ACOUSTIC DETECTION OF THE SHORT PULSE CALL OF BRYDE'S WHALES USING TIME DOMAIN FEATURES AND HIDDEN MARKOV MODELS

MASTER THESIS

Author:

GAELLE VANESSA WACHE NGATEU

A thesis presented for the degree of
Master of Engineering in Electronic Engineering in the
Faculty of Engineering at Stellenbosch University

Supervisor: Prof. Daniel Jaco J. VERSFELD

March 2020

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own original work, and that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Gaelle Vanessa Wache Ngateu

March 2020

STELLENBOSCH UNIVERSITY

Abstract

ACOUSTIC DETECTION OF THE SHORT PULSE CALL OF BRYDE'S WHALES USING TIME DOMAIN FEATURES AND HIDDEN MARKOV MODELS

Gaelle Vanessa WACHE NGATEU
Department of Electrical and Electronic Engineering, University of Stellenbosch
Private Bag X1, Matieland 7602, South Africa.

Thesis: MSc (Electronic Engineering)

March 2020

The biological group of cetaceans is frequently studied nowadays as passive acoustic monitoring (PAM) is commonly used to extract the acoustic signals produced by cetaceans, in the midst of noise sounds made by either man during shipping, gas and oil explorations or by natural sounds like seismic surveys, wind and rain. In this research work, the acoustic signal of short pulse call of inshore Bryde's whales is detected using time domain features and hidden Markov models (HMM). HMM is deployed as a detection and classification technique due to its robustness and low time complexity during the detection phase. However, some parameters such as the choice of features to be extracted from the acoustic short pulse call of inshore Bryde's whales, the frame durations of each call and the number of states used in the model affect the performances of the automated HMM. Therefore, to measure performances like sensitivity, accuracy and false positive rate of the automated HMM; three time domain features (average power, mean and zero-crossing rate) were extracted from a dataset of 44hr26mins recordings obtained close to Gordon's bay in False bay, South

Africa. Moreover, to extract these features the frame durations of each vocalisation was varied thrice; 1 ms, 5 ms and 10 ms. Also, the HMM used three different number of states (3 states, 5 states and 10 states) which were varied independently so as to evaluate the HMM. On an overall performance, the HMM yields best performances when it uses 10 states with a short frame duration of 1 ms and average power as the extracted feature. With regard to this, the automated HMM shows to be 99.56% sensitive, and dependable as it exhibits a low false positive rate of 0.1 with average power inferred as the best time domain feature used to detect the short pulse call of inshore Bryde's whales using the HMM technique.

UNIVERSIEIT STELLENBOSCH

Opsomming

Die akoestiese opsporing van die walvis van Bryde se walvisse met behulp van tyddomeinfunksies en verborge Markov-modelle.

Gaelle Vanessa WACHE NGATEU

Departement Elektriese en Elektroniese Ingenieurswese, Universiteit van Stellenbosch, Privaatsak X1, Matieland 7602, Suid Afrika.

Verhandeling: MSc (Elektroniese Ingenieurswese)

Maart 2020

Die biologiese groep valkaansoorte word deesdae gereeld bestudeer as passiewe akoestiese monitering (PAM) word gereeld gebruik om die akoestiese seine wat deurwa lvisse, te midde van geraasgeluide wat deur een of ander man gemaak is tydens die vervoer, gas en olieverkenning of deur natuurlike klanke soos seismiese opnames, wind en ren. in hierdie navorsingswerk, die akoestiese sein van 'n kort polsslag van Bryde se walvisse word opgespoor met behulp van tyddomeinfunksies en verborge Markov-modelle (HMM). HMM word gebruik as 'n opsporing- en klassifikasietegniek vanwe die robuustheid en lae tydskompleksiteit tydens die opsporingsfase. Sommige parameters soos die keuse van funksies wat uittrek uit die akoestiese kortpulsoproep van Bryde's aan wal walvisse, die raamduur van elke oproep en die aantal toestande wat in die model gebruik word benvloed die optredes van die outomatiese HMM. Daarom, om uitvoerings te meet soos sensitiwiteit, akkuraatheid en vals positiewe tempo van die outomatiese HMM; drie keer domeineienskappe (gemiddelde drywing, gemiddelde en nul kruising) is onttrek uit a datastel van opnames van 44 uur 26 min naby Gordonsbaai in Valsbaai, Suid Afrika. Om hierdie kenmerke te onttrek, is die raamduur van elke vokalisasie ook onthul was drie keer gevarieerd; 1 ms, 5 ms en 10 ms. Die HMM het ook drie verskillende getalle gebruik van state (3 state, 5 state en 10 state) wat onafhanklik van mekaar verskil het evalueer die HMM. Op ’n algehele prestasie lewer die HMM die beste prestasies as dit 10 toestande gebruik met ’n kort raamduur van 1 ms en gemiddelde drywing as die onttrek funksie. Met betrekking hiertoe blyk die outomatiese HMM 99 : 56% te wees sensitief en betroubaar, aangesien dit ’n lae vals positiewe koers van 0: 1 met die gemiddelde toon krag afgelei as die beste tyddomeinfunksie wat gebruik word om die kort polsoproep van Kry die walvisse van Bryde met behulp van die HMM-tegniek.Die biologiese groep valkaansoorte word deesdae gereeld bestudeer as passiewe akoestiese monitering (PAM) word gereeld gebruik om die akoestiese seine wat deur walvisse, te midde van geraasgeluide wat deur een of ander man gemaak is tydens die vervoer, gas en olieverkenning of deur natuurlike klanke soos seismiese opnames, wind en ren. in hierdie navorsingswerk, die akoestiese sein van ’n kort polsslag van Bryde se walvisse word opgespoor met behulp van tyddomeinfunksies en verborge Markov-modelle (HMM). HMM word gebruik as ’n opsporing- en klassifikasietegniek vanwe die robuustheid en lae tydskompleksiteit tydens die opsporingsfase. Sommige parameters soos die keuse van funksies wat uittrek uit die akoestiese kortpulsoproep van Bryde’s aan wal walvisse, die raamduur van elke oproep en die aantal toestande wat in die model gebruik word benvloed die optredes van die outomatiese HMM. Daarom, om uitvoerings te meet soos sensitiwiteit, akkuraatheid en vals positiewe tempo van die outomatiese HMM; drie keer domeineienskappe (gemiddelde drywing, gemiddelde en nul kruising) is onttrek uit a datastel van opnames van 44 uur 26 min naby Gordonsbaai in Valsbaai, Suid Afrika. Om hierdie kenmerke te onttrek, is die raamduur van elke vokalisasie ook onthul was drie keer gevarieerd; 1 ms, 5 ms en 10 ms. Die HMM het ook drie verskillende getalle gebruik van state (3 state, 5 state en 10 state) wat onafhanklik van mekaar verskil het evalueer die HMM. Op ’n algehele prestasie lewer die HMM die beste prestasies as dit 10 toestande gebruik met ’n kort raamduur van 1 ms en gemiddelde drywing as die onttrek funksie. Met betrekking hiertoe blyk die outomatiese HMM 99 : 56% te wees sensitief en betroubaar, aangesien dit ’n lae vals positiewe koers van 0: 1 met die gemiddelde toon krag afgelei as die beste tyddomeinfunksie wat gebruik word om die kort polsoproep van Kry die walvisse van Bryde met behulp van die HMM-tegniek.

***Dedication*.**

To all those who stumble on rocky grounds and lose hope, may they remember that God is always faithful and with Him everything is possible. For, He is the Way, the Truth and the Life!

Acknowledgements

I feel humbled to share words of gratitude to the following category of people who have played exceptional roles in my life, towards the completion of this piece of work.

I give immense thanks to God for His unconditional love, continuous strength, guidance and inspiration throughout my studies. It has truly not been by my might, but by His grace and special support. Lord God, with you I grow in faithfulness and I shall forever give you all the glory!

Distinct words of thankfulness goes to my parents for the kind of education they instilled in me, which gave me more reasons to seek for better and integral education. They have always been there to encourage and support my ambitions. Papa and Mama, I can't thank you enough, for you opened my door to all good opportunities.

I remain indebted to the Mandela-Rhodes Foundation (MRF) without who I will not have had the opportunity to start a Master's program. MRF has fully sponsored my 2 years of Master's study at Stellenbosch University. Apart from financial support, MRF always provided a sphere of family. Thank you MRF!

I am profoundly grateful to my supervisor, Prof. D.J.J. Versfeld for taking me throughout the world of academic research in a very patient and understanding manner. It has been a new and rocky path for me, but he stood firm with me and for me, till the completion of this work. I very much appreciate your guidance.

I am exceptionally grateful to my dear friend and mentor Dr. Igor Miranda. You have been more than caring, educative and supportive. I learnt and continue to learn so much from you. Thanks for always being you!

Special thanks goes to my friend Mr. Babalola Oluwaseyi for his endless support both within and without the academic research milieu. Seyi, thank you a lot!

I whole-heartedly acknowledge my dearest friend and sister Miss Alvine Ghomsi, and all my pool of friends for their unceasing care and encouragement. You all have been a true source of motivation and inspiration. Thank you for being there!

Words will surely never explain how grateful I am to my siblings, cousins, aunts and uncles, and my extended family for their continuous prayers and moral assistance during this period of my studies.

My Christian family at St.Nicholas Church, from Fr.Wim to the parishioners, thank you so much for the spiritual support.

The least but not the last words of appreciation goes to all those who assisted me in whatsoever way, directly or indirectly and whom today I am omitting to mention. I have you all at heart and I say thank you a bunch!

Contents

List of Figures

List of Tables

Abbreviations

MP	**M**arkov **P**roperty
MC	**M**arkov **C**hain
HMM	**H**idden **M**arkov **M**odel
GMM	**G**aussian **M**iture **M**odel
SVM	**S**upport **V**ector **M**achine
VQ	**V**ector **Q**uantisation
ANN	**A**rtificial **N**eural **N**etwork
SCF	**S**pectrogram **C**orrelator **F**ilter
FFT	**F**ast **F**ourier **T**ransform
STFT	**S**hort **T**ime **F**ourier **T**ransform
DCT	**D**iscrete **C**osine **T**ransformation
EM	**E**xpectation **M**aximisation
ML	**M**aximum **L**ikelihood
MFCC	**M**el **F**requency **C**epstral **C**oefficient
LPC	**L**inear **P**redictive **C**oefficient
ZCR	**Z**ero **C**rossing **R**ate
FPR	**F**alse **P**ositive **R**ate

Chapter 1

Introduction

1.1 Introduction

Acoustic signal processing is based on the principal concept of extracting critical information from noisy, ambiguous measurement data, while signal processing is the process of acquiring, storing, displaying, and generating signals [2]. It implies extracting information from signals and converting them to wanted information, thereby playing an important role in the practice of all aspects of acoustics. In particular, acoustic signals are responses to a specific stimulus, which are used by most marine mammals to relate with the environment. The biological group of cetaceans like whales, dolphins and porpoises communicate with each other and their surroundings, identify their sexual partners and even echo-locate preys due to the acoustic signals they produce. The acoustic signals or vocalisations produced by some species are known as moans, clicks, and pulses are contain different kinds of noise like anthropogenic noise (man made, shipping, pile driving), noise from ocean fauna (whales, fish) and natural non biological noise (wind, rain, waves, seismic). The presence of noise in these vocalisations makes it difficult to identify a cetacean by it's acoustic signal. So, there is a need to detect specific marine mammals vocalisation and an attempt to localise them relative to a towed array. This could be achieved by carrying out passive acoustic monitoring (PAM) as it helps in mitigating these various noises by extracting the signal from the noise, improving the signal-to-noise ratio, interpreting the detected data in real time based on the experience level of the passive acoustic system or detector.

Moreover, PAM is used to detect acoustic signals in the exclusion zone and the detection range is dependent on acoustic background noise levels which vary between the types of vessels. Usually, acoustic sensors are used to survey and monitor marine mammals in the ecosystem. Sound recorders like the acoustic sensors are deployed in the field to collect acoustic data which may last for hours, days or weeks. These recordings are processed after collection to extract ecological and acoustical data of interest, such as detecting calls from particular marine mammals. In addition, the recorded dataset can later be used to estimate species habitation, abundance, population density and group composition, track marine mammals' behavioural spatial and temporal patterns, and measure acoustic representations for biodiversity metrics [3].

The task of this research is to build a detector to find a specific signal, being the short pulse call of inshore Bryde's whales. Acoustic signals are similar to speech, therefore in order to detect wanted acoustic signals, speech recognition approaches are borrowed in this context. For instance, the statistical based approach has shown to be the most suitable detection technique given its robustness, efficiency and reduced computational time [4], [5]. The statistical model employed in this study is the Hidden Markov Models (HMM), which is also in the class of probabilistic graphical model. The HMM is used for predicting a sequence of unknown (hidden) variables given a set of acoustic characteristics, also known as the observations. The acoustic characteristics obtained as the dataset is traditionally accessed by an ordinary visual observation process [6], and it usually takes a longer period of time. More so, the collected dataset has less accurate information due to man-made activities such as geographical seismic surveys, shipping traffic, and offshore explorations. On the other hand, passive acoustic monitoring is employed for such acoustics data collection because it deals with the use of hydrophones to monitor marine mammals movements, build vocalisation repertoire, and responses to anthropogenic activities. Also, the acoustic dataset collected is usually large and difficult to analyse and detect by human experts, thus detection and classification techniques like HMM is used.

According to [7], there are two main allopatric forms of Bryde's whales as members of the Balaenopteridae family in the class of cetaceans. The inshore allopatric form of Bryde's whale scientifically called *Balaenoptera edeni* was discovered in [8]. This inshore Bryde's whales with short pulse calls have been used in this thesis, and they are one of the world's most endangered marine mammals. The population growth of Brydes whales is quite small, though not ample information is provided regarding their population size estimate. Anthropogenic and marine activities are major threats preventing Bryde's whales from properly communicating, navigating, and feeding. Bryde's whales vocalisations from Eastern Tropical Pacific, Southern Caribbean, and Northwest Pacific report a frequency range from 21 Hz to 207 Hz and a time span from 0.35 to 2.8 s [9]. Other Bryde's whales calls from Southeast Brazil disclose more call types lasting for 0.8 to 1.5 s within a frequency range of 9 to 670 Hz. [9]. Moreover, for a case study of Bryde's whales in the Gulf of California, its sound is described to have fundamentals varying between $90 - 900$ Hz and a duration interval of 25 ms to 1.4 s [10].

Ocean noise are often at low frequency, just like the low frequency range of Bryde's whale sounds. Thus, the oceean noise tends to interrupt Bryde's whales possibility of properly listening to other environmental sounds, and this may eventually cause them to dislocate and even get extinct. It is therefore important to recognise this particular short pulse call of the inshore Bryde's whales' in the presence of many other sounds. A better approach to ensure this, is to begin by detecting their presence in the oceans. This implies developing a detector (HMM) characterised by the short pulse call of the Bryde's whale. As such, a recognition process could entirely be represented in a general flow diagram as depicted in Figure 1.1:

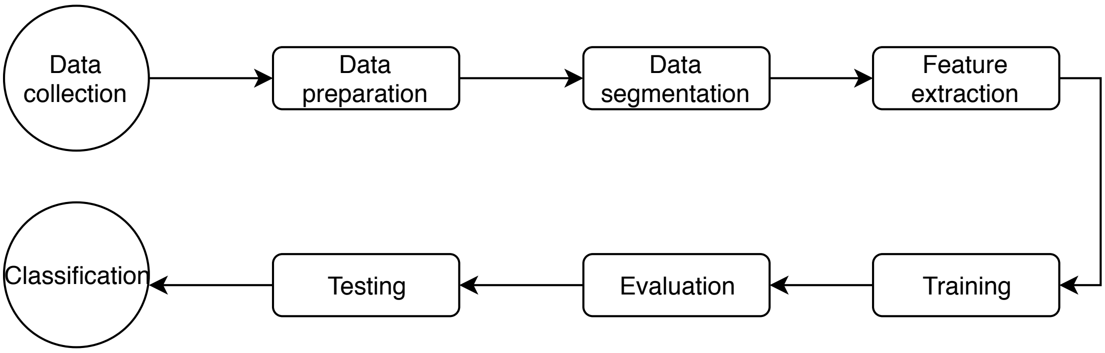

FIGURE 1.1: Simple flow diagram of a machine learning process

1.2 Research question

In order to detect this short pulse call of the Bryde's whale and using an HMM, the short pulse calls are extracted using time domain features to also reduce computational time during the HMM training process. Also, the extracted features depend on different frame durations of each call. The frame duration equally indicates the frame size of the dataset used in training the model. Moreover, the number of states used in the HMM plays a role in the recognition of these extracted features. So, the research question is stipulated as:

How do the time domain features, frame duration and number of states influence the HMM's performance for the detection of short pulse call of inshore Bryde's whales?

1.3 Research scope, objectives and contribution

The extent of this research is on detecting the short pulse call of inshore Bryde's whales in the time domain using the automated hidden Markov model. This is achieved by performing the following objectives:

1. To implement the time domain features using HMM for short pulse calls of inshore Bryde's whales. Here, three main time domain features (average power, mean, and zero-crossing rate) are analysed according to the frame durations and number of states used in the model.
2. To determine the best time domain extracted feature for short pulse calls of inshore Bryde's whales using the HMM technique.

Thus, a major contribution of this research is extracting features in the time domain for short pulse calls of inshore Bryde's whales using HMM.

1.4 Research outline

This research looks at developing an acoustic detector for the short pulse call of inshore Brydes whales based on time domain features and using hidden Markov models. To achieve this, the thesis has been organised in 5 chapters as follows:

Chapter 1 focuses on the general overview of speech recognition, passive acoustic monitoring and characteristics of Bryde's whales short pulse calls. The research question, scope, objectives, contribution and outline are also covered in this chapter.

Chapter 2 presents several types of speech recognition systems and related work by other researchers. Both the discrete HMM and continuous (Gaussian model) HMM are discussed as a proof of concept for hidden Markov model. Literature overview on feature extraction methods in frequency and time domains, the training, decoding and detecting processes have been discussed in this chapter. Also, a numerical example has been performed to better explain these processes.

Chapter 3 describes how the dataset was collected and analysed. This chapter also discusses the various stages involved to implement the GM-HMM in context with the research question. Thus, the steps to evaluate the performances of the automated model are mentioned in this section of the thesis.

Chapter 4 discusses the results obtained from implementing the hidden Markov model, based on a variation of the time durations, time domain features and the number of states using in the model. It presents comparisons that arise from changing these three main parameters in the course of training the model.

Chapter 5 concludes the thesis write-up together with some suggestions and future work to be compared with the current research work.

Chapter 2

Background and Literature review

2.1 Introduction

This section is centred on speech recognition systems related to other researchers' work, reason being that whale sounds detection and classification is quite a common pattern recognition exercise. Good knowledge of the acoustic characteristics of the sound aimed at detecting and classifying is a basic and vital step to consider, so feature extraction is also elaborated in this section. The chapter also discusses the general concept of hidden Markov models. It mainly entails recognising patterns from a dataset based on statistical information obtained from the sample pattern(s).

2.2 Speech recognition systems

Whale acoustic sounds are similar to speech signals and the process of studying and detecting these speech signals is also known as speech recogniton. Here, speech recognition approaches are borrowed to detect the wanted short pulse call of inshore Brydes whales. There are a couple of techniques used for processing, modelling and recognising these signals, some of them are:

Dynamic Time Warping (DTW); an algorithm for measuring similarity and optimal match under some restrictions, between two temporal sequences or curves which may vary in time or speed or lengths. This technique is quite useful as a distance measure for time series, for classification, and to find matching areas between two time series [11]. A huge amount of killer whale sounds were recorded from Marineland of Antibes in France. The recording was done with an HTI hydrophone straight into a hard-drive. Perceptual and DTW methods were implemented on 5 repeated call types. Each pair of peak contours were compared and a dissimilarity matrix was computed. This resulted to 5 sounds mismatched, thus an 88% match with perceptual results. On a second though longer trial, now considering both absolute frequencies and contours, with the same multidimensional scaling and k-means clustering, this time around a nearly-perfect match was obtained as only a single sound differed from what was perceived. These confirm that DTW can automatically classify killer whale sounds although it takes a longer period of time to determine the sound contours [12]. Effective classification by DTW has been evaluated based on four aspects: low frequency contour (LFC), the high frequency contour (HFC), their derivatives, and weighted sums. For the weighted sums, distances corresponding to LFC and HFC, LFC and its derivative, HFC and its derivative were computed. This evaluation was done on Northern Resident whale calls and four call types were mixed up and impossible to be separated by visual observation. This implied a more difficult test for DTW. Out of four algorithms examined, the results agreement with the perceptual data varied between $70 - 90\%$. A maximum of 90% was however a better result as compared to 98% in [12], given the complexity of the contours [13]. Other modelling techniques which serve as speech recognition systems are hidden Markov models, support vector machine, and artificial neural networks.

Artificial Neural Networks (ANN); these are computational models inspired by biological nervous systems like animal brains. ANNs imitate the operation, structure and function of these biological nervous systems in the course of processing information. It is also deployed as good tools for finding patterns which are quite complex and numerous for a human being to identify, extract and thereby teach a machine to recognise the patterns. Some whale sounds recordings from St.Lawrence Estuary and the blue whale calls were identified and classified using ANN. This was done by extracting their features based on short time Fourier transform (STFT) and wavelet packet transform (WPT). Spectrogram correlation and matched filter techniques were used to detect 3 different vocalisation categories. So, for 50 recurrent tests, the STFT and WPT based approaches were evaluated with a multi-layer perceptron (MLP) and resulted to an overall classification performance of 86.25% and 84.22% for STFT/MLP and WPT/MLP correspondingly [14]. ANN was used by [15], for marine mammal call discrimination. Specifically, for a dataset of about 1475 bowhead whale song notes and noise inclusive, the ANN classification technique resulted to an accuracy rate of 98.5%. This performance was twice better than other classifiers like the spectrogram correlator filter (SCF), which had previously been used to detect the same whale song. SCF had a better outcome in comparison with HMM and a time series matched filter as well.

Support Vector Machine (SVM); a linear model for classification and regression problems. Its basic principle is to create a more generalised separator or hyperplane which will make the separation between existing data more visible. So much so that, new datasets to be separated are easily classified. In other words, given labelled training data (supervised learning), the algorithm outputs an optimal hyper plane which classifies new data. Worthy to note is that it can also work well for non-linear problems. Some sound data are examined in [16] to detect individual humpback whales sounds using SVM. As characteristic parameters, cepstral coefficients were extracted from the dataset and used or trained in two ways: non-randomly and randomly. The former used biased training data since it only represented the start of the humpback song, thus the SVM failed to generalise and correctly detect when presented with the full song. Whereas, with the randomly trained feature vectors obtained from the whole song file, recognition was nearly proportional to its data size in the main file. This resulted to an accuracy rate of 99% (implying over 23% better than the case when non-random data was used), thereby outperforming the GMM 88% accuracy rate which was an earlier best result, trained on cepstral coeffcients. Furthermore, [17] presents a multi–class SVM known as the Class–Specific SVM to train and classify; not just as a binary classifier, but being able to categorise click vocalisations from Blainville's beaked whales from other clicks made by small odontocetes like delphinids, and human-made noise. The most visible separating hyperplanes for each category against the noise-only referenced category were observed with a reliable accuracy of 98% on the *mesoplodon* clicks.

In a spectrogram, frequency contours in a precise frequency interval at a particular time can be obtained using a frequency contour detection algorithm. Technically, it traces spectral peaks over time, groups them in a spectrogram according to how close their corresponding frequencies are. It then forms an even contour shape for the time being. This algorithm is generally used to detect frequency contours of both non-animal and animal sounds such as signature whistles of dolphins, clicks, moans, bird chirps et cetera. This method was used on a dataset of humpback whale sounds and later quantified. Interestingly, it performs well at recognising reverberating sounds and outputs a 3% false positive rate with a target missed-call rate of 25% [18]. In addition to this, time–frequency parameters (TFPs) can be obtained from these frequency contours so as to give the contour shape some features [19]. For the case study of 320 tonal calls of 4 different whales, the TFP features (the minimum and maximum frequencies, the start and end frequencies, the frequencies at 25%, 50%, 75% of the time duration, and the time duration) were extracted from the correct fundamental frequency contours detected on those four whale calls. The output of the classification with the TFPs and the SVM gives an average accuracy rate of 25% for all the species used [19].

Hidden Markov Models (HMM); a mathematical model in the probabilistic model category used to estimate a sequence of unspecified (hidden) variables with a list of observations. In this case, the mathematical model is said to be a Markov process whose future behaviour is independent of the past (hidden) events, but instead depends on the present behaviour (Markov property) [20]. Recognition process with the HMM technique deals with a sequence of desired features obtained from the sequence of observations. Bryde's whales produce unique sounds and their auditory range are pointed out by using characteristic features like MFCCs. According to [6], many Bryde's whales recordings were considered with different detectors developed at various frequency ranges. But, based on the HMM detector, these distinct Bryde's whales sounds were 77% sensitive to detection and 23% were detected automatically by HMM. Also, in a particular study of about 75 calls of killer whales recorded and classified both by perception and using other techniques like HMM and GMM. Cepstral coefficients (CC) were the extracted features. For the GMM, $1-6$ Gaussians were used with about $8-30$ CCs. Interestingly, it is observed that computation is not too sensitive on the number of Gaussians as it gave a result of 92% successful match over the 85% achieved by perceptual classification. This GMM results were obtained using 30 features and 2 Gaussians. On the other hand, the left-to-right HMM model made use of $1 - 4$ Gaussians and 3 variable parameters. Its set of states differed between $5 - 17$, with $18 - 42$ features. Comparatively with the human classification, it achieved over 90% for $18 - 42$ feautres with $9 - 17$ states. In addition to this, the HMM classification obtained a performance of over 95% while using $24 - 30$ features and $13 - 17$ states. For this case study, though the HMM classification outperformed the GMM, it also indicates that both techniques are quite successful to automatically classify killer whale sounds [21].

2.3 Hidden Markov Model

A hidden Markov model is also defined as a probabilistic graphical Markov model with an unobserved Markov chain. This implies that its state sequence is partially observed, so the word "hidden". Each set of states is associated with a probability distribution. Also, there exists two main topologies of HMM depending on their state connectivity. Ergodic HMM as it is fully connected to one another, that means it requires just a single step to transit from a state to another [20]. The Bakis HMM known as the Left-to-Right HMM alignment, whereby states transit mostly from a smaller-numbered state to a greater-numbered state. Also, the Left-to-Right HMM has a fundamental property such that to transit from a state i to another state j: $tr_{ij} = 0, \; j < i$.

2.3.1 Basic definitions

Definition 2.1. Stochastic Process (SP): It is an operation in which the value of any variable in the process changes over time in an unknown way. As such, it is likely categorised as a discrete time SP, continuous time SP, discrete variable process and continuous state SP.

Definition 2.2. Markov Property (MP): In statistics, the term Markov property refers to the memoryless property of a stochastic process. It is named after the Russian mathematician called Andrei Andreyevich Markov. The Markov property states that the conditional probability distribution of future states $(X_{t+1} = x_{t+1})$ of a stochastic process depends only on the current states and not on the sequence of events that happened before it. Equation 2.1 describes it as [20]:

$$P\left[X_{t+1} = x_{t+1} \mid X_1, \ldots, X_t = x_1, \ldots, x_t\right] = P\left[X_{t+1} = x_{t+1} \mid X_t = x_t\right]. \tag{2.1}$$

Definition 2.3. Markov Model: It is a sequence of random variables which obeys the Markov property. In other words, it is a stochastic process such that given its present behaviour, the future behaviour does not depend on the past behaviour.

Definition 2.4. Markov Chain: It is a Markov model whose random variables change over time and the state sequence is fully observable.

Definition 2.5. Initial probabilities: Given a sequence of N states:

$$X = \{x_1, x_2, \ldots, x_N\},$$

the *ith* state in X is denoted as x_i. Therefore, initial probability indicates which state is likely to be the initial state of the process or the distribution as:

$$\pi_i = P\left(x_i\right),$$

$$\pi_i = \pi_{x_1}.$$

The initial probability distribution is represented as:

$$\pi = \left(\pi_1, \pi_2, \ldots, \pi_N\right). \tag{2.2}$$

Initial state probabilities could be distributed through all the existing states, however it is worth noting that all π probabilities should absolutely sum up to 1;

$$\sum_{i=1}^{n} \pi_i = 1. \tag{2.3}$$

Definition 2.6. Transition probabilities: It is the probability of moving from a state to another in a step. In a Markov model, the probability of transiting to the next state depends on the current state only. Given a set of N states;

$$X = \{x_1, x_2, \ldots, x_N\},$$

the ith state in X being denoted as x_i. The transition probability denoted as tr_{ij}, is the probability that transiting to state j is only dependent on state i.

$$tr_{ij} = P\left(n_{t+1} = j \mid n_t = i\right). \tag{2.4}$$

Therefore, a transition probability matrix TR is usually represented as an $N \times N$ matrix shown in Equation 2.5:

$$TR = \begin{bmatrix} tr_{1,1} & tr_{1,2} & \ldots & tr_{1,N} \\ tr_{2,1} & tr_{2,2} & \ldots & tr_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ tr_{N,1} & tr_{N,2} & \ldots & tr_{N,N} \end{bmatrix}, \tag{2.5}$$

where N = number of states.

Definition 2.7. Emission probabilities: This is the probability that an observation k is being emitted from a particular state j. It is denoted as $e_j\left(k\right)$. k can also be represented as O_i. This implies that:

$$e_j\left(k\right) = P\left(x_t = k \mid n_t = j\right). \tag{2.6}$$

Generally, the emission probability matrix E is expressed as an $N \times M$ matrix shown in Equation 2.7:

$$E = \begin{bmatrix} e_{1,1} & e_{1,2} & \ldots & e_{1,M} \\ e_{2,1} & e_{2,2} & \ldots & e_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ e_{N,1} & e_{N,2} & \ldots & e_{N,M} \end{bmatrix}, \tag{2.7}$$

where M = number of observations.

Definition 2.8. Stationary Assumption: Contextually, a process is said to be stationary if the transition probabilities are independent of the time. For all t:

$$P\left[X_{t+1} = x_j \mid X_t = x_i\right] = p_{ij}. \tag{2.8}$$

2.3.2 Discrete HMM

The below example about a lady's aura is considered to illustrate how HMM works in discrete cases. If the lady's feeling has to be determined on a specific day, someone needs to observe the kind of activity she will mostly enjoy doing on that day. Explicitly, her feelings could either be"merry" denoted as M or "sad" denoted as S. We go by the assumption that the probability of her being merry on two successive days is 0.8 and the probability of her being sad on two successive days is 0.6. By analogy with the HMM, the transition probability matrix is as shown in Equation 2.9:

$$TR = \begin{array}{c} \\ M \\ S \end{array} \begin{array}{c} \begin{array}{cc} M & S \end{array} \\ \begin{bmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{bmatrix} \end{array}. \tag{2.9}$$

Given that she has 3 main activities which are dancing denoted as "D", strolling denoted as "S" and taking a nap denoted as "N". She does one of them in a day. We equally assume the probabilistic relationship between her feelings and her activities (observables) is comparable to an HMM such that the emission probability matrix is given in Equation 2.10:

$$E = \begin{array}{c} \\ M \\ S \end{array} \begin{array}{c} \begin{array}{ccc} D & S & N \end{array} \\ \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.1 & 0.4 & 0.5 \end{bmatrix} \end{array}. \tag{2.10}$$

For this example, the states are the lady's feelings being Merry and Sad, implies transitioning from one state to another and it is known as a Markov process. The Markov property is also observed since moving to the next state (feeling) depends only on the present state and the state transition probabilities as depicted in Equation 2.9. However, her feelings are not known until she carries out one of the 3 activities, this implies that her feelings are hidden. It means the states are hidden, thus a Hidden Markov Model. Parameters used to describe an HMM are [1]:

π = initial state distribution

TR = transition probability matrix

tr = transition probability

E = emission probability matrix

e = emission probability

$O = \{O_1, O_2, \ldots, O_T\}$ = sequence of observation or emission

L = length of the sequence of observation

M = number of observation symbols

$V = \{1, 2, \ldots, M\}$ = set of possible observations

N = number of states

$Q = \{q_1, q_2, \ldots, q_N\}$ = distinct states

Summarising the *feelings* example in the form of a general hidden Markov model is depicted in Fig. 2. Only the sequence of observation O_i is seen, while the sequence of state represented by X_i is hidden. The transition probability tr is the probability to move from a current state to the next state and e is the emission probability that an observation 0_i was emitted from a state X_i.

Given that the sequence of observations;

$$O = (D \ D \ S \ N \ N \ S)\,.$$

Let D, S and N be represented by $1, 2$ and 3 respectively. That is to say;

$$O = (1 \ 1 \ 2 \ 3 \ 3 \ 2)\,. \tag{2.11}$$

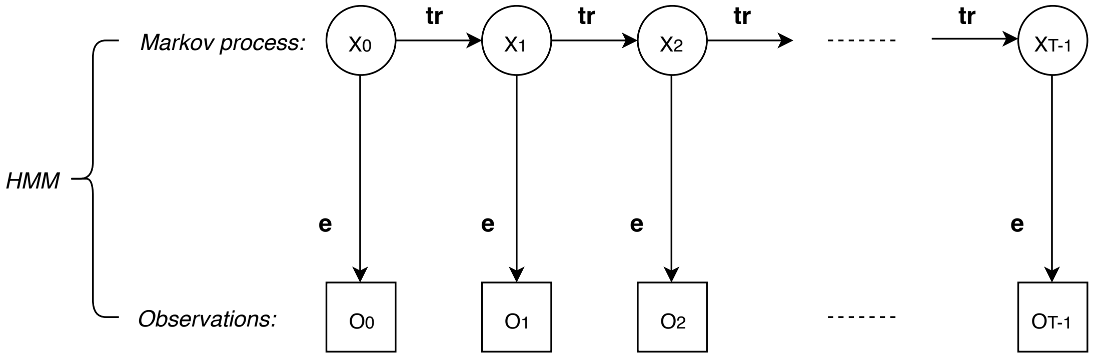

FIGURE 2.1: A General Hidden Markov Model [1]

For such an example, the model parameters can be assumed as stated below:

$$L = 6,$$

$$\pi = \{0.6, 0.4\},$$

$$TR = \begin{bmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{bmatrix},$$

$$E = \begin{bmatrix} 0.1 & 0.4 & 0.5 \\ 0.6 & 0.3 & 0.1 \end{bmatrix},$$

$$V = \{1, 2, 3\},$$

$$M = 3, \ and$$

$$N = 2.$$

Worth noting is that each row in TR and E is a probability distribution given that all elements in a row sum up to 1. TR is an $N \times N$ matrix which equals:

$$tr_{ij} = P\left(q_{t+1} = j \mid q_t = i\right), \tag{2.12}$$

and E is an $N \times M$ matrix which equals:

$$e_j\left(k\right) = P\left(x_t = k \mid q_t = j\right). \tag{2.13}$$

.

2.3.3 The three basic problems of HMM

HMM is actually defined and applied by solving three problems, namely: The evaluation problem, the decoding problem and the learning also known as the training problem.

I. **The Evaluation Problem**: This problem looks at the *parameter* space. It tends to answer the question of *What the probability of occurrence of a particular sequence of observation will be, given a model with assumed or known parameters.* Alternatively stated, it computes the likelihood that a known model denoted as λ will generate a particular sequence of L observations:

$$O = \{O_1, O_2, \ldots, O_L\} \textbf{ at L, time } t = T,$$

and the model λ has a triplet of parameters;

$$\lambda = (\pi, tr, e) .$$

Assuming a sequence of state is $X = \{x_1, x_2, \ldots, x_T\}$. To compute the likelihood $P(O \mid \lambda)$, we go by the definition of e as explained in Equation 2.6:

$$P(O \mid X, \lambda) = \prod_{t=1}^{T} P(O_t \mid x_t, \lambda) = e_{x_1}(O_1)\, e_{x_2}(O_2) \ldots e_{x_T}(O_T), \tag{2.14}$$

and by the definition of π and tr as stated in Equations 2.2 and 2.4 respectively:

$$P(X \mid \lambda) = \pi_{x_1} tr_{x_1,x_2} tr_{x_2,x_3} \ldots tr_{x_{T-1},x_T}. \tag{2.15}$$

Now, by the definition of conditional probability:

$$P(O, X \mid \lambda) = \frac{P(O \cap X \cap \lambda)}{P(\lambda)}. \tag{2.16}$$

Also,

$$\begin{aligned} P\left(O \mid X, \lambda\right) P\left(X \mid \lambda\right) &= \frac{P\left(O \cap X \cap \lambda\right)}{P\left(X \cap \lambda\right)} \frac{P\left(X \cap \lambda\right)}{P\left(\lambda\right)} \\ &= \frac{P\left(O \cap X \cap \lambda\right)}{P\left(\lambda\right)}. \end{aligned} \tag{2.17}$$

Thus,

$$P\left(O, X \mid \lambda\right) = P\left(O \mid X, \lambda\right) P\left(X \mid \lambda\right). \tag{2.18}$$

So, to obtain the likelihood $P\left(O \mid \lambda\right)$ of the observation sequence O, all the possible sequences of states are summed. This implies that:

$$\begin{aligned} P\left(O \mid \lambda\right) &= \sum_X P\left(O, X \mid \lambda\right) \\ &= \sum_X P\left(O \mid X, \lambda\right) P\left(X \mid \lambda\right) \\ &= \sum_X \pi_{x_1} e_{x_1}\left(O_1\right) tr_{x_1, x_2} e_{x_2}\left(O_2\right) tr_{x_2, x_3} \dots e_{x_L}\left(O_L\right) tr_{x_{L-1}, x_L}. \end{aligned} \tag{2.19}$$

Evaluating the obtained likelihood as illustrated in Equation 2.19 requires all the possible state paths to increase exponentially with the length of the observation sequence. This is not a suitable method. A better approach is using the *forward procedure*, also known as the *forward-algorithm* denoted Fw. We obtain the forward probability variable by computing the probability of the partial observation $O_1, O_2, \dots, O_t$ up and including the state at $x_t = q_i$ till time t. That is:

$$Fw_t\left(i\right) = P\left(O_1, O_2, \dots O_t, x_t = q_i \mid \lambda\right). \tag{2.20}$$

To calculate the forward algorithm recursively, 3 steps are considered:

i. Initialisation: for $i = 1, \dots, N$

$$Fw_1\left(i\right) = \pi_i e_i\left(O\right). \tag{2.21}$$

ii. Iteration: for $t = 1,\ldots,T-1$ and $j = 1,\ldots,N$. At T, the length of observation sequence is L. So,

$$Fw_{t+1}(j) = \left(\sum_{i=1}^{N} Fw_t(i)\, tr_{ij}\right) e_j(O_{t+1}) \,. \tag{2.22}$$

iii. Termination:

$$P(O \mid \lambda) = \sum_{i=1}^{N} Fw_T(i). \tag{2.23}$$

Thus, given the sequence of observation in Equation (2.11) as $O = (1\ 1\ 2\ 3\ 3\ 2)$, the likelihood that the model will generate O is computed based on the given parameters π, tr and e. The process involves obtaining the forward probability using Equation 2.22 as:

$$Fw = \begin{bmatrix} 1.000 & 0.4000 & 0.1750 & 0.5418 & 0.8894 & 0.9393 & 0.8218 \\ 0 & 0.6000 & 0.8250 & 0.4582 & 0.1106 & 0.0607 & 0.1782 \end{bmatrix}.$$

Thereafter, Fw values are evaluated using Equation 2.23 to obtain the likelihood as:

$$P_{lik1} = -6.7512$$

II. **The Decoding Problem**: This problem looks at the state space. It answers the question of *What the single most likely or optimal sequence of states to had generated a particular sequence of observations is.* In other terms, it finds the hidden part of the HMM. This fundamental problem of HMM is solved using the *Viterbi Algorithm VA*. The Viterbi algorithm is a dynamic programming algorithm as it updates or finds new solutions based on the previous solution for a given problem. Basically, it calculates all the possible paths (state sequences) and only outputs the path with the highest probability. Let VA be denoted as v. So, v is computed recursively such that:

At the initial stage, for $t = 1$ and $j = 1,\ldots,N$. It means:

$$v_1(j) = \pi_j e_j(O_1) \,. \tag{2.24}$$

Thus, for each element in the observation sequence and at each state, their corresponding state probabilities are computed. For each of them, the maximum probability is considered. These corresponding maximum probabilities then form the most probable state path. This implies that for : $t = 2, \ldots, T$ and $j = 1, \ldots, N$;

$$v_t\left(j\right) = \max\{v_{t-1}\left(i\right) tr_{ij} e_j\left(O_t\right)\}. \tag{2.25}$$

VA therefore finds the *best path* of the model. However, it may not inform about the *most probable state* for an observation O_i. The posterior probability denoted PP rather indicates the probability of a state x at time t, given a sequence of observations. The posterior probability approach is computed using both the *forward* and *backward algorithms.* The *backward algorithm* denoted as Bw accounts for the probability of partial sequence of observation, starting at the end of the sequence O_T and works recursively and reversely till O_{t+1} [20], thus similar to the forward algorithm. The backward probability variable is defined as:

$$Bw_t\left(i\right) = P\left(O_{t+1}, O_{t+2}, \ldots, O_T \mid x_t = n_i, \lambda\right). \tag{2.26}$$

So, Bw is calculated recursively by:

i. Initialisation: for $i = 1, \ldots, N$ implies:

$$Bw_T\left(i\right) = 1. \tag{2.27}$$

ii. Iteration: for $t = T$-1, $\ldots$, 1 and for $i = 1, \ldots, N$

$$Bw_t(i) = \sum_{j-1}^{N} tr_{ij} e_j(O_{t+1}) Bw_{t+1}(j). \tag{2.28}$$

iii. Termination: for $t = 1,\ldots,T$ and for $i = 1,\ldots,N$

$$PP_t\,(i) = P\,(x_t = n_i \mid O, \lambda) = \frac{Fw_t\,(i)\,Bw_t\,(i)}{P\,(O, \lambda)}. \tag{2.29}$$

As a continuation, the backward probability is obtained using Equation 2.28 as:

$$Bw = \begin{bmatrix} 1.0000 & 0.6709 & 1.2694 & 1.2428 & 1.0454 & 1.0064 & 1.0000 \\ 2.3291 & 1.2194 & 0.9429 & 0.7129 & 0.6346 & 0.9005 & 1.0000 \end{bmatrix}.$$

Then, the posterior probability PP is computed based on the forward and backward algorithms according to Equation 2.29 to obtain:

$$PP = \begin{bmatrix} 0.2684 & 0.2221 & \boxed{0.6733} & \boxed{0.9298} & \boxed{0.9453} & \boxed{0.8218} \\ \boxed{0.7316} & \boxed{0.7779} & 0.3267 & 0.0702 & 0.0547 & 0.1782 \end{bmatrix}.$$

Thus, the maximum probability value at each column of PP is chosen as:

$$PP_{max} = \begin{bmatrix} 0.7316 & 0.7779 & 0.6733 & 0.9298 & 0.9453 & 0.8218 \end{bmatrix},$$

such that the row index of PP_{max} corresponds to the states 1 and 2 which are the *Merry* and *Sad* moods of the lady, as explained in Section 2.3.2. This results to the best state sequence: $B_{ss} = [2\ 2\ 1\ 1\ 1\ 1]$. Hence B_{ss} indicates the best states to had generated the sequence of observation.

III. **The Learning Problem**: The learning problem is the most cumbersome amongst all the 3 problems. It is also known as the training problem and it faces an optimisation criterion. This problem answers the question of *What the model parameters are, given a trained sequence of observation.* Training seeks to find the model that best fits the given data. That is to say, the initial model parameter $\lambda = (\pi, tr, e)$ is continuously re-estimated to find the one that maximises the occurrence or likelihood of the given observation sequence. It computes λ such that $P\,(O \mid \lambda)$ is optimal.

To achieve this, several algorithms could be used: Viterbi training algorithm, maximum likelihood estimation and Baum-Welch (BW) algorithm. By default, BW algorithm is used and it applies the Forward and Backward algorithms as well. Analogous to Equation 2.29; knowing an observation sequence and the model parameters, the probability of being in a current state at a specific time, and being in a future state at another specific time is given as:

$$\begin{aligned} PP_t\left(i,j\right) &= P\left(x_t = n_i, x_{t+1} = n_j \mid O, \lambda\right) \\ &= \frac{Fw_t\left(i\right) tr_{ij} e_j\left(O_{t+1}\right) Bw_{t+1}\left(j\right)}{P\left(O \mid \lambda\right)}. \end{aligned} \tag{2.30}$$

Both Equation 2.29 and Equation 2.30 are used to re-estimate the triplet model parameters.

- For $i = 1,\ldots,N$ we assume:

$$\pi_i = P_1\left(i\right). \tag{2.31}$$

- For each sequence where $i = 1,\ldots,N$ and $j = 1,\ldots,N$; the probability of moving from state i to state j is obtained by the ratio:

$$tr_{ij} = \frac{Expected\ number\ of\ transitions\ from\ states\ n_i\ to\ n_j}{Expected\ number\ of\ times\ stopped\ at\ state\ n_i}.$$

  This is re-estimated and interpreted as:

$$tr_{ij} = \frac{\sum\limits_{t=1}^{T-1} PP_t\left(i,j\right)}{\sum_{t=1}^{T-1} PP_t\left(i\right)}. \tag{2.32}$$

- For $j = 1,\ldots,N$ and $k = 1,\ldots,M$; the probability of observing a symbol k or O_i is similarly obtained by the ratio:

$$e_j(k) = \frac{Expected\ number\ of\ times\ k\ is\ observed\ when\ the\ model\ is\ in\ states\ n_i}{Expected\ number\ of\ times\ being\ in\ state\ n_i}.$$

  This is shown as:

$$e_j(k) = \frac{\sum_{t\in\{1,\ldots,T\},O_t=k} PP_t\left(i,j\right)}{\sum_{t=1}^{T} PP_t(i)}. \tag{2.33}$$

The Baum Welch algorithm is computed iteratively till the optimisation criterion is attained. In other words, the training process ends when the maximum likelihood is reached. This could be when the change in log likelihood is amply less than some pre-established threshold or when the maximum number of iterations is reached. The iterative training process can be summarised in the following steps:

Given an observation sequence for which $j = 1,\ldots,N$;

i. Initialisation: The model 3 parameters $\lambda = (\pi, tr, e,)$ are rationally approximated but can also be assigned random values. Also, a threshold value is initially chosen.

ii. Recurrence: The forward and backward variables are computed, as well as the posterior state probabilities depicted in Equations 2.20, 2.26, 2.29 and 2.30 respectively. These results are used to estimate new values of π, tr and e such that the likelihood $P(O \mid \lambda)$ is maximised.

iii. Termination: The training process can either end when the change in the log likelihood is relatively small to the predefined threshold or when maximum iteration is exceeded.

Considering the example in Section 2.3.2, in the training stage, the Baum Welch algorithm utilises the same Forward and Backward algorithms obtained in the evaluation and decoding problems, alongside the posterior state probabilities in Equations 2.29 and 2.30 to re-estimate the initial tr and e at each iteration. The estimated tr is :

$$tr = \begin{bmatrix} 0.7490 & 0.2510 \\ 0.4986 & 0.5014 \end{bmatrix},$$

while the estimated e is :

$$e = \begin{bmatrix} 0.0000 & 0.4990 & 0.5010 \\ 0.9959 & 0.0041 & 0.0000 \end{bmatrix}.$$

Finally, from the result of the estimated tr and e, a new likelihood is computed as: $P_{lik2} = -6.4082$. Since this new likelihood is greater than the previous one, the estimated tr and e are returned into the BW algorithm and a stopping condition in is attained. The training process is expected to end. It implies attaining a convergence criterion. An end criterion is stopping the process when the difference in overall log likelihood is infinitesimal. Another stop criterion could be to end at the maximum number of iterations [22].

2.3.4 HMM-Gaussian model for one dimensional feature space

Considering the dataset to be short pulse calls of inshore Bryde's whales, this implies that the acoustic signal is continuous. As such, the process of analysing and detecting the wanted signal is slightly different from using a discrete dataset. In the case of processing continuous data, particular features are extracted from the signal and each state is represented by a Gaussian distribution with a mean and variance. So, a Gaussian distribution function with a scalar variable can also be used as:

$$P(O \mid X) = \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp^{-\frac{(o-\mu_x)^2}{2\sigma_x^2}}.$$

Basically, it has two parameters which can be estimated, given a sequence of observation $\{O_1, O_2, \ldots, O_N\}$:

$$\hat{\mu_x} = \frac{1}{N}\sum_{i=1}^{N} O_i,$$

$$\hat{\sigma_x^2} = \frac{1}{N}\sum_{i=1}^{N}(O_i - \mu_x)^2.$$

So, transition and emission probabilities of single multivariate Gaussian:

$$tr_{ij} = P(X = j \mid X = i),$$
$$e_j(O) = P(O \mid X = j) = \mathcal{N}\left(O; \mu_j, \sum_j\right),$$

where $\mu_j = mean$ and $\sum_j = covariance\ matrix.$

Feature extraction: For a continuous dataset it is essential to select good features as this will equally enhance the performance of our HMM. This implies that the feature extraction is an indispensable step in the process of pattern recognition and classification. It aims at reducing data dimensionality by extracting relevant information from the main dataset to form a sequence of wanted features [23]. The extracted features are the main characteristic information present within each frame of the signal or dataset. Often than not, data size is voluminous and consequently takes longer processing time. As such, there is a need to reduce the dimension of the original data and simultaneously maintain its wanted characteristics, to eventually improve on the computational time of the whole training and detection processes.

In line with this, and related to the basic concept of HMM; detection is based on a *sequence* of feature characteristics and not on a single frame (feature). Hence, the extracted desired features of each segment are combined and represented in a lower dimensional feature vector [24] as the sequence of observations. Dimensionality reduction can be done by feature selection or feature extraction.

Feature selection is the act of choosing relevant feature variables from a raw dataset D to form a subset of features D'. Feature selection does not transform the properties of the raw data, it rather removes unnecessary or redundant features and considers the relevant ones. Let $D = \{X_1, X_2, \ldots, X_N\}$, thus feature selection results to the subset $D' = \{X_{i_1}, X_{i_2}, \ldots, X_{i_M}\}$, with $M < N$.

$$\begin{array}{ccc} D & & D' \\ \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} & \xrightarrow{\textit{Feature Selection}} & \begin{bmatrix} X_{i_1} \\ X_{i_2} \\ \vdots \\ X_{i_M} \end{bmatrix}. \end{array} \tag{2.34}$$

Feature Extraction is the process of transforming the existing features into a lower dimensional space [25]. It rather creates a subset of brand new features B, by combinations of existing features F [26]. B is a compact feature vector that represents interesting parts and the most relevant information of F; where $F = \{f_1, f_2, \ldots, f_n\}$ and the extracted feature vector $B = \{b_1, b_2, \ldots, b_n\}$ with $B < F$. Each feature in B is a function of F. The main difference between the two methods is that feature selection filters useful dataset features while feature extraction produces entirely new smaller datasets with interesting features identified from the original dataset, making information more separable.

$$\begin{array}{ccc} F & & B \\ \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix} & \xrightarrow{Feature\ \ Extraction} & \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} \end{array} = f\left(\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}\right). \tag{2.35}$$

The feature extraction process could be done in time or frequency domains. It is dependent on the kind of features aimed at detecting. Some time domain features include average power, mean, zero-crossing rate (ZCR), while Mel Frequency Cepstral Coefficients (MFCC) and Linear Predictive Coefficients (LPC) are often the extracted features in the frequency domain [27], [28].

MFCC is a commonly used method for processing signals. It has been used to extract features in speech recognition, detection analysis of drone sound, identification of images, recognition of gestures, and in a few cetacean vocalization detection and classification algorithms. Because of its low computational complexity, MFCC is often used. Though it is simple and robust, it usually has a better performance compared to other feature extraction methods [29]. However, it is sensitive to noise due to its spectral characteristics. Features are created by using the Mel frequency scale to convert signals from the time domain into the frequency domain. In general, feature extraction using MFCC include seven successive steps: Pre-emphasis, Framing, Windowing, FFT, Mel-scale filter bank, Logarithm operation, and DCT [30], [31].

MFCCs are computed as discussed in [32]

$$\gamma_m = \sum_{i=1}^{n} X_i \cos\left(\frac{m\,(i-0.5)\,\pi}{n}\right), \tag{2.36}$$

where,

$m = 1,\ 2, \ldots, n$ represent the number of cepstral coefficients.

X_i = logarithmic energy of the i^{th} Mel spectrum band.

Linear Predictive Coefficient is another method of signal processing regularly used for linear prediction. It is used in speech coding, analysis of cetacean vocalisations, speech synthesis and recognition[33]. The basic concept of LPC technique is a linear combination of previous acoustic samples so as to predict a value for the actual sound signal $H(n)$ as defined by [33],[34] in Equations 2.37 and 2.38.

$$\hat{H}(n) = \sum_{k=1}^{m} a_k H(n-k), \tag{2.37}$$

where $H(n)$ is the actual sample and a_k are m^{th} order of linear prediction coefficients obtained by summing up the squared differences between the current and linearly predicted samples, such that:

$$d\left[n\right] = H(n) - \hat{H}(n), \tag{2.38}$$

where $\hat{H}(n)$ is the predicted value, and $d\left[n\right]$ is the prediction error. The feature extraction methods employed in this study are in the time domain. The three time domain features are average power, mean and zero-crossing rate which are described as follows:

A. **Average Power** (P_{avg})

The average power of a signal is the sum of the absolute squares of its time-domain samples divided by the signal length. In other terms, given a frame with N sampling points which is equivalent to the frame length of a snippet, then the value of these sampling points are squared, added and divided by N as illustrated in Equation 2.39. The P_{avg} indicates the loudness of the frame and it is represented as:

$$P_{avg} = \frac{1}{N}\sum_{n=1}^{N}(X_n)^2. \tag{2.39}$$

The main reason for choosing P_{avg} as a time domain feature is because it provides a basis for separating voiced from unvoiced components of speech signal. Also, it is a good measuring tool to differentiate detectable and silent sounds with a high signal-to-noise ratio [28].

B. **Mean** (μ)

The mean of an acoustic signal is similar to the P_{avg} but for the fact that it sums up all the sampling points N per signal frame such that :

$$\mu = \frac{1}{N}\sum_{n=1}^{N} X_n. \tag{2.40}$$

In general, the mean is used to reduce the background noise or remove the DC component which could most probably alter the signal's waveform. So, the computed mean per frame of a sound snippet serves as the time domain feature vector.

C. **Zero-crossing rate** ($\mathcal{ZCR}$)

$\mathcal{ZCR}$ is described as the number of times a sound signal changes its sign from positive to negative or otherwise, in a precise frame [35]. This feature can be represented as the amount of time-domain zero-crossing in a processing frame as expressed in [27]. The ZCR simply measures the frequency content of a sound signal without necessarily working in the frequency domain which is computed as [28]:

$$\mathcal{ZCR} = \frac{1}{2(K-1)}\sum_{n=1}^{K-1}|sgn[x_{n+1}] - sgn[x_n]|, \tag{2.41}$$

where $sgn[x_n]$ is a signum function such that:

$$sgn[x_n] = \begin{cases} 1, & x_n \geq 0 \\ -1, & x_n < 0. \end{cases} \tag{2.42}$$

$\mathcal{ZCR}$ is considered as a time domain feature extraction since it measures a wide variance and amplitude range for the ZCR curve [28] which is the case for the Bryde's whale vocalisation as it falls within an amplitude band.

Hence, these time-domain features are each obtained from frames as illustrated in Equation 3.5, for every whale and noise snippets. Given a frame $f_1 = \{r_1, r_2, \ldots, r_a\}$, the average power feature is computed for f_1 as:

$$P_{avg} = \frac{1}{a}(r_1^2 + r_2^2 + \cdots + r_a^2). \tag{2.43}$$

Similarly, the mean feature for f_1 is derived as:

$$\mu = \frac{r_1 + r_2 + \cdots + r_a}{a}. \tag{2.44}$$

Also, the zero-crossing rate feature for f_1 is calculated as:

$$ZCR = \frac{1}{2(a-1)}\Big(|r_2 - r_1| + |r_3 - r_2| + \cdots + |r_a - r_{a-1}|\Big). \tag{2.45}$$

Once the wanted features have been extracted, the dataset is selected such that the whale and noise extracted features are respectively chosen based on a certain specified amount. They then represent the sequence of observation which is used to train the desired HMM.

Training: As mentioned in Section 2.3.3, to train an HMM model implies estimating the best model parameters that will maximise the occurrence of a given sequences of extracted features. In other words, training an HMM model is faced with an optimisation problem, and Expectation Maximisation (EM) is employed in this case, since it iteratively attempts to estimate the Maximum Likelihood (ML) of a model's parameters. The model parameters to be estimated are:

i. Initial state probabilities π

ii. Transition probabilities tr

iii. Emission probabilities e

However, for a Gaussian model, e is represented by the mean μ and the standard deviation σ which is the case in this study. Thus, each emission distribution e is represented as:

$$e = \{\boldsymbol{\mu_1}, \boldsymbol{\mu_2}, \ldots, \boldsymbol{\mu_J}, \boldsymbol{\sigma_1}, \boldsymbol{\sigma_2}, \ldots \boldsymbol{\sigma_J}\}, \tag{2.46}$$

where $j = 1, \ldots, J$.

In order to achieve the training process, the Expectation (E) and Maximisation (M) steps are carried out.

Expectation step: The E - step assigns a probability to every feature value, depending on the current distribution state parameters. It actually finds the expectation of the log-likelihood given the sequence of features or observations O and the current parameter estimates λ^{old}.
Applying this step to an HMM:

$$\begin{aligned}
Q(\lambda, \lambda^{\text{old}}) &= \sum_{x} p(X \mid O, \lambda^{\text{old}}) \ln p(O, X \mid \lambda) \\
&= \sum_{x} p(X \mid O, \lambda^{\text{old}})[\ln p(x_1) + \sum_{n=2}^{N} \ln p(x_n \mid x_{n-1}) + \sum_{n=1}^{N} \ln p(o_n \mid x_n)] \\
&= \sum_{x_1} p(x_1 \mid O, \lambda^{\text{old}}) \ln p(x_1) + \sum_{n=2}^{N} \sum_{x_{n-1}x_n} p(x_{n-1}x_n \mid O, \lambda^{\text{old}}) \ln p(x_n \mid x_{n-1}) + \\
&\sum_{n=1}^{N} \sum_{x_n} p(x_n \mid O, \lambda^{\text{old}}) \ln p(o_n \mid x_n) \\
&= \sum_{j=1}^{J} \gamma(x_{nj}) \ln \pi_j + \sum_{n=2}^{N} \sum_{i=1}^{J} \sum_{j=1}^{J} \xi(z_{(n-1),i} x_{n,j}) \ln tr_{ij} + \sum_{n=1}^{N} \sum_{j=1}^{J} \gamma(x_{nj}) \ln p(o_n \mid x_{nj}, e),
\end{aligned} \tag{2.47}$$

where

λ^{old} = A constant which represents the current estimates of parameters.

λ = Parameter to be optimised while trying to maximise the log-likelihood.

O = Sequence of observation or feature vector.

X = Sequence of states. $p(X \mid O, \lambda^{\text{old}})$ = Conditional probability distribution of O.

$\gamma(x_{nj}) = p(x_{nj} \mid O, \lambda^{\text{old}})$ represents the probability of being in state $n = j$.

$\xi(x_{n-1,i} x_{n,j}) = p(x_{n-1,i} x_{n,j} \mid O, \lambda^{\text{old}})$ represents the transitional probability from state i to state j.

The posterior state probability given a sequence of features is obtained as:

$$\begin{aligned}
\gamma(x_n) &= p(x_n \mid O) \\
&= \frac{p(O \mid x_n)p(x_n)}{p(O)} \\
&= \frac{o_{n-1}o_n)p(o_{n+1}, o_{n+2}, \ldots, o_N \mid x_n)p(x_n)}{p(o_1, o_2, \ldots, o_n)p(o_{n+1}, o_{n+2}, \ldots, o_N \mid o_1, o_2, \ldots, o_n)} \\
&= \frac{p(o_1, o_2, \ldots, o_n, x_n)p(o_{n+1}, o_{n+2}, \ldots, o_N \mid x_n)}{p(o_1, o_2, \ldots, o_n)p(o_{n+1}, o_{n+2}, \ldots, o_N \mid o_1, o_2, \ldots, o_n)} \\
&= \hat{\alpha}(x_n)\hat{\beta}(x_n).
\end{aligned} \tag{2.48}$$

We observe that;

$$\hat{\alpha}(x_n) = \frac{p(o_1, o_2, \ldots, o_n, x_n)}{p(o_1, o_2, \ldots, o_n)}, \tag{2.49}$$

$$\hat{\beta}(x_n) = \frac{p(o_{n+1}, o_{n+2}, \ldots, o_N \mid x_n)}{p(o_{n+1}, o_{n+2}, \ldots, o_N \mid o_1, o_2, \ldots, o_n)}, \tag{2.50}$$

where Equations (2.49) and (2.50) are the *Forward* and *Backward algorithm* terms respectively.

Decomposing the transitional probability from a previous state to a current state and using Bayes' rule:

$$\begin{aligned}
\xi(x_{n-1}x_n) &= p(x_{n-1}x_n \mid O) \\
&= \frac{p(O \mid x_{n-1}x_n)p(x_{n-1}x_n)}{p(O)} \\
&= \frac{p(o_1, o_2, \ldots, o_{n-1} \mid x_n)p(o_n \mid x_n)p(o_{n+1}, o_{n+2}, \ldots, o_N \mid x_n)p(x_{n-1}x_n)}{p(O)} \\
&= \frac{p(o_1, o_2, \ldots, o_{n-1}, x_n)/p(x_n)p(0_n \mid x - n)p(o_{n+1}, o_{n+2}, \ldots, o_N \mid x_n)p(x_{n-1}x_n)}{p(O)} \\
&= \hat{\alpha}(x_{n-1})p(o_n \mid x_n)p(x_n \mid x_{n-1})\hat{\beta}(x_n).
\end{aligned} \tag{2.51}$$

Maximisation step: The M - step actually updates the expectation for the distributional state parameters based on the new assigned data. So, it calculates new model parameters by maximizing the expected log-likelihood previously obtained during the E - step.

In order to maximise π, [20] the Lagrange optimisation techniques like the Lagrange function is used, and $\sum_{i=1}^{J} \pi_i = 1$ is considered as the constraint such that:

$$L_1(\boldsymbol{\pi}, \Lambda) = \sum_{j=1}^{J} \gamma(z_n j) \ln \pi_j + \Lambda \left(\textstyle\sum_{i=1}^{J} \pi_i - 1 \right). \tag{2.52}$$

To simplify Equation (2.52), the gradient of the Lagrange function is set to 0, thereby letting the partial derivatives of all the components of the function equals 0;

$$\frac{\partial L_1(\boldsymbol{\pi}, \Lambda)}{\partial \pi_j} = 0,\ j = 1, \ldots, J \Rightarrow \pi_j = \frac{\gamma_1 j}{\sum_{i=1}^{j} \gamma_{1i}},\ j = 1, \ldots, J \tag{2.53}$$

Similarly, to maximise tr:

$$\sum_{l=1}^{J} tr_{il} = 1,\ l = 1, \ldots, K \tag{2.54}$$

is taken as the constraint of the Lagrange function. This implies that;

$$\begin{aligned} L_2(\boldsymbol{tr}, \Lambda_1, \ldots, \Lambda_j) = & \sum_{n=2}^{N} \sum_{i=1}^{J} \sum_{j=1}^{J} \xi(x_{n-1,i} x_{n,j}) \ln tr_i j \\ & + \Lambda_1 \left(\textstyle\sum_{l=1}^{J} tr_1 l - 1 \right) + \cdots + \Lambda_J \left(\textstyle\sum_{l=1}^{J} tr_J l - 1 \right). \end{aligned} \tag{2.55}$$

Again, setting the Lagrange function to 0;

$$\frac{\partial L_2(tr, \Lambda_1, \ldots, \Lambda_j)}{\partial tr_{ij}} = 0\ j = 1, \ldots, J \Rightarrow tr_{ij} = \frac{\sum_{n=2}^{N} \xi(x_{n-1,i} x_{n,j})}{\sum_{l=1}^{J} \sum_{n=2}^{N} \xi(x_{n-1,i} x_{n,l})}\ i, j = 1, \ldots, J \tag{2.56}$$

The last model parameter to be optimised is the emission probability matrix e, with emphasis in a Gaussian distribution. Thus, the probability density function:

$$p(\boldsymbol{O}_n \mid X_{nj}, e) = N(\boldsymbol{O}_n \mid \boldsymbol{\mu}_j \boldsymbol{\sigma}_j), \tag{2.57}$$

where $e = \{\boldsymbol{\mu}_j, \boldsymbol{\sigma}_j\}$, $j = 1, \ldots, J$

With respect to the Lagrange function:

$$\begin{aligned} L_3(e, e^{\text{old}}) &= \sum_{n=1}^{N} \sum_{j=1}^{J} \gamma(x_n j) \ln p(o_n \mid x_{nj}, e) \\ &= \sum_{n=1}^{N} \sum_{j=1}^{J} \gamma(x_n j) \left[-\frac{D}{2} \ln(2\pi) - \frac{1}{2} \ln |\sigma_j| - \frac{1}{2} (o_n - \mu_j)^{\mathrm{T}} \sigma_k^{-1} (o_n - \mu_j) \right]. \end{aligned} \tag{2.58}$$

Therefore, to estimate μ_j and σ_j, the derivative of Equation (2.58) with respect to $e = \{\boldsymbol{\mu}_j, \boldsymbol{\sigma}_j\}$, $k = 1, \ldots, K$ is equated to 0. As such;

$$\frac{\partial L_3(e, e^{\text{old}})}{\partial \boldsymbol{\mu}_j} = 0 \; j = 1, \ldots, J \Rightarrow \boldsymbol{\mu}_j = \frac{\sum_{n=1}^{N} \gamma_{nj} o_n}{\sum_{n=1}^{N} \gamma_{nj}}, \; j = 1, \ldots, J \tag{2.59}$$

$$\frac{\partial L_3(e, e^{\text{old}})}{\partial \sigma_j} = 0 \; j = 1, \ldots, J \Rightarrow \boldsymbol{\sigma}_j = \frac{\sum_{n=1}^{N} \gamma_{nj} (\boldsymbol{O}_n - \boldsymbol{\mu}_j)(\boldsymbol{O}_n - \boldsymbol{\mu}_j)^{\mathrm{T}}}{\sum_{n=1}^{N} \gamma_{nj}} \quad j = 1, \ldots, J \tag{2.60}$$

The training process is expected to end. It implies attaining a convergence criterion. An end criterion is stopping the process when the difference in overall log likelihood is infinitesimal. Another stop criterion could be to end at the maximum number of iterations [22].

This process is better explained in a summarised four-stepped order using the simple *numerical example* below. Two Gaussian HMM models are considered as the Whale sound HMM and the Noise HMM, denoted by M_w and M_n respectively with initial parameters given as $M_w = \{\pi_w, tr_w, e_w\}$ and $M_n = \{\pi_n, tr_n, e_n\}$. Each model comprises of 2 states and the observation sequence Obs_{seq} is given as:

$$\begin{aligned} Obs_{seq} =& \{0.9892, 0.2430, 1.0069, 0.5957, 1.9102, 0.6344, 0.2903, 1.6480, \\ & 0.8577, 0.5957, 1.1201, 1.0367, 0.8315, 0.5957, 1.5561, 0.1951, 0.8819, \\ & 1.2898, 0.9416, 0.4276, 1.4719, 1.6355, 0.1468, 1.5563, 0.9988, 1.2943\}. \end{aligned} \tag{2.61}$$

i. Initialisation of parameters; let the initial parameters be given as:

$$\pi_w = \{0.3263, 0.6737\},\ \pi_n = \{0.6142, 0.3858\}.$$

$$tr_w = \begin{bmatrix} 0.4964 & 0.5036 \\ 0.4875 & 0.5125 \end{bmatrix},\ tr_n = \begin{bmatrix} 0.0441 & 0.9559 \\ 0.5504 & 0.4496 \end{bmatrix}.$$

Also, the emission probabilities distribution are obtained as:

$$e_w = \begin{cases} \mu_w = \{1.7968, 0.7071\} \\ \sigma_w = \{0.0351, 0.0930\} \end{cases},\ e_n = \begin{cases} \mu_n = \{0.4213, 1.6277\} \\ \sigma_n = \{0.0329, 0.0491\}. \end{cases}$$

ii. E - step; using eq. (2.48) the posterior probabilities (γ) are computed for both the Whale and Noise snippets (ws and ns respectively) as:

$$\gamma_{ws_1} = \begin{bmatrix} 0.0000 & 1.0000 \\ 0.9972 & 0.0029 \end{bmatrix},\ \gamma_{ws_2} = \begin{bmatrix} 0.0000 & 1.0000 \\ 0.0043 & 0.9957 \\ 0.0007 & 0.9993 \end{bmatrix},$$

$$\gamma_{ws_3} = \begin{bmatrix} 0.0000 & 1.0000 \\ 1.0000 & 0.0000 \end{bmatrix},\ \gamma_{ws_4} = \begin{bmatrix} 0.0000 & 1.0000 \\ 0.9883 & 0.0117 \end{bmatrix}.$$

and:

$$\gamma_{ns_1} = \begin{bmatrix} 1.0000 & 0.0000 \\ 0.0000 & 1.0000 \\ 1.0000 & 0.0000 \\ 0.0000 & 1.0000 \end{bmatrix},\ \gamma_{ns_2} = \begin{bmatrix} 1.0000 & 0.0000 \\ 0.0000 & 1.0000 \\ 1.0000 & 0.0000 \\ 0.0000 & 1.0000 \end{bmatrix},$$

$$\gamma_{ns_3} = \begin{bmatrix} 0.9965 & 0.0035 \\ 0.0000 & 1.0000 \\ 0.0000 & 0.0000 \end{bmatrix}, \gamma_{ns_4} = \begin{bmatrix} 1.0000 & 0.0000 \\ 0.0000 & 1.0000 \\ 1.0000 & 0.0000 \\ 0.0000 & 1.0000 \end{bmatrix}.$$

It is important to note that, from the annotation and segmentation process 4 snippets are each obtained for the Whale snippet and Noise snippet.

More so, using Equation (2.51) the decomposed transitional probabilities (ξ) are calculated for the Whale snippets as:

$$\xi_{ws_1} = \begin{bmatrix} 0.0000 & 0.0000 \\ 0.9972 & 0.0029 \end{bmatrix}, \xi_{ws_2} = \begin{bmatrix} 0.0008 & 1.0000 \\ 0.0050 & 1.9909 \end{bmatrix},$$

$$\xi_{ws_3} = \begin{bmatrix} 0.0008 & 0.0000 \\ 1.0000 & 0.0000 \end{bmatrix}, \xi_{ws_4} = \begin{bmatrix} 0.0008 & 0.0000 \\ 0.9883 & 0.0117 \end{bmatrix}.$$

Likewise, ξ is computed for the Noise snippets as:

$$\xi_{ns_1} = \begin{bmatrix} 0.0000 & 2.0000 \\ 1.0000 & 0.0000 \end{bmatrix}, \xi_{ns_2} = \begin{bmatrix} 0.0000 & 2.0000 \\ 1.0000 & 0.0000 \end{bmatrix},$$

$$\xi_{ns_3} = \begin{bmatrix} 0.0000 & 0.9965 \\ 0.0000 & 1.0035 \end{bmatrix}, \xi_{ns_4} = \begin{bmatrix} 0.0000 & 2.0000 \\ 1.0000 & 0.0000 \end{bmatrix}.$$

iii. M - step; using Equation (2.53), the initial state probabilities of the Whale and Noise models are optimised as:

$$\hat{\pi}_w = \{0.0000, 1.0000\}, \ \hat{\pi}_n = \{0.9991, 0.0009\}.$$

Moreover, the initial transition probabilities are maximised using Equation (2.56) as:

$$\hat{tr}_w = \begin{bmatrix} 0.0008 & 0.9992 \\ 0.5986 & 0.4014 \end{bmatrix}, \hat{tr}_n = \begin{bmatrix} 0.0000 & 1.0000 \\ 0.7493 & 0.2507 \end{bmatrix},$$

while, the initial emission probabilities are equally maximised using Equations (2.59) and (2.60) as:

$$\hat{e}_w = \begin{cases} \hat{\mu}_w = \{1.7956, 0.7047\} \\ \hat{\sigma}_w = \{0.0354, 0.0911\}, \end{cases} \quad \hat{e}_n = \begin{cases} \hat{\mu}_n = \{0.4218, 1.6287\} \\ \hat{\sigma}_n = \{0.0331, 0.0484\}. \end{cases} \tag{2.62}$$

iv. If convergence is reached, then the process stops. Else, it goes back to the E - step.

After training and prior to the detection phase, the Whale sound model (M_w) and Noise model (M_n) are combined into a single model. This is attained by using their respective estimated model parameters as $\{\hat{\pi}_w, \hat{tr}_w, \hat{e}_w\}$ and $\{\hat{\pi}_n, \hat{tr}_n, \hat{e}_n\}$ as:

(a.) The initial state probability distributions are concatenated as $\hat{\pi}_{wn} = [\hat{\pi}_n, \hat{\pi}_w]$, such that $\hat{\pi}_{wn} = \{0.9991, 0.0009, 0.0000, 1.0000\}$.

(b.) The transition probability matrices tr_w and tr_n are represented as a block diagonal matrix B:

$$B = \left[\begin{array}{c|c} tr_w & 0 \\ \hline 0 & tr_n \end{array}\right],$$

where $\hat{tr}_w$ is an $N \times N$ matrix with states $\{w_1, w_2, \dots w_N\}$, and $\hat{tr}_n$ is an $N \times N$ matrix with states $\{w'_1, w'_2, \dots w'_N\}$. In this example, $N = 2$ states per model. Therefore, the combined $\hat{tr}_{wn}$ is:

$$\hat{tr}_{wn} = \begin{array}{c} \\ w'_1 \\ w'_2 \\ w_1 \\ w_2 \end{array} \begin{array}{c} \begin{array}{cccc} w'_1 & w'_2 & w_1 & w_2 \end{array} \\ \begin{bmatrix} 0.0000 & 1.0000 & 0 & 0 \\ 0.7493 & 0.2507 & 0 & 0 \\ 0 & 0 & 0.0008 & 0.9992 \\ 0 & 0 & 0.5986 & 0.4014 \end{bmatrix} \end{array}. \tag{2.63}$$

(c.) Parameters associated with the emission distribution of both models ($\hat{e}_w$ and $\hat{e}_n$) are equally combined. Since the emission distribution is represented by μ

and σ, these inner estimated parameters for both models are concatenated as $\hat{\mu}_{wn} = [\hat{\mu}_n, \hat{\mu}_w]$ and $\hat{\sigma}_{wn} = [\hat{\sigma}_n, \hat{\sigma}_w]$. That is:

$\hat{\mu}_{wn} = \{0.4218, 1.6287, 1.7956, 0.7047\}$,

$\hat{\sigma}_{wn} = \{0.0331, 0.0484, 0.0354, 0.0911\}$.

Decoding: This section aims at finding the *corresponding* sequence of states, say $X = \{x_1, x_2, \ldots, x_N\}$, given a sequence of features and a model. Of importance, is considering the commonly known optimal criterion of obtaining the single best path (sequence of states). This is like optimising $p(X, O \mid \lambda)$ [20]. The Viterbi algorithm best attempts to resolve the decoding process as:

$$\begin{aligned} X_{optimised} &= \underbrace{\max}_{z} p(X \mid O) = \underbrace{\max}_{z} p(X, O) \\ &= \underbrace{\max}_{z} p(O_1) \prod_{n=2}^{N} p(O_n \mid X_{n-1}) \prod_{n=1}^{N} p(O_n \mid X_n). \end{aligned} \tag{2.64}$$

So, the Viterbi algorithm is simplified in the below formula as it discards, step wisely, paths with low probability while storing the previous step. It means:

$$V_{1j} = p(o_1 \mid x_{1j}) \times p(x_{1j}), \tag{2.65}$$

and for $n = 2, \ldots, N$

$$V_{nj} = \max(V_{n-1,i} \times p(x_{nj} \mid x_{n-1,i}) \times p(o_n \mid x_n i)), \tag{2.66}$$

where the state path $(n-1) = i$ and path $(n) = \text{argmax } V_n i$.

In other words, it calculates the probability of each extracted feature with respect to all the existing number of states in the model(s). Only the highest probability is retained step-likely. It then allocates the state number to which each extracted feature belongs and a state path is later obtained as a combination of the retained maximum probabilities.

In this example, the existing number of states of the model M_{wn} are $\{w_1, w_2, w'_1, w'_2\}$ and given a another sequence of observation as the test data (often represented as the extracted features), then the output of the decoding phase is a state sequence associated with the combination and permutation of all the existing four states as depicted in Table 1.

Detection: The performance of the automated HMM model is determined during the detection phase. The measure of a successful (positive) or unsuccessful (negative) detection of the Bryde's whale call often produces four outcomes. For a better understanding of these outcomes, the Bryde's whale call is said to be the correct signal while the noise sound is considered as the wrong signal. Thus:

i. True Positive, $\mathbb{T}_p$: It measures the number of correctly identified sounds. Clearly explained as the number of times the output of the automated model matches the manually annotated sound.

ii. False Positive, $\mathbb{F}_p$: It determines the number of incorrectly identified sounds. This implies the number of times the automated model predicts a wrong signal as the supposed call to be detected. A low false positive rate indicates the dependability of the model.

iii. False Negative, $\mathbb{F}_n$: It measures the number of incorrectly rejected sounds. This is the number of times the model misses the targetted (manually annotated) sound.

iv. True Negative, $\mathbb{T}_n$: It evaluates the number of correctly rejected sounds. That is, the number of times the automated model predicts the wrong signal just as the wrongly annotated signal.

As emphasis, the Whale sound model constitutes of states $M_w = \{w_1, w_2, \dots w_N\}$ whilst Noise model constitutes of states $M_n = \{w'_1, w'_2, \dots w'_N\}$. With this regard, four inherent assumptions considered during the detection of the Bryde's whale sound from the M_{wn} are:

- If the sequence of states corresponding to the sequence of observation (a segment of the Whale extracted feature samples) comprises of *more than or equal to* 75% of $w_1, w_2, \dots w_N$ *states*, then the sound is said to be correctly detected or identified. It means $\mathbb{T}_p$.

- If the sequence of states corresponding to the sequence of observation of the wrong signal (a segment of the Noise extracted feature samples) constitutes of *more than or equal to* 75% of $w_1, w_2, \dots w_N$ *states*, then the sound is said to be incorrectly detected. So, it is $\mathbb{F}_p$.

- If the states sequence corresponding to the sequence of extracted feature samples of the Whale signal includes *more than or equal to* 75% of $w'_1, w'_2, \dots w'_N$ *states*, then the sound is to be said to be rejected incorrectly. Thus $\mathbb{F}_n$.

- If the states sequence corresponding to the sequence of extracted feature samples of the Noise (wrong) signal consists of *more than or equal to* 75% of $w'_1, w'_2, \dots w'_N$ *states*, then the sound is said to be rejected correctly. That is, $\mathbb{T}_n$.

So, in line with the numerical example, the detection outcome shown in Table 1 has the following number of occurrences:

$$\begin{aligned} \mathbb{T}_p &= 1, \\ \mathbb{F}_p &= 1, \\ \mathbb{F}_n &= 3, \ \ \textit{and} \\ \mathbb{T}_n &= 3. \end{aligned}$$

The aforementioned outcomes are used to qualify the performance of the detector being the hidden Markov model for our case study. The performance parameters of the automated HMM evaluated in comparison to the extracted features are *Sensitivity*, *Accuracy*, and *False Positive Rate*:

- Sensitivity Tpr; It is also called True Positive Rate and measures how often the prediction truly is, considering that the wanted sound is manually known. It is computed by:

$$Tpr = \frac{\mathbb{T}_p}{\mathbb{F}_n + \mathbb{T}_p}. \tag{2.67}$$

Sound type	**Test data**	**Probability**				**Path**	**Outcome**
		w'_1	w'_2	w_1	w_2		
NS	0.4510	$\boxed{0.7722}$	-13.7436	-24.7705	-0.0745	1	T_n
	1.6355	-21.4567	$\boxed{0.5951}$	0.3897	-4.4750	2	
	0.3024	$\boxed{0.5695}$	-17.5900	-30.7228	-0.6094	1	
	1.7330	-25.1737	0.4831	$\boxed{0.6961}$	-5.5231	3	
WS	0.2599	$\boxed{0.3892}$	-18.7724	-32.5373	-0.8067	1	F_n
	1.7186	-24.6087	$\boxed{0.6678}$	0.5119	-5.3623	2	
NS	0.3302	$\boxed{0.6581}$	-16.8360	-29.5624	-0.4910	1	T_n
	1.5952	-20.0047	$\boxed{0.5840}$	0.1847	-4.0724	2	
	0.5024	$\boxed{0.6869}$	-12.5177	-22.8541	0.0543	1	
	1.3809	-13.1032	$\boxed{-0.0393}$	-1.6762	-2.2298	2	
WS	0.7503	-0.8445	-7.3802	-14.6709	$\boxed{0.2674}$	4	T_p
	1.1088	-6.3400	-2.1990	-5.9074	$\boxed{-0.6169}$	4	
	1.0364	-4.9180	-3.0310	-7.3844	$\boxed{-0.3248}$	4	
NS	0.8200	-1.6084	-6.1658	-12.6847	$\boxed{0.2060}$	4	F_p
	1.3167	-11.3079	-2.4851	$\boxed{-0.4104}$	-1.7765	3	
	0.5586	$\boxed{0.5024}$	-11.2414	-20.8466	0.1618	1	
WS	1.5490	-18.4005	$\boxed{0.5300}$	-0.1066	-3.6328	2	F_n
	0.3644	$\boxed{0.7352}$	-15.9281	-28.1611	-0.3566	1	
	1.6976	-23.7917	0.5464	$\boxed{0.6159}$	-5.1307	3	
NS	0.2671	$\boxed{0.4234}$	-18.5706	-32.2281	-0.7721	1	T_n
	1.7401	-25.4561	$\boxed{0.7079}$	0.4672	-5.6036	2	
	0.2812	$\boxed{0.4865}$	-18.1741	-31.6200	-0.7052	1	
	1.6698	-22.7303	$\boxed{0.5781}$	0.5279	-4.8315	2	
WS	0.4048	$\boxed{0.7806}$	-14.8889	-26.5519	-0.2147	1	F_n
	1.4976	-16.6911	$\boxed{0.4181}$	-0.5015	-3.1711	2	
	0.6191	$\boxed{0.2387}$	-9.9410	-18.7858	0.1972	1	

Table 1: Decoding and detection outputs

- Accuracy *Acc*; On an overall basis, it measures how often the automated model is correct or "true". It is calculated as:

$$Acc = \frac{\mathbb{T}_p + \mathbb{T}_n}{\mathbb{F}_p + \mathbb{F}_n + \mathbb{T}_p + \mathbb{T}_n}. \tag{2.68}$$

- False Positive rate *Fpr*; This parameter estimates how often the automatic model predicts a wrong sound as the known sound. Its computation employs the formula:

$$Fpr = \frac{\mathbb{F}_p}{\mathbb{T}_n + \mathbb{F}_p}. \tag{2.69}$$

Such a detection rate should not be enhanced. The empirical output should tend towards zero, thus indicating the model's reliability.

Therefore, from Equations (2.67) to (2.69):

$$\begin{aligned} Tpr &= \frac{1}{3+1} = 0.25 \\ Acc &= \frac{1+3}{1+3+1+3} = 0.5 \\ Fpr &= \frac{1}{3+1} = 0.25 \end{aligned}$$

To conclude, the numeric example presents a model able to detect a Bryde's whale call with a sensitivity of 25%, 50% accuracy and false positive of 0.25.

Chapter 3

Research Methodology

3.1 Introduction

This chapter addresses how the short pulse calls of inshore Bryde's whales dataset was collected and the approaches used to implement the automated detector which is the hidden Markov model. The methods carried out involve segmenting and framing the dataset, extracting time domain features from the dataset. Also, other subsections of this chapter explain the training, decoding and detection processes of the hidden Markov model.

3.2 Data collection

The raw sound data used in this work was collected by our research group within a couple of days in January 2019. The dataset consists of short pulse calls of inshore Bryde's whales together with dissimilar sounds like those of other marine mammals and noise of various forms. This dataset is as a result of four different recordings with a total of 44hr26mins which is provided to analyse the inshore Bryde's whales short pulse calls. During the four instances when the sound datasets were collected, a single Brydes whale was visually seen thrice while three Brydes whales were also visually observed in one of the instances. In terms of the geographical location, the data was recorded close to Gordon's bay harbour situated at $34°8'57.5''S$ $18°51'26.7''E$ and in False bay situated at $34°12'38.9''S$ $18°38'27.3''E$, South Africa, as seen in Figure 3.1. Normal protocols were stringently followed as stipulated by the South African Department of Environmental Affairs. Amongst others were: the recommended minimum distance on sighting of whales was respected, and it was identified every moment it was observed. Dipped hydrophones were used for the series of recordings done. In the course of carrying out this task, a hydrophone was joined to a Zoom H1N recorder. Specifically, Aquarian Audio H2A-XLR Hydrophone with sensitivity -180 dB re: $1V$ at a frequency interval of 10 Hz - 100 kHz and the recorder working at 24 bit resolution at sampling frequency $F_s = 96000$ Hz . The raw data was stored as a .wav file, that is to say in an uncompressed and lossless format, so as to keep its original properties. The hydrophone was immersed to a depth of approximately 7 m from a sailboat of about 8 m.

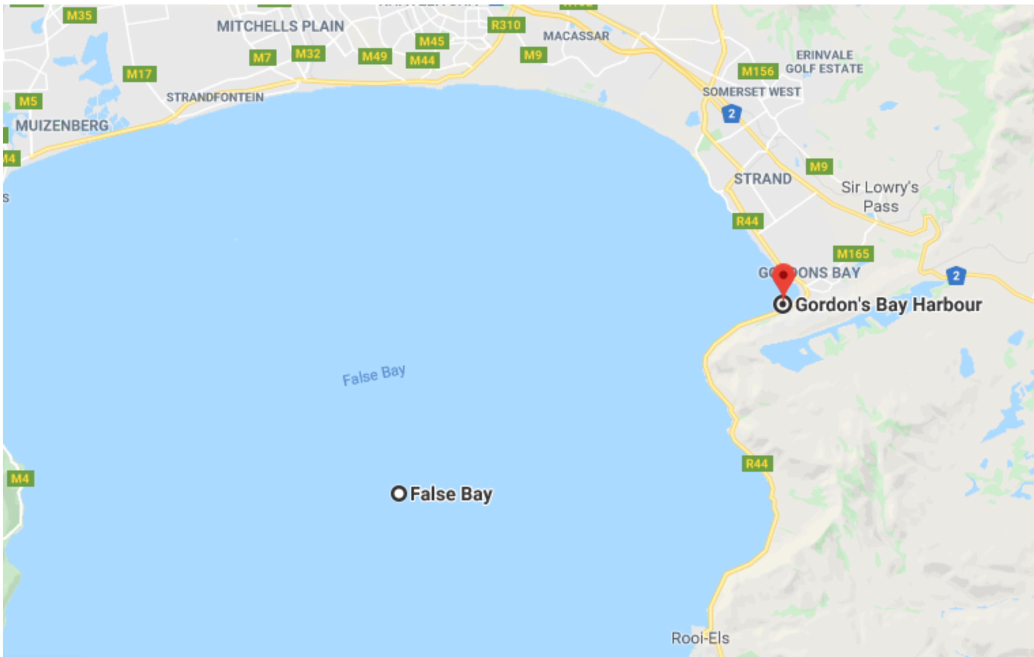

FIGURE 3.1: Source of the raw dataset recordings in South Africa (Image obtained from Google map)

3.3 Data preparation

Data pre-processing into appropriate formats is a significant step to carry out. Following the recordings and collection of the acoustic dataset, the raw data was processed before any further use. Of interest, is maintaining wanted signal quantities and removing undesired ones. This is achieved by using the established MATLAB Butterworth bandpass filter given its smooth frequency response in the passband. Passband frequency was chosen between 90 Hz and 46000 Hz, reason being that fundamental call frequencies of Bryde's whale are generally greater than 90 Hz [9], and the upper band frequency being half the sampling rate of the data. Moreover, to remove any average voltage (DC components) from the data acquired from the Butterworth bandpass filter, the mean is subtracted from it. As a result, we obtain a pre-processed data with less attenuation since a $3rd$ order Butterworth bandpass filter was used. Thus, the data is segmented for further analysis.

3.4 Data annotation and segmentation

The pre-processed sound dataset is analysed by visual and auditory inspection on a software, *Sonic Visualiser* - version 3.2.1 as seen in Figure 3.2. The analytic step involves annotating the pre-processed dataset into two main acoustic categories known as segments. The segments include the Bryde's whale vocalization labelled as *Whale snippet* - *WS*, and any sound (produced either by anthropogenic activities or other marine mammals) other than the Bryde's whale call is labelled *Noise snippet* - *NS*. In other terms, during the visual and auditory inspection on Sonic Visualiser, several sounds such as the short pulse calls of Brydes whales, sounds arising from the breaking waves on the ocean, ambient noise, other unidentified marine mammals vocalizations and underwater noise from some ships were detected. These auditorily detected sounds other than the Brydes whales short pulse call were then grouped into a single category and annotated as Noise snippet while the Brydes whales short pulse call was categorized and annotated as Whale snippet. The main reason for this was that we want to detect Bryde's whales using a simple detector. Thus, the

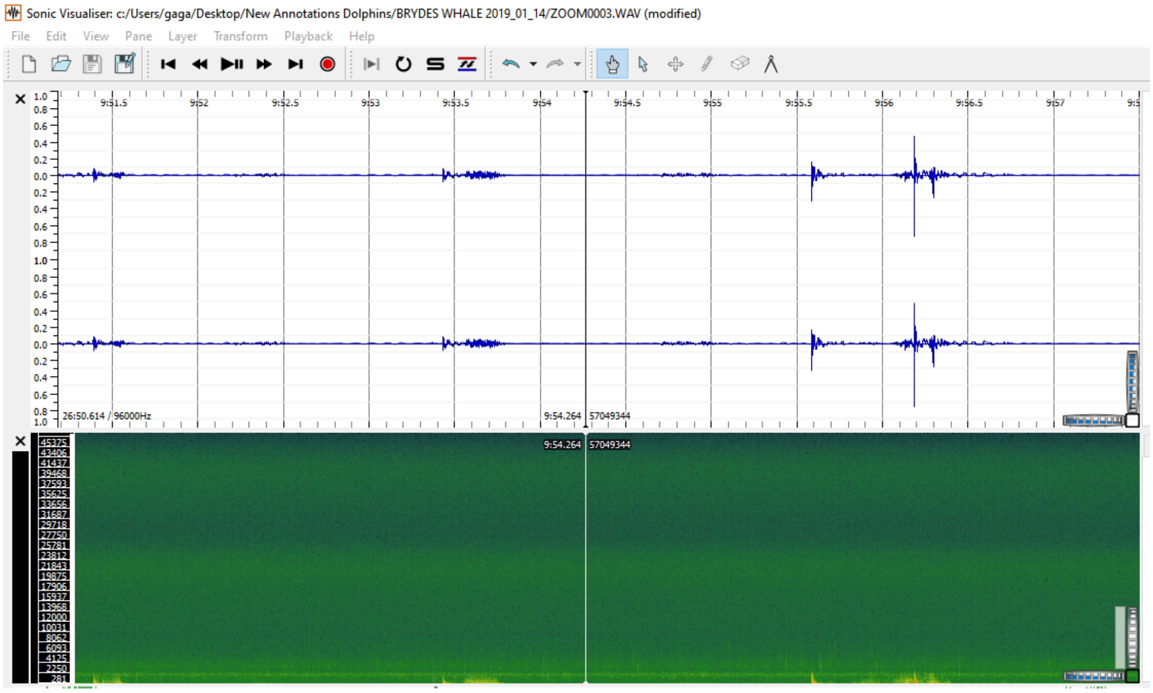
Sonic Visualiser: c:/Users/gaga/Desktop/New Annotations Dolphins/BRYDES WHALE 2019_01_14/ZOOM0003.WAV (modified)
File Edit View Pane Layer Transform Playback Help
9:51.5
9:52
9:52.5
9:53
9:53.5
9:54
9:54.5
9:55
9:55.5
9:56
9:56.5
9:57
26:50.614 / 96000Hz
9:54.264 57049344

FIGURE 3.2: Sound data annotation on Sonic Visualiser

pre-processed data is visually observed to be:

$$D = [\underbrace{d_1, d_2, d_3}_{WS_1}, \underbrace{d_4, d_5}_{NS_1}, \underbrace{d_6, d_7, d_8, d_9}_{WS_2}, \underbrace{d_{10}, d_{11}, d_{12}}_{NS_2}, \ldots, \underbrace{d_{n-5}, d_{n-4}, d_{n-3}}_{WS_p}, \underbrace{d_{n-2}, d_{n-1}, d_n}_{NS_q}],$$

where d represent the sampling points of the pre-processed data.

p is the total number of whale snippets obtained from annotating D.

q is the total number of noise snippets obtained from annotating D.

Also, from D, it is seen that each respective snippet has a different length or call duration. Hence, all the annotated and categorised snippets are stored in two corresponding sets WS_i and NS_j such that:

$$WS_i = \begin{Bmatrix} WS_1 \\ WS_2 \\ \vdots \\ WS_p \end{Bmatrix} \equiv \begin{Bmatrix} [d_1, d_2, d_3] \\ [d_6, d_7, d_8, d_9] \\ \vdots \\ [d_{n-5}, d_{n-4}, d_{n-3}] \end{Bmatrix}, \tag{3.1}$$

where $i = 1, 2, \dots p$, and

$$NS_i = \begin{Bmatrix} NS_1 \\ NS_2 \\ \vdots \\ NS_q \end{Bmatrix} \equiv \begin{Bmatrix} [d_4, d_5] \\ [d_{10}, d_{11}, d_{12}] \\ \vdots \\ [d_{n-2}, d_{n-1}, d_n] \end{Bmatrix}, \tag{3.2}$$

where $i = 1, 2, \dots q$.

3.5 Frame extraction

Both whale and noise snippets are eventually split into smaller portions known as frames. The frame extraction process is based on the frame size or frame length (F_l) and the number of sampling points in each snippet. The frame length is derived by multiplying the sampling frequency F_s by the frame duration F_d as:

$$F_l = F_s \times F_d, \tag{3.3}$$

where F_l equally represents the number of sampling points in a frame. In this piece of work, the frame duration is chosen in terms of a short, medium and long duration assumed as 1 ms, 5 ms, and 10ms respectively.

Therefore, the number of frames x in a snippet is obtained as:

$$x = \frac{r}{F_l}, \tag{3.4}$$

where r is number of samplings points in a snippet. Hence, every whale and noise snippet stored in the corresponding sets WS_i and NS_i is split into frames by dividing each of them with the frame length. For a better understanding of the frame extraction process, suppose an arbitrary whale snippet to be:

$$WS_1 = [\underbrace{r_1, r_2, r_3, \dots, r_a}_{f_1}, \underbrace{r_{a+1}, \dots, r_{2a}}_{f_2}, \underbrace{r_{2a+1}, \dots, r_{3a}}_{f_3}, \dots, \underbrace{r_{(x-1)a+1}, \dots, r_{xa}}_{f_x}], \tag{3.5}$$

where $f_1, f_2, \dots, f_x$ represent the frames, x is number of frames and $a = F_l$.

Worthy to note is that, in cases where r is not evenly split by F_l, then r is padded with zeros to maintain consistency while evenly splitting r by F_l. Likewise, any other noise snippet is split into frames as illustrated in Equation 3.5. For the purpose of this piece of work, frame extraction has been implemented based on three different frame durations: 1, ms 5, ms and 10 ms, one at a time. Once the frames have been extracted, and depending on the kind of feature aimed at extracting and detecting, the feature extraction step is performed on every frame.

3.6 Feature extraction

In this thesis, three time domain features have been used to extract wanted characteristics from the segmented dataset. The extracted features are average power, mean and zero-crossing rate (ZCR). The process of extracting each feature is performed at three different frame lengths as a result of the three assumed frame durations 1 ms, 5 ms and 10 ms. This process is implemented on all the whale and noise snippets. Consequently, two respective sets of extracted features are obtained for both the whale and noise snippets denoted as W_f and N_f respectively.

Let p_i, m_i z_i be the respective P_{avg}, mean and ZCR extracted feature samples corresponding to the i^th frame f_i, where $i = 1, 2, \ldots, x$. Thus, the set of extracted features for all the whale snippets is represented as:

$$W_f = \begin{array}{c} \\ f_1 \\ f_2 \\ \vdots \\ f_x \end{array} \!\!\begin{array}{c} \begin{array}{ccc} P_{avg} & Mean & ZCR \end{array} \\ \begin{bmatrix} p_1 & m_1 & z_1 \\ p_2 & m_2 & z_2 \\ \vdots & \vdots & \vdots \\ p_x & m_x & z_x \end{bmatrix} \end{array}, \tag{3.6}$$

while the set of extracted features for all the noise snippets is represented as:

$$N_f = \begin{array}{c c} & \begin{array}{ccc} P_{avg} & Mean & ZCR \end{array} \\ \begin{array}{c} f_1 \\ f_2 \\ \vdots \\ f_x \end{array} & \begin{bmatrix} p_1 & m_1 & z_1 \\ p_2 & m_2 & z_2 \\ \vdots & \vdots & \vdots \\ p_x & m_x & z_x \end{bmatrix} \end{array}. \tag{3.7}$$

Different feature values are obtained as the frame durations are varied from 1 ms, 5 ms and 10 ms. It is important to note that these extracted features are also known as the observation sequence, which have been used to train and evaluate the model.

3.7 Data selection

The extracted features have been selected such that 70% is used to train the corresponding whale and noise models and the remaining percentage (30%) is later used as another sequence of observation or test data, to evaluate the model.

3.8 Training

The process of training a hidden Markov model is explained in detail in Section 2.3.4. 70% of the data represented by the extracted features are used in training the model. The HMM also begins the training process with other key model parameters such as the initial state probability distribution, the transition and emission probabilities. Also, the number of states used in the course of training the models are varied from 3 states, 5 states and 10 states. So, rather than using just the sample of Obs_{seq} as shown by the numeric example in Section 2.3.4, 70% the three time domain extracted features has been used as the Obs_{seq} to train the models. Therefore the performance of this HMM is influenced by varying three things; frame durations, extracted features and the number of states used in the models. Hence, this process aims at re-estimating the model parameters given an observation sequence.

3.9 Decoding

The decoding step finds the best state sequence given an observation sequence and a model. It is done such that each extracted feature sample from each snippet is evaluated on a state distribution likelihood. This means, the probability that a feature sample or an observation belongs to one of the existing states that comprises that model. As indicated in Section 2.3.4, it is achieved using the Viterbi algorithm and at the end of the process, the maximum probabilities of the state distribution likelihood as computed in Equation 2.64, is considered for the corresponding frame. The states thereby form a path known as the best state sequence to have generated the observation sequence given the model as well.

3.10 Detection

The detection phase actually verifies how performant the developed HMM is, with respect to the different parameters variations. In other words, it establishes a relationship between the annotation and decoding processes. This means that it checks the occurrence of a sound previously annotated as either the Bryde's whale call or as a Noise sound, to have been detected by the HMM exactly as identified or otherwise. The process tells us how accurate, sensitive and reliable our model is, and which time domain extracted feature exhibits the best performance. The numeric example in Section 2.3.4, clearly elaborates how the performances have been measured using Equations (2.67) to (2.69). While Chapter 4 will analyse the results obtained from this multiple variations.

Chapter 4

Results and Discussion

Numerical analysis of the automated detector are presented in this chapter. They examine three major parameters, namely: the frame duration, extracted feature and number of states varied several times. The research methodology in Chapter 3 are carried out to produce the below tables as will be discussed in the subsections of Chapter 4. Of importance, the time domain features (average power, mean and zero crossing rate) are computed depending on the *frame duration* of a snippet.

4.1 Frame duration - 1 ms

No of States (N)	**Sensitivity**			**Accuracy**			**False Positive rate**		
	P_{avg}	$Mean$	ZCR	P_{avg}	$Mean$	ZCR	P_{avg}	$Mean$	ZCR
3	97.69	95.43	90.81	94.91	92.26	87.58	0.14	0.23	1.83
5	98.98	96.54	91.89	96.27	93.31	88.72	0.15	0.17	1.59
10	99.56	96.99	95.71	96.34	94.69	93.55	0.10	0.13	1.42

TABLE 1: Performance comparison of Feature Extraction at frame duration = 1ms

Table 1 shows the performances of the average power (P_{avg}), mean and zero-crossing rate (ZCR) extracted features using different states with a frame duration of 1 ms. It is observed in Table 1 that the three extracted features have the highest percentage of sensitivity when 10 states are used compared to when 3 and 5 states are used. Likewise, the extracted features are mostly accurate when 10 states are used by a model in comparison to 3 and 5 states. Moreover, the false positive rate performances of P_{avg}, mean and ZCR indicate that using 10 states produce the best performance as compared to the model with 3 and 5 states. Consequently, the performances of each extracted feature are analysed using 10 number of states.

In this case, the P_{avg} is the most sensitive (99.56%) feature compared to 96.99% and 95.71% sensitivity for mean and ZCR accordingly. Furthermore, Table 1 shows that P_{avg} feature is most accurate since it yields 96.34% as compared with the mean and ZCR that produce 94.69% and 93.55% respectively. In addition, the P_{avg} continues to show the best performance as it exhibits the lowest false positive rate (FPR) of 0.10 in comparison to the mean with 0.13 FPR and ZCR with 1.42 FPR. By implication, ZCR will produce the least performance, followed by the mean. But, the P_{avg} has the best performance with regard to Table 1, and could be considered as a time domain feature for an HMM.

4.2 Frame duration - 5 ms

No of States (N)	**Sensitivity**			**Accuracy**			**False Positive rate**		
	P_{avg}	*Mean*	*ZCR*	P_{avg}	*Mean*	*ZCR*	P_{avg}	*Mean*	*ZCR*
3	97.61	94.81	82.42	93.72	91.59	79.25	0.12	0.17	1.87
5	98.93	95.83	86.89	94.96	91.94	82.41	0.14	0.19	1.83
10	99.45	96.13	91.36	95.71	92.28	87.47	0.11	0.16	1.59

Table 2: Performance comparison of Feature Extraction at frame duration = 5ms

Table 2 indicates the performance comparison of each feature extraction with an increased frame duration from 1 ms to 5 ms. This implies that fewer number of frames are used, thus reducing the computational time to train and detect the dataset. Similar to the frame duration of 1 ms, the features exhibit the best performances in terms of sensitivity, accuracy and false positive rate when 10 states are used in a model as compared to using 3 and 5 states. As a result, we analyse each extracted feature based on the sensitivity, accuracy and FPR measures when $N = 10$. The percentage of correctly identified sounds being the sensitivity, presents a P_{avg} feature of 3.32% and 8.09% more than the mean and ZCR respectively. Moreso, the P_{avg} exhibits 3.43% and 8.24% accuracy gain over the mean and ZCR respectively. In addition, the P_{avg} shows a low FPR of 0.11 in comparison to 0.16 and 1.59 FPR produced by the mean and ZCR respectively. This result indicates that extracting the P_{avg} as a feature enhances the performance of the model compared to the mean and ZCR features.

However, a trade off occurs between the computational time and all the three performance parameters (sensitivity, accuracy and FPR) considered for the model. For sensitivity; P_{avg}, mean and ZCR yield a performance loss of 0.11%, 0.86%, and 4.35% respectively in Table 2. A similar performance loss of 0.63%, 2.41%, and 6.08% is obtained for P_{avg}, mean and ZCR respectively, with regard to accuracy in Table 2. Also, considering the FPR, the extracted features in Table 2 produce 0.01%, 0.03%, and 0.17% performance less than Table 1. The performance loss is due to the use of less amount of data (as feature vectors) during the training and detection phases.

4.3 Frame duration - 10 ms

No of States (N)	**Sensitivity**			**Accuracy**			**False Positive rate**		
	P_{avg}	*Mean*	*ZCR*	P_{avg}	*Mean*	*ZCR*	P_{avg}	*Mean*	*ZCR*
3	97.45	93.84	81.77	93.14	88.78	77.25	0.20	0.33	1.98
5	98.88	94.79	86.65	93.82	90.27	81.94	0.16	0.21	1.91
10	99.25	95.82	90.40	94.73	91.51	85.37	0.14	0.20	1.70

Table 3: Performance comparison of Feature Extraction at frame duration = 10ms

Table 3 shows a further increase in the frame length. This is to verify the performance of the extracted features based on an increase in the frame duration from 5 ms to 10 ms. Similar to Tables 1 and 2, the model exhibits the best performance when 10 states are used. Here, P_{avg} yields a 99.25% sensitivity measure as compared with mean and ZCR that produce 95.82% and 90.40% respectively. In addition, the P_{avg} has an accuracy performance of 94.73% compared to 91.51% of the mean and 85.37% of the ZCR. Therewithal, the P_{avg} exhibits the least FPR of 0.14 as compared to 0.20 and 1.70 of the mean and ZCR respectively. Hence, the P_{avg} exhibits an overall performance gain compared to the mean and ZCR, making it the most competent time domain feature for the automated model.

4.4 Different frame durations for N = 10

Frame duration (ms)	**Sensitivity**			**Accuracy**			**False Positive rate**		
	P_{avg}	*Mean*	*ZCR*	P_{avg}	*Mean*	*ZCR*	P_{avg}	*Mean*	*ZCR*
1	99.56	96.99	96.71	96.34	94.69	93.55	0.10	0.13	1.42
5	99.45	96.13	91.36	95.71	92.28	87.47	0.11	0.16	1.59
10	99.25	95.82	90.40	94.73	91.51	85.37	0.14	0.20	1.70

TABLE 4: Performance comparison of Feature Extraction at different frame durations for N = 10

Table 4 illustrates that increasing the frame duration from 5 ms to 10 ms results in a reduced sensitivity of 0.2%, 0.31%, 0.96% for P_{avg}, mean, ZCR respectively. Comparing the sensitivity measure of Tables 1 and 3 shows an ample difference of 0.31%, 1.17%, 5.31%, for P_{avg}, mean, ZCR respectively. Likewise, the accuracy performance of each extracted feature (P_{avg}, mean, ZCR) is reduced by 0.98%, 0.77%, 2.1% and 1.61%, 3.18%, 8.18%, when the frame length increases from 5 ms to 10 ms and 1 to 10 ms respectively. Furthermore, the FPR measure shows a decrease in P_{avg}, mean, ZCR of 0.03, 0.04, 0.11 and 0.04, 0.07, 0.28 as a result of an increase in the frame duration from 5 ms to 10 ms and 1 ms to 10 ms respectively.

A more in-depth analysis of the performance parameters is shown in Figures 4.1 to 4.3. In Figure 4.1, the sensitivity performance of the time domain features are compared based on the frame duration. As the number of states used increases, P_{avg} is slightly more sensitive compared to the mean and ZCR at frame duration of 1 ms. In the same way, P_{avg} presents the highest sensitivity performance as compared with the mean and ZCR at frame durations of 5 ms and 10 ms. Moreover, the P_{avg} offers the general best performance in all the three frame durations. Hence, the P_{avg} is the most sensitive time domain feature for the automated model.

In addition, Figure 4.2 shows the accuracy performance comparison of the time domain features according to the frame durations. The figure shows that P_{avg} offers the highest accuracy performance compared to the mean and ZCR at 1 ms frame duration. Also, the accuracy of P_{avg} at frame duration of 1 ms approaches the ZCR at 1 ms when the model constitutes 10 states. Even more, P_{avg} at 10 ms frame duration exhibits an accuracy which is greater than than the accuracy of the mean and ZCR at the frame duration of 5 ms. This implies that P_{avg} yields a more accurate model as compared with mean and ZCR, despite the increase in the frame duration of the model.

Furthermore, the time domain features at a frame duration of 10 ms demonstrate the least FPR in Figure 4.3 since the features with the lowest FPR is an indication of a model with best performance. The P_{avg} at frame duration of 1 ms has approximately the same performance in comparison to the mean and ZCR when 5 and 10 states are used in the model. Nevertheless, the P_{avg} yields the least FPR at 5 and 10 when the model constitutes any of the three number of states.

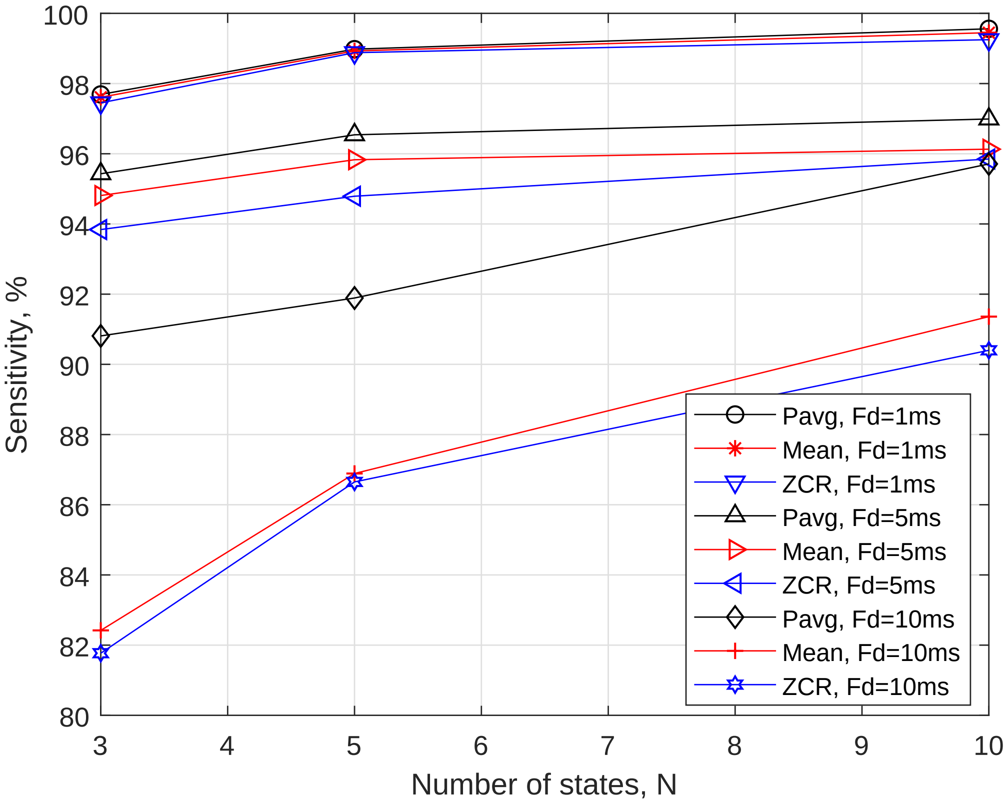

FIGURE 4.1: Sensitivity Performance Comparison of time domain extracted features

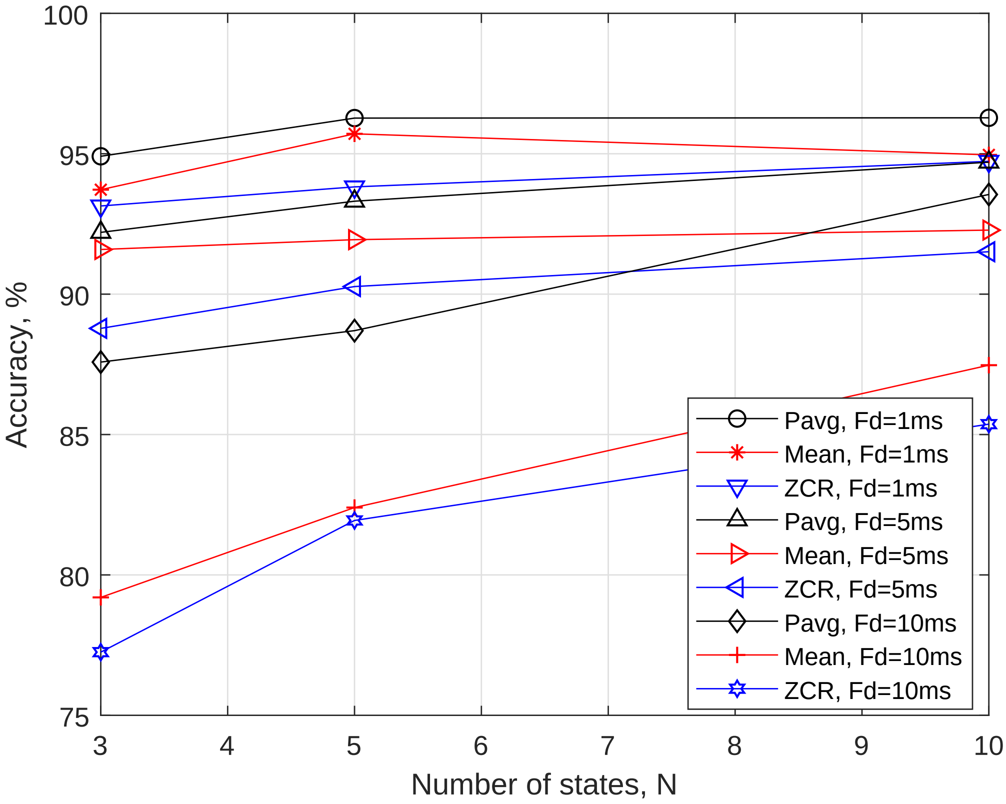

FIGURE 4.2: Accuracy Performance Comparison of time domain extracted features

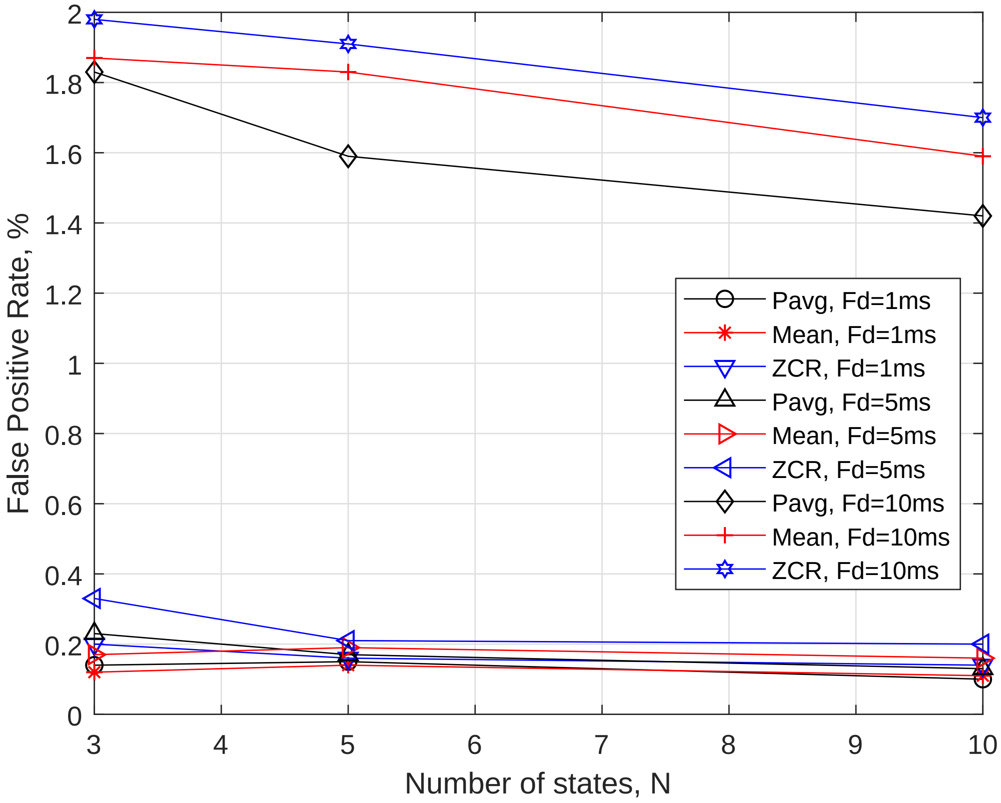

FIGURE 4.3: FPR Performance Comparison of time domain extracted features

Chapter 5

Conclusion and Future work

5.1 Conclusion

The research work has developed an automated acoustic detector for Brydes whales vocalisations, based on time domain features and the hidden Markov model technique. The raw sound dataset of the Brydes whales calls was recorded within $44hr26mins$ and collected from Gordons bay harbour to False bay. This dataset has been pre-processed, annotated and segmented into two main acoustic categories, the *Whale* snippets (WS) and *Noise* snippets (NS). Each of the acoustic categories have equally been split into frames to critically analyse them. The analysis done on the frames is the feature extraction process which aims at reducing the dimensionality of the sound dataset. This is achieved by computing wanted time domain feature on each frame of the corresponding snippets.

The three time domain feature extractions performed on these frames are the average power (P_{avg}), mean and the zero-crossing rate (ZRC). They have been considered because P_{avg} provides a basis for separating voiced from unvoiced components of speech signal. While the mean removes DC components from the raw dataset and the ZCR measures a wide variance and amplitude range of a signal. So, the (P_{avg}), mean and ZRC are the features extracted from both the whale and noise snippets. Moreover, each feature has been extracted at three different frame lengths. This is because we have considered three dissimilar frame durations of the calls at 1 ms, 5 ms, and 10 ms. Eventually, these extracted features have been used to train the hidden Markov models.

Furthermore, the number of states used in a model during the training process has also been varied as either 3, 5 or 10 states. On a general observation, the model exhibits best performances when 10 states are used. Also, with regard to the frame duration of the snippets, the model yields an overall best performance when a short frame duration of 1 ms is considered, in comparison to 5 ms and 10 ms. Moreover, the model has offered the best performance while using the P_{avg} as the extracted feature, as compared with the mean and (ZRC) extracted feature. However, the mean presents a similar low false positive rate as the P_{avg} when the model has 5 and 10 states, at 1 ms frame duration. Also, with respect to these three time domain features, the model has shown to be sensitive and dependable as it yields a low false positive rate in the overall performance. From the analysis and discussion of the results obtained in this study, it can be inferred that average power is the best time domain feature used to detect the short pulse call of inshore Bryde's whales based on hidden Markov model technique.

5.2 Future work

More work on the detection of Bryde's whale vocalisations based on the time domain features and HMM can be performed using other time domain features and the results should be compared with the results in this thesis. Also, a further work can be done by increasing the number of states used in the model for comparative purpose. More so, future studies can address the frame duration assumptions made in this study by considering much longer frame lengths, and compared the results with those in this work.

Chapter 6

Conclusion

This proposal is on developing an efficient iterative soft-decision decoding algorithm based on the parity-check equations, which necessitates finding a new class of moderately long cyclic codes. The results of this research aims at making a significant contribution to the problem of having an efficient iterative decoding algorithm for non-MDS codes and also the problem of obtaining asymptotically good moderately long cyclic codes with reduced encoding and decoding complexity. The literature review has shown that there is no known study that categorically deals with the problem of transforming the parity-check matrix for non-MDS codes. Hence a major contribution is to provide a practical solution to the transformation process and improve on the iterative steps of the PTA decoder by presenting a punctured parity-check transformation algorithm.

The methods used in literature on constructing a class of q-ary good infinite length cyclic codes provide potential for a novel solution on extending from existing double error-correcting codes to a general multiple error-correcting code. We have identified the problem with the existing PTA process, while work is ongoing to develop a technique of modifying the PTA for both MDS and non-MDS codes.

We have clearly mentioned the project scopes on decoding and code construction, method of testing and the possibility of achieving the set objectives within available resources and time-line. The schedule is for the PhD research to be successfully completed by September 2019 in order to graduate in December 2019.

Bibliography

[1] M. Stamp. “A revealing introduction to hidden Markov models.”, 2004.

[2] W. M. Hartmann and J. V. Candy. “Acoustic Signal Processing.” In *Springer Handbook of Acoustics*, pp. 519–563. Springer New York, 2014.

[3] E. Browning, R. Gibb, P. Glover-Kapfer, and K. E. Jones. “Passive acoustic monitoring in ecology and conservation.” Tech. rep., Oct. 2017.

[4] S. N. S. and R. Deshmukh. “Speech Recognition System – A Review.” *IOSR Journal of Computer Engineering*, vol. 18, no. 04, pp. 01–09, Apr. 2016.

[5] K. Singh. “Speech Recognition: A Review of Literature.” *In ternational Journal of Engineering Trends and Technology (IJETT)*, vol. 37, no. 6, Jul. 2016.

[6] R. C. R.L. Putland, L. Ranjard and C. Radford. “A hidden Markov model approach to indicate Brydes whale acoustics.” *Ecological Indicators*, vol. 84, pp. 479–487, Jan. 2018.

[7] *The Society for Marine Mammalogy*, Aug. 2019. URL https://www.marinemammalscience.org/species-information/list-marine-mammal-species-subspecies.

[8] J. Anderson. *Anatomical and zoological researches: comprising an account of the zoological results of the two expeditions to western Yunnan in 1868 and 1875; and a monograph of the two cetacean genera, Platanista*. B. Quaritch,, 1878.

[9] E. M. Oieson, J. Barlow, J. Gordon, S. Rankin, and J. A. Hildebrand. “Low frequency calls of Bryde’s whales.” *Marine Mammal Science*, vol. 19, no. 2, pp. 407–419, Apr. 2003.

[10] P. L. Edds, D. K. Odell, and B. R. Tershy. “Vocalisations of a captive juvenile and free-ranging adult-calf pairs Bryde’s whales, Balaenoptera edeni.” *Marine Mammal Science*, vol. 9, no. 3, pp. 269–284, Jul. 1993.

[11] S. Salvador and P. Chan. “Toward Accurate Dynamic Time Warping in Linear Time and Space.” *Intelligent Data Analysis*, vol. 11, no. 5, pp. 561–580, Oct. 2007.

[12] J. C. Brown, A. Hodgins-Davis, and P. J. O. Miller. “Classification of vocalizations of killer whales using dynamic time warping.” *The Journal of the Acoustical Society of America*, vol. 119, no. 3, pp. EL34–EL40, Mar. 2006.

[13] J. C. Brown and P. J. O. Miller. “Automatic classification of killer whale vocalizations using dynamic time warping.” *The Journal of the Acoustical Society of America*, vol. 122, no. 2, pp. 1201–1207, Aug. 2007.

[14] M. Bahoura and Y. Simard. “Blue whale calls classification using short-time Fourier and wavelet packet transforms and artificial neural network.” *Digital Signal Processing*, vol. 20, no. 4, pp. 1256–1263, Jul. 2010.

[15] J. R. Potter, D. K. Mellinger, and C. W. Clark. “Marine mammal call discrimination using artificial neural networks.” *The Journal of the Acoustical Society of America*, vol. 96, no. 3, pp. 1255–1262, Sep. 1994.

[16] S. Mazhar, T. Ura, and R. Bahl. “Vocalization based Individual Classification of Humpback Whales using Support Vector Machine.” In *OCEANS 2007*. IEEE, Sep. 2007.

[17] S. Jarvis, N. DiMarzio, R. Morrissey, and D. Morretti. “Automated Classification of Beaked Whales and Other Small Odontocetes in the Tongue of the Ocean, Bahamas.” In *OCEANS 2006*. IEEE, Sep. 2006.

[18] D. K. Mellinger, S. W. Martin, R. P. Morrissey, L. Thomas, and J. J. Yosco. “A method for detecting whistles, moans, and other frequency contour sounds.” *The Journal of the Acoustical Society of America*, vol. 129, no. 6, pp. 4055–4061, Jun. 2011.

[19] Q. Wang, L. Wang, and X. Xu. "Automatic detection and classification of marine mammal tonal calls." In *2017 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*. IEEE, Oct. 2017.

[20] L. Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition." *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[21] J. C. Brown and P. Smaragdis. "Hidden Markov and Gaussian mixture models for automatic call classification." *The Journal of the Acoustical Society of America*, vol. 125, no. 6, pp. 221–224, Jun. 2009.

[22] R. Durbin, S.Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis*. Cambridge University Press , New York, 1998.

[23] S. K. A and C. E. "Keyword spotting system for Tamil isolated words using Multidimensional MFCC and DTW algorithm." In *2015 International Conference on Communications and Signal Processing (ICCSP)*. IEEE, Apr. 2015.

[24] H. A. Javaid, N. Rashid, M. I. Tiwana, and M. W. Anwar. "Comparative Analysis of EMG Signal Features in Time-domain and Frequency-domain using MYO Gesture Control." In *Proceedings of the 2018 4th International Conference on Mechatronics and Robotics Engineering - ICMRE 2018*. ACM Press, Feb. 2018.

[25] S. Khalid, T. Khalil, and S. Nasreen. "A survey of feature selection and feature extraction techniques in machine learning." In *2014 Science and Information Conference*. IEEE, Aug. 2014.

[26] M. N. Murty and V. S. Devi. *Pattern Recognition*. Springer London, 2011.

[27] D. Li, I. K. Sethi, N. Dimitrova, and T. McGee. "Classification of general audio data for content-based retrieval." *Pattern Recognition Letters*, vol. 22, no. 5, pp. 533–544, Apr. 2001.

[28] T. Zhang and C.-C. Kuo. "Audio content analysis for online audiovisual data segmentation and classification." *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 4, pp. 441–457, May 2001.

[29] P. Somervuo, A. Harma, and S. Fagerlund. "Parametric Representations of Bird Sounds for Automatic Species Recognition." *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 6, pp. 2252–2263, Nov. 2006.

[30] L. Shi, I. Ahmad, Y. He, and K. Chang. "Hidden Markov model based drone sound recognition using MFCC technique in practical noisy environments." *Journal of Communications and Networks*, vol. 20, no. 5, pp. 509–518, Oct. 2018.

[31] S. Gupta, J. Jaafar, W. F. wan Ahmad, and A. Bansal. "Feature Extraction Using Mfcc." *Signal & Image Processing : An International Journal*, vol. 4, no. 4, pp. 101–108, Aug. 2013.

[32] F. Zheng, G. Zhang, and Z. Song. "Comparison of different implementations of MFCC." *Journal of Computer Science and Technology*, vol. 16, no. 6, pp. 582–589, Nov. 2001.

[33] C.-H. Min and A. H. Tewfik. "Automatic characterization and detection of behavioral patterns using linear predictive coding of accelerometer sensor data." In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*. IEEE, Aug. 2010.

[34] L. R. Rabiner and R. W. Schafer. "Introduction to Digital Speech Processing." *Foundations and Trends in Signal Processing*, vol. 1, no. 1–2, pp. 1–194, 2007.

[35] S. Zahid, F. Hussain, M. Rashid, M. H. Yousaf, and H. A. Habib. "Optimized Audio Classification and Segmentation Algorithm by Using Ensemble Methods." *Mathematical Problems in Engineering*, vol. 2015, pp. 1–11, 2015.