayed versions of elephant rumbles.

Table 5.1: Prediction performance with dataset-1

	Precision	Recall	F1-score
Negative	0.79	0.94	0.84
Positive	0.92	0.69	0.79
Avg/Total	0.84	0.84	0.82

Table 5.2: Prediction performance with dataset-2

	Precision	Recall	F1-score
Negative	0.91	0.99	0.95
Positive	0.99	0.84	0.90
Avg/Total	0.94	0.93	0.93

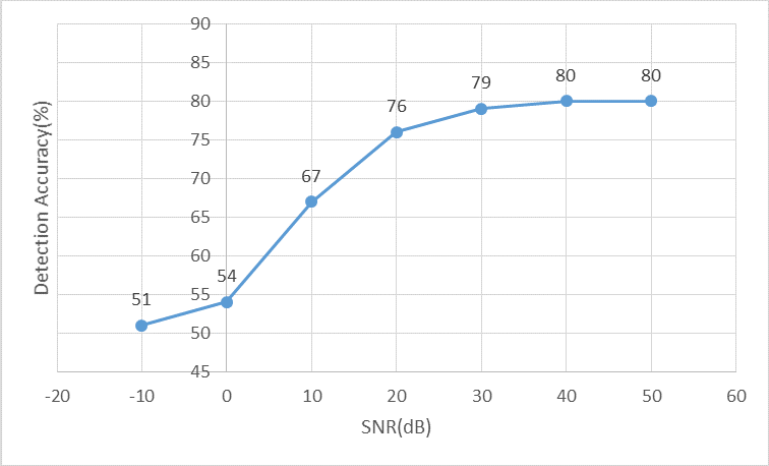

Figure 5.12: Detection accuracy with white noise under different SNR

As described before here we only consider the frequency range from 10 to 150 Hz, this means that the approach depends on the fundamental infrasonic components of elephant rumbles and the first few harmonics. Furthermore, these results were achieved with the 600 Hz sampling rate for signal recording and processing, thus proving that the proposed approach is efficient enough to operate on top of the resource limited hardware platform of the ELOC deployment unit.

To evaluate the classification performance under the noisy conditions, we conducted the test experiments with 5 noise types under different SNR. Figures 5.12 - 5.16 represent the detection accuracy under different noise types. It is clear that the proposed approach performs well in the noise conditions when the SNR_{db} is higher than 10.

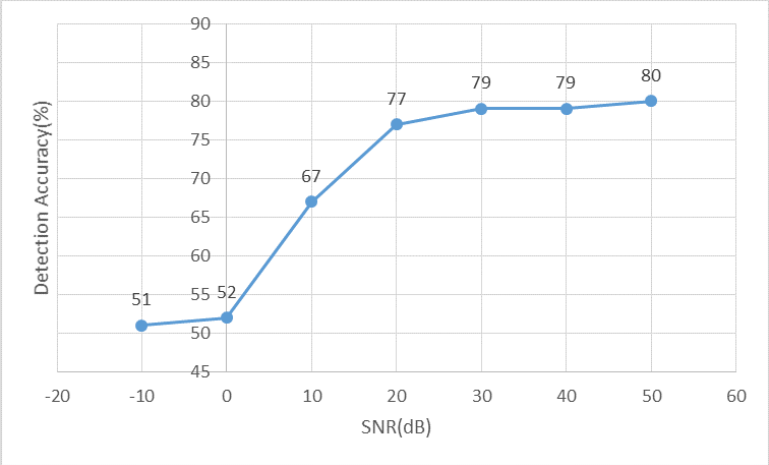

Figure 5.13: Detection accuracy with pink noise under different SNR

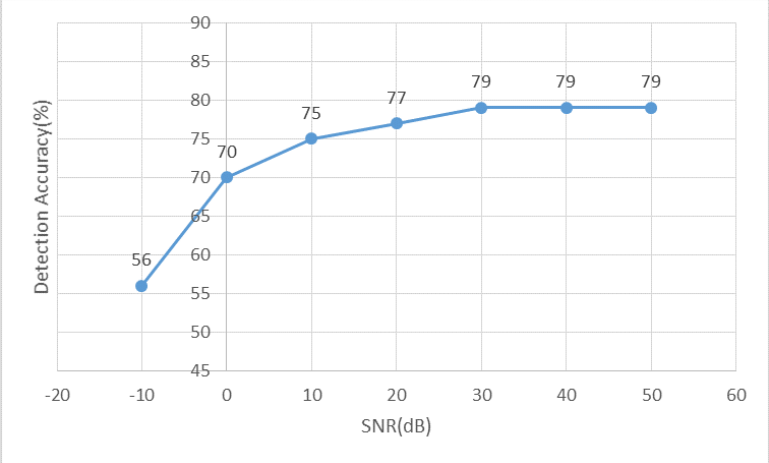

Figure 5.14: Detection accuracy with petrol engine noise under different SNR

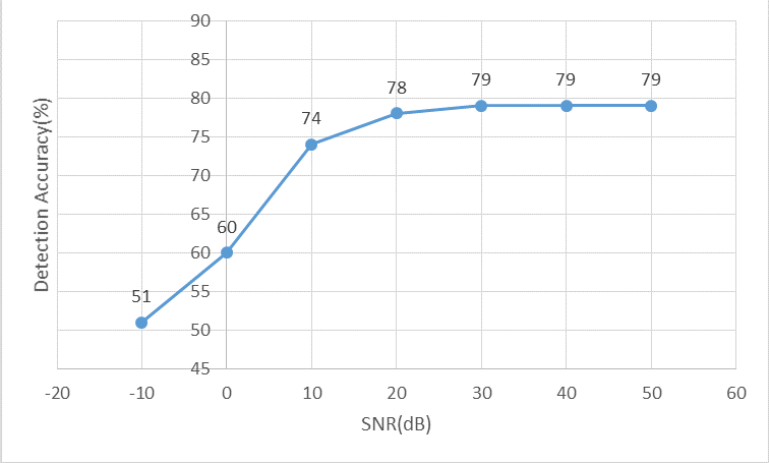

Figure 5.15: Detection accuracy with rain noise under different SNR

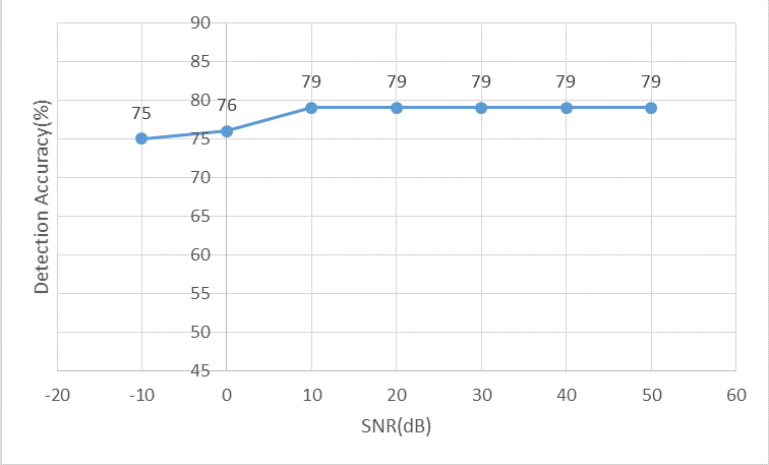

Figure 5.16: Detection accuracy with helicopter noise under different SNR

Chapter 6

Conclusions

6.1 Introduction

This chapter includes a discussion of the research problem, limitations of the current work and implications for further research.

6.2 Conclusions

This study presents the complete sound processing pipeline for infrasonic elephant rumble detection, which is capable of operating in the noisy natural situations, on top of the resource-limited low-cost hardware which is the ELOC node. During the study, the aim was to assess the feasibility of extending the ELOC node as an automatic elephant detection system and formulate a mechanism to extend the ELOC node as an automatic elephant detection system. According to the evaluation results, we can state that the proposed approach has achieved this aim of the study.

This study also contributes to the domain of digital signal processing since the study is the first attempt of wavelet-based feature extraction in the domain of infrasound elephant rumble detection. The study has shown that wavelet-based feature extraction is usable in elephant rumble detection.

During this study, we found out the main drawback of the MFCC based elephant rumble detection method proposed by Ranathunga et al. [14]. The evidence from the current study falsifies the below conclusion of that study "Although the infrasonic range is between 0 Hz to 20hz, Taking the upper limit as 300Hz has had a high impact on the accuracy of detection. We observed a reduction in accuracy when this range is decreased".

6.3 Limitations and Implications for further research

Since this approach has been mainly tested on the detection of replayed versions of elephant rumbles and on the influence of the artificially generated noise, comprehensive testing and tuning of the proposed method in the natural environment is the fundamental requirement to extend the ELOC node as the final system to tame the HEC.

The proposed approach only attempts to identify whether the dominant infrasonic signal of a given period emits from the elephant or not. Therefore, this approach is not efficient when other powerful infrasounds with a similar frequency range of elephant rumbles are present. Thus further research can be carried out to overcome this limitation.

Furthermore, this approach only considers the detection of elephant infrasound rumbles since it travels a more considerable distance from the sound source than other elephant calls. However, elephants emit a variety of calls, from the infrasound range to human audible range. These calls can also be utilized when building a complete elephant detection system. Thus, further studies can be carried out, taking other elephant call types into consideration.

Bibliography

[1] "Elephant: an Endangered Species." [Online]. Available: http://www.bagheera.com/inthewild/van_anim_elephant.htm

[2] "Issues_ Human-elephant conflict _ WWF." [Online]. Available: http://wwf.panda.org/what_we_do/endangered_species/elephants/asian_elephants/areas/issues/elephant_human_conflict/

[3] "CCRSL.ORG — Tracking Elephants." [Online]. Available: http://ccrsl.gadola.com/Programs/tracking-elephants

[4] J. D. WOOD, B. MCCOWAN, W. R. L. JR., J. J. VILJOEN, and L. A. HART, "Classification of african elephant loxodonta africana rumbles using acoustic parameters and cluster analysis," *Bioacoustics*, vol. 15, no. 2, pp. 143–161, 2005. [Online]. Available: https://doi.org/10.1080/09524622.2005.9753544

[5] S. Nair, R. Balakrishnan, C. S. Seelamantula, and R. Sukumar, "Vocalizations of wild asian elephants (elephas maximus): Structural classification and social context," *The Journal of the Acoustical Society of America*, vol. 126, no. 5, pp. 2768–2778, 2009. [Online]. Available: https://doi.org/10.1121/1.3224717

[6] J. Prince, "Surveillance and tracking of elephants using vocal spectral information," vol. 03, pp. 664–671, 05 2014.

[7] L. Seneviratne and G. Rossel, "Elephant Infrasound Calls as a Method for Electronic Elephant Detection." *Channels*, pp. 1–7, 2004. [Online]. Available: http://sasiri.tripod.com/Items/IEE Paper_Elephant Detection_Final3.pdf

[8] D. Henley and G. Hoidale, "Attenuation and dispersion of acoustic energy by atmospheric dust," *The Journal of the Acoustical Society of . . .*, vol. 88002, no. May, pp. 437–445, aug 1973. [Online]. Available: http://asa.scitation.org/doi/10.1121/1.1913597 http://asadl.org/jasa/resource/1/jasman/v54/i2/p437_s1

[9] T. L. Szabo, "Time domain wave equations for lossy media obeying a frequency power law," *The Journal of the Acoustical Society of America*, vol. 96, no. 1, pp. 491–500, jul 1994. [Online]. Available: http://asa.scitation.org/doi/10.1121/1.410434

[10] K. B. Payne, M. Thompson, and L. Kramer, "Elephant calling patterns as indicators of group size and composition: The basis for an acoustic monitoring system," *African Journal of Ecology*, vol. 41, no. 1, pp. 99–107, 2003. [Online]. Available: http://www.birds.cornell.edu/brp/elephant/PDFs/ Payneetal_AfrJEcol2003.pdf

[11] P. Dabare, C. Suduwella, A. Sayakkara, D. Sandaruwan, C. Keppitiyagama, K. De Zoysa, K. Hewage, and T. Voigt, "Listening to the Giants," in *Proceedings of the 6th ACM Workshop on Real World Wireless Sensor Networks - RealWSN '15*. New York, New York, USA: ACM Press, 2015, pp. 23–26. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2820990.2821000

[12] "INFILTEC: The Inexpensive Infrasound Monitor Project. - simple microbarograph design for DIY." [Online]. Available: http://www.infiltec.com/Infrasound@home/

[13] A. Sayakkara, N. Jayasuriya, T. Ranathunga, C. Suduwella, N. Vithanage, C. Keppitiyagama, K. De Zoysa, K. Hewage, and T. Voigt, "Eloc: Locating Wild Elephants using Low-cost Infrasonic Detectors," in *13th International Conference on Distributed Computing in Sensor Systems*, 2017. [Online]. Available: http://uu.diva-portal.org/smash/record.jsf?pid=diva2%3A1158106&dswid=3013 http://uu.diva-portal.org/smash/record.jsf?pid=diva2%3A1158106&dswid=1560

[14] T. Ranathunga, N. Jayasuriya, K. Gunawardana, and C. Silva, "Poster Abstract : Resource-Efficient Detection of Elephant Rumbles," pp. 17–19, 2017.

[15] C. Dissanayake, R. Kotagiri, M. Halgamuge, B. Moran, and P. Farrell, "Propagation constraints in elephant localization using an acoustic sensor network," in *ICIAFS 2012 - Proceedings: 2012 IEEE 6th International Conference on Information and Automation for Sustainability*. IEEE, sep 2012, pp. 101–105. [Online]. Available: http://ieeexplore.ieee.org/document/6419889/

[16] M. M, M. Devaki, and R. Scholar, "Elephant Localization And Analysis of Signal Direction Receiving in Base Station Using Acoustic Sensor Network," *International Journal of Innovative Research in Computer and Communication Engineering (An ISO Certified Organization)*, vol. 3297, no. 2, 2007. [Online]. Available: www.ijircce.com

[17] M. Zeppelzauer, A. S. Stöger, and C. Breiteneder, "Acoustic detection of elephant presence in noisy environments," in *Proceedings of the 2nd ACM international workshop on Multimedia analysis for ecological data - MAED '13*. New York, New York, USA: ACM Press, 2013, pp. 3–8. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2509896.2509900

[18] P. J. Venter and J. J. Hanekom, "Automatic detection of African elephant (Loxodonta africana) infrasonic vocalisations from recordings," *Biosystems Engineering*, vol. 106, no. 3, pp. 286–294, jul 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S153751101000070X

[19] A. E. Villanueva-Luna, A. Jaramillo-nuñez, D. Sanchez-lucero, C. M. Ortiz-lima, J. G. Aguilar-soto, A. Flores-gil, and M. May-alarcon, "De-Noising Audio Signals Using MATLAB Wavelets Toolbox," in *Engineering Education and Research Using MATLAB*, 2011, pp. 25–54. [Online]. Available: http://cdn.intechopen.com/pdfs/21375.pdf

[20] M. Sifuzzaman, M. R. Islam, and M. Z. Ali, "Application of Wavelet Transform and its Advantages Compared to Fourier Transform," *Journal of Physical Sciences*, vol. 13, pp. 121–134, 2009. [Online]. Available: www.vidyasagar.ac.in/journal

[21] A. Kandaswamy, C. S. Kumar, R. P. Ramanathan, S. Jayaraman, and N. Malmurugan, "Neural classification of lung sounds using wavelet coefficients," *Computers in Biology and Medicine*, vol. 34, no. 6, pp. 523–537, sep 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0010482503000921

[22] J. S. Lim and A. V. Oppenheim, "Advanced topics in signal processing," *Advanced topics in signal processing*, pp. 289 – 337, 1987. [Online]. Available: https://dl.acm.org/citation.cfm?id=42745&CFID=1021698089&CFTOKEN=33792897

[23] O. Rioul and M. Vetterli, "Wavelets and Signal Processing," *IEEE Signal Processing Magazine*, vol. 8, no. 4, pp. 14–38, 1991.

[24] U. Orhan, M. Hekim, and M. Ozer, "EEG signals classification using the K-means clustering and a multilayer perceptron neural network model," *Expert Systems with Applications*, vol. 38, no. 10, pp. 13 475–13 481, sep 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417411006762

[25] O. Farooq and S. Datta, "Robust features for speech recognition based on admissible wavelet packets," p. 1554, 2001. [Online]. Available:

http://digital-library.theiet.org/content/journals/10.1049/el_20011029 http://digital-library.theiet.org/content/journals/10.1049/el_20011029%0A

[26] F. Z. Göü, B. Karlk, and G. Harman, "Classification of Asthmatic Breath Sounds by Using Wavelet Transforms and Neural Networks," *International Journal of Signal Processing Systems*, vol. 3, no. 2, 2014. [Online]. Available: http://www.ijsps.com/index.php?m=content&c=index&a=show&catid=36&id=125

[27] G. Bianchi and R. Sorrentino, *Electronic filter simulation & design*. McGraw-Hill, 2007. [Online]. Available: https://books.google.com/books?id=5S3LCIxnYCcC&pg=PT32&dq=Butterworth-approximation+maximally-flat#v=onepage&q=Butterworth-approximation maximally-flat&f=false

[28] H. He, Y. Tan, and Y. Wang, "Optimal base wavelet selection for ECG noise reduction using a comprehensive entropy criterion," *Entropy*, vol. 17, no. 9, pp. 6093–6109, sep 2015. [Online]. Available: http://www.mdpi.com/1099-4300/17/9/6093/

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2012. [Online]. Available: http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html http://dl.acm.org/citation.cfm?id=2078195%5Cnhttp://arxiv.org/abs/1201.0490

[30] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research (JMLR)*, vol. 3, no. 3, pp. 1157–1182, 2003. [Online]. Available: http://www.jmlr.org/papers/volume3/guyon03a/guyon03a.pdf

[31] B. Mcfee, C. Raffel, D. Liang, D. P. W. Ellis, M. Mcvicar, E. Battenberg, and O. Nieto, "librosa: Audio and Music Signal Analysis in Python," *PROC. OF THE 14th PYTHON IN SCIENCE CONF*, no. Scipy, pp. 1–7, 2015. [Online]. Available: https://bmcfee.github.io/papers/scipy2015_librosa.pdf

Appendices

Appendix A

Code Listings

```
1       def extract_features(signal,siganl_type, sample_rate, frame_length,
           hop_length, feature_class,final_feature_vector):
2           zeroCrossingRate = librosa.feature.zero_crossing_rate(y=
               signal, frame_length=frame_length, hop_length=hop_length)
3           # print ("zeroCrossingRate", zeroCrossingRate)
4           feature_divider(zeroCrossingRate, "zero_crossing_rate",
               final_feature_vector, feature_class, siganl_type)
5
6           chromaStft = librosa.feature.chroma_stft(y=signal, sr=
               sample_rate, tuning=0.3, n_fft=frame_length, hop_length=
               hop_length)
7           # print ("chromaStft", chromaStft) # tuning
8           feature_divider(chromaStft, "chroma_stft",
               final_feature_vector, feature_class, siganl_type)
9
10          chromaCqt = librosa.feature.chroma_cqt(y=signal, fmin=10,
               n_octaves=3, sr=sample_rate, hop_length=hop_length)
11          # print ("chromaCqt", chromaCqt) # fmin and n_octaves
12          feature_divider(chromaCqt, "chroma_cqt", final_feature_vector
               , feature_class, siganl_type)
13
14          chromaCens = librosa.feature.chroma_cens(y=signal, fmin=10,
               n_octaves=3, sr=sample_rate, hop_length=hop_length)
15          # print ("chromaCens", chromaCens)
16          feature_divider(chromaCens, "chroma_cens",
               final_feature_vector, feature_class, siganl_type)
17
18          melspectrogram = librosa.feature.melspectrogram(y=signal, sr=
               sample_rate, n_mels=30, fmin=10, fmax=150, n_fft=
               frame_length, hop_length=hop_length)
19          # print ("melspectrogram", melspectrogram)
20          feature_divider(melspectrogram, "melspectrogram",
               final_feature_vector, feature_class, siganl_type)
21
22          mfcc = librosa.feature.mfcc(y=signal, sr=sample_rate, n_mfcc
               =5, n_mels=30, fmin=10, fmax=150, n_fft=frame_length,
               hop_length=hop_length)
```

```
            # print ("mfcc", mfcc)
            feature_divider(mfcc, "mfcc", final_feature_vector,
               feature_class, siganl_type)

            rmse = librosa.feature.rmse(y=signal, frame_length=
               frame_length, hop_length=hop_length)
            # print ("rmse", rmse)
            feature_divider(rmse, "rmse", final_feature_vector,
               feature_class, siganl_type)

            spectralCentroid = librosa.feature.spectral_centroid(y=signal
                , sr=sample_rate, n_fft=frame_length, hop_length=
               hop_length)
            # print ("spectralCentroid", spectralCentroid)
            feature_divider(spectralCentroid, "spectral_centroid",
               final_feature_vector, feature_class, siganl_type)

            spectral_bandwidth = librosa.feature.spectral_bandwidth(y=
               signal, sr=sample_rate,n_fft=frame_length, hop_length=
               hop_length)
            # print ("spectralCentroid", spectral_bandwidth)
            feature_divider(spectral_bandwidth, "spectral_bandwidth",
               final_feature_vector, feature_class, siganl_type)

            spectral_contrast = librosa.feature.spectral_contrast(y=
               signal, sr=sample_rate, n_fft=frame_length, hop_length=
               hop_length, fmin=5,n_bands=5)
            # print ("spectral_contrast", spectral_contrast)
            feature_divider(spectral_contrast, "spectral_contrast",
               final_feature_vector, feature_class, siganl_type)

            spectral_rolloff = librosa.feature.spectral_rolloff(y=signal,
                sr=sample_rate, n_fft=frame_length, hop_length=hop_length
               ,roll_percent=0.85)
            # print ("spectral_rolloff", spectral_rolloff) #roll_percent
            feature_divider(spectral_rolloff, "spectral_rolloff",
               final_feature_vector, feature_class, siganl_type)

            poly_features = librosa.feature.poly_features(y=signal, sr=
               sample_rate, n_fft=frame_length, hop_length=hop_length,
               order=1)
            # print ("poly_features", poly_features)
            feature_divider(poly_features, "poly_features",
               final_feature_vector, feature_class, siganl_type)
```

```
        def feature_divider(feature_vector, feature_name,
           final_feature_vector, feature_class, siganl_type):

            # check if final_feature_vector empty or not
            if not final_feature_vector:
```

```
5               for _ in range(len(feature_vector[0])):
6               final_feature_vector.append({'feature_class': feature_class})
7
8               frame_count = len(final_feature_vector)
9
10              for feature in range(len(feature_vector)):
11              # print (feature_vector)
12              for frame in range(frame_count):
13              name = str(feature_name) + " - " + str(feature) + " - " + str
                    (siganl_type)
14              current_frame = final_feature_vector[frame]
15              current_frame[name] = feature_vector[feature][frame]
```

```
1       if __name__ == "__main__":
2               # Open feature file
3               csv_file_name = "../feature_files/
                    with_wavelet_600_filterd_5_150_order_9_lenth1_shift_0_6.
                    csv"
4               dataframe = csv_table_read(csv_file_name)
5               dataframe_remove_target = dataframe.drop(['feature_class'],
                    axis=1)
6
7               feature_vector = dataframe_remove_target.values[:]
8               target_vector = dataframe['feature_class'][:]
9               feature_names = dataframe.columns.tolist()
10
11              # Load Preprocessor
12
13              filename = 'QuantileTransformer.sav'
14              qt = pickle.load(open(filename, 'rb'))
15
16              # Load Feature Selector
17              filename = 'selector_LogisticRegression_test_1.sav'
18              selector = pickle.load(open(filename, 'rb'))
19
20              # prepossess
21              standard_feature_vector = qt.transform(feature_vector)
22
23              # select
24              selected_feature_vector = selector.transform(
                    standard_feature_vector)
25
26              # Train Classifier
27
28              # cross validation\n",
29
30              clf = LogisticRegression()
31              scores = cross_val_score(clf, selected_feature_vector,
                    target_vector, cv=3)
32              print("Cross Validation Accuracy: %0.2f (+/- %0.2f)" % (
```

```
                scores.mean(), scores.std() * 2))
33
34              # model train
35              clf.fit(selected_feature_vector, target_vector)
36
37              # save the model to disk
38              filename = 'classifier_out_ok.sav'
39              pickle.dump(clf, open(filename, 'wb'))
```

Appendix B

Tables

B.1 Top 50 features

$Rank$	$FeaturenamewithsignalType$	$Importance$
1	$melspectrogram - 19 - rm\_approx$	$3.4898999999999999E-2$
2	$mfcc - 1 - cA3$	$2.8339E-2$
3	$mfcc - 1 - rm\_approx$	$2.8213999999999999E-2$
4	$mfcc - 1 - rm\_approx\_cd1$	$2.8199999999999999E-2$
5	$melspectrogram - 8 - cd1$	$2.5072000000000001E-2$
6	$chroma\_cens - 7 - cd2$	$2.4284E-2$
7	$melspectrogram - 18 - row$	$2.3698E-2$
8	$mfcc - 0 - row$	$1.9717999999999999E-2$
9	$melspectrogram - 19 - cd1$	$1.8227E-2$
10	$spectral\_bandwidth - 0 - cA3$	$1.7437999999999999E-2$
11	$chroma\_cens - 0 - cd2$	$1.5233999999999999E-2$
12	$rmse - 0 - all\_recon$	$1.4393E-2$
13	$melspectrogram - 21 - cA3$	$1.4312E-2$
14	$poly\_features - 0 - all\_recon$	$1.4302E-2$
15	$melspectrogram - 19 - rm\_approx\_cd1$	$1.3606E-2$
16	$chroma\_stft - 9 - row$	$1.2425E-2$
17	$spectral\_contrast - 5 - row$	$1.2241E-2$
18	$chroma\_stft - 6 - cd1$	$1.0773E-2$
19	$melspectrogram - 23 - row$	$1.0763E-2$
20	$chroma\_cens - 2 - cd3$	$1.0463999999999999E-2$
21	$melspectrogram - 24 - cA3$	$1.0337000000000001E-2$
22	$spectral\_rolloff - 0 - rm\_approx\_cd1$	$9.1070000000000005E-3$
23	$melspectrogram - 29 - cd1$	$8.9960000000000005E-3$
24	$chroma\_cens - 8 - row$	$8.7080000000000005E-3$
25	$chroma\_stft - 3 - rm\_approx\_cd1$	$8.1069999999999996E-3$

$Rank$	$FeaturenamewithsignalType$	$Importance$
26	$chroma\_cens - 6 - cd2$	$7.4079999999999997E-3$
27	$chroma\_cens - 8 - all\_recon$	$6.8380000000000003E-3$
28	$melspectrogram - 27 - rm\_approx\_cd1$	$6.4780000000000003E-3$
29	$melspectrogram - 22 - rm\_approx\_cd1$	$6.4469999999999996E-3$
30	$chroma\_cens - 5 - cd2$	$6.156E-3$
31	$chroma\_cqt - 7 - cd2$	$6.1539999999999997E-3$
32	$chroma\_stft - 8 - cd2$	$6.0610000000000004E-3$
33	$melspectrogram - 25 - cA3$	$5.6270000000000001E-3$
34	$melspectrogram - 2 - all\_recon$	$5.5890000000000002E-3$
35	$chroma\_cens - 10 - rm\_approx$	$5.2659999999999998E-3$
36	$melspectrogram - 1 - all\_recon$	$5.2610000000000001E-3$
37	$chroma\_cens - 3 - row$	$4.8560000000000001E-3$
38	$chroma\_stft - 6 - cA3$	$4.8500000000000001E-3$
39	$melspectrogram - 19 - cd3$	$4.7869999999999996E-3$
40	$melspectrogram - 19 - cA3$	$4.705E-3$
41	$chroma\_stft - 3 - cd1$	$4.6670000000000001E-3$
42	$rmse - 0 - rm\_approx\_cd1$	$4.4910000000000002E-3$
43	$melspectrogram - 28 - cd1$	$4.4879999999999998E-3$
44	$chroma\_cens - 2 - cA3$	$4.4019999999999997E-3$
45	$chroma\_cens - 10 - cd3$	$4.3880000000000004E-3$
46	$melspectrogram - 20 - rm\_approx\_cd1$	$4.3470000000000002E-3$
47	$melspectrogram - 7 - cd1$	$4.1929999999999997E-3$
48	$chroma\_cens - 4 - cd3$	$4.1339999999999997E-3$
49	$chroma\_cqt - 6 - cd2$	$4.0400000000000002E-3$
50	$chroma\_cens - 5 - rm\_approx$	$4.0379999999999999E-3$