Classification of Public Radio Broadcast Context for Onset Detection

C. O. B. Weerathunga
Index No : 13001329

Supervised by

Dr. K. L. Jayaratne

Dr. P. V. K. G. Gunawardana

Submitted in partial fulfillment of the requirements of the
B.Sc. in Computer Science (Hons) Final Year Project (SCS4123)

University of Colombo School of Computing
Sri Lanka

December, 2017

Declaration

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

Candidate Name : C.O.B. Weerathunga

Signature of Candidate : ..... Date : .....

This is to certify that this dissertation is based on the work of Ms. C.O.B Weerathunga under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Supervisor Name : Dr. K.L. Jayaratne

Signature of Supervisor : ..... Date : .....

Supervisor Name : Dr. P.V.K.G. Gunawardana

Signature of Supervisor : ..... Date : .....

Abstract

The rapid development of the modern information and communication technologies has influenced the various aspects of the human communication and behavior, including the mass media communication and journalism. This development allowed the deployment of different mass communication applications by motivating people in the content analysis of different communication media. Radio broadcasting can be identified as a communication media which is so close to every citizen in a country. Content analysis in the radio broadcast context for various commercial application development (i.e. news monitoring, song monitoring, speaker recognition etc.) emerged as a major research area which facilitates the broadcast monitoring process.

This dissertation focuses on the investigation of a unified methodology for the onset detection in Sri Lankan radio broadcast context with the approach of classification of the broadcast context. Various audio patterns in the broadcast context were observed and a supervised learning approach was employed in the classification process. Different audio features were examined with respect to the broadcast context. Identified audio semantics in the broadcast context were used in refining the output gained in supervised learning models. Onsets were predicted using the classification results.

The evaluation method was carried out with ground truth data obtained from a Sri Lankan FM broadcast recording. The proposed approach provided the accuracies of 41% for news, 76% for radio commercials, 75% for songs and 59% for other voice related segment classification. The onset detection model was successful in predicting the onsets with an error rate of (+/-) 2.5s with approximately 82% of accuracy level.

The proposed strategy can be easily adapted in broader audio detection and classification tasks including additional real world speech-communication scenarios with some improvements to the proposed classification model.

Preface

Extracting various types of information from audio content is a topic which has considerable amount of research. Extraction of some basic information from the audio sources and utilize them for audio onset detection purposes play a major role under this research domain. There are several different approaches for the audio onset detection. The basic flow in most of them is to extract some basic audio features and come up with a classification model for the separation of audio content between classes.

A novel approach has been proposed for this audio class classification in public radio broadcast context in Sri Lanka. Chapter 3 discusses the novel proposed approach of the content classification with the onset detection technique. The concept used for the onset detection and classification process behind this approach was solely my own work and has not been proposed in any other study related to the classification of content in FM radio broadcast. The applicability of onset detection is novel in this domain. The proposed approach consists of audio feature extraction in a public radio broadcast context and a classification model is employed in the classification process of the radio broadcast context categories. A novel onset detection approach has been proposed for the onset detection in the radio broadcast context.

The results in Chapter 5 are completely depends on the experiments carried out by the researcher. Data gathering, model development and analysis of them are entirely my own work. The analytical calculations carried out were done by me in conjunction with the supervisors.

Acknowledgement

I would like to express my sincere gratitude to my research supervisor, Dr. K.L. Jayaratne, senior lecturer of University of Colombo School of Computing and my research co-supervisor, Dr. P.V.K.G. Gunawardana, senior lecturer of University of Colombo School of Computing for providing me continuous guidance and supervision throughout the research.

I would also like to extend my sincere gratitude to Mr. G.K.A. Dias, senior lecturer of University of Colombo School of Computing and Dr. K.P.M.K. Silva, lecturer of University of Colombo School of Computing for providing feedback on my research proposal and interim evaluation to improve my study. I also take the opportunity to acknowledge the assistance provided by Dr. H.E.M.H.B. Ekanayake as the final year computer science project coordinator.

I appreciate the feedback and motivation provided by my friends to achieve my research goals. This thesis is also dedicated to my loving family who has been an immense support to me throughout this journey of life. It is a great pleasure for me to acknowledge the assistance and contribution of all the people who helped me to successfully complete my research.

Contents

List of Figures

List of Tables

Acronyms

AAC	Advanced Audio Coding
AIFF	Audio Interchange File Format
AM	Amplitude Modulation
ANN	Artificial Neural Network
CBID	Content Based Identification
CNN	Convolutional Neural Network
COG	Center of Gravity
DAB	Digital Audio Broadcasting
DRT	Digital Radio Tracker
FLAC	Free Lossless Audio Codec
FM	Frequency Modulation
GMM	Gaussian Mixture Model
HFC	High Frequency Content
HM	Hardness Measure
HMM	Hidden Markov Model
LE	Local Energy
MFCC	Mel-Frequency Cepstral Coefficient
RNN	Recurrent Neural Network
RMS	Root Mean Square
RTFI	Resonator Time-Frequency Image
SD	Spectral Difference
TSS	Transient Steady State
ZCR	Zero Crossing Rate

Chapter 1

Introduction

1.1 Background to the research

Radio broadcasting plays an important role in today's communication by providing opportunities to people to keep up-to-date on the news and trends. It is a unidirectional wireless transmission over radio waves. Radio stations broadcast different content to receivers. In radio broadcasting the receiver is known as the listener. Large portion of the world population listen to AM/FM radio over the airwaves which is higher than TV viewer-ship, PC use, smart-phone and tablet usage [1].

Information which is given over the radio broadcasts to a large number of listeners. Many countries consist of large number of radio broadcasting channels which transmits different content types. Sri Lanka itself contains around 100 broadcasting channels which are generally being divided along linguistic lines with state and private media operators. Radio broadcasting is important for both developed and developing countries for information provision as well as for entertainment. The overall content which is broadcasted on a radio station can be categorized in to different sections. Some of them are as news, music, commercials, radio dramas, discussions and interviews, sport commentaries, religious programs etc.

Monitoring radio broadcast content is an essential part in every country's broadcasting act. Authorized people in mass media and information corporations, singers, composers, lyricists, advertising agents, government defense organizations and law enforcement authorities might need information in those broadcasting content for different purposes. Composers, singers and lyricists need monitoring of music/songs which are broadcasted in the radio to ensure the copyrights, royalty payments, for security rights of the composition etc. Advertising agents are keen on the frequency of advertisements which were broadcasted and the time period of broadcasting which makes a huge impact for the company's revenue. Regulating authority needs monitoring so as to ensure whether all these broadcasted content goes with the government enforced laws. Government defense organizations and law enforcement authorities keep an alert on the radio broadcast content, especially on news and radio dis-

cussion for their name referencing. With all these requirements of different stakeholders it is clear that an automated monitoring for radio broadcast context is a requirement in today's society.

In monitoring, it is essential to identify the different classes which we can divide the radio broadcast context into. Since a radio stream is continuous it contains all these afore mentioned categories (i.e. news, commercials, songs, radio dramas, phone conversations and other human voice content). In order to identify different classes/categories a well-trained classification model will be required. With the classification model the onsets of different broadcast content can be identified and further tuning of classification results can be performed in identifying the correct events/categories in the broadcast context.

Onset detection is the technique of finding the starting point of the content in an audio context. The proposed approach of this research is also based on onset detection of radio broadcast context with the assist of a classification process of the broadcasting context.

1.1.1 Onset

Onset detection is an active research area in audio processing. Different multimedia intelligent management systems use the techniques of onset detection for the classification of different context in audio files. The main goal of onset detection can be seen as the detection of events in an audio signal [2]. Automatic detection of events in an audio signal will give new possibilities to a number of applications related to information retrieval, proper segmentation of audio content, extraction of important audio features, segmented compression etc. [2][3]

Onset refers to the beginning of a musical note or other sound [4]. It is the moment where the start of an abrupt change in amplitude of a signal occurs. Figure 1.1 shows the 'onset', 'transient', 'attack' and 'decay' of an audio signal [5].

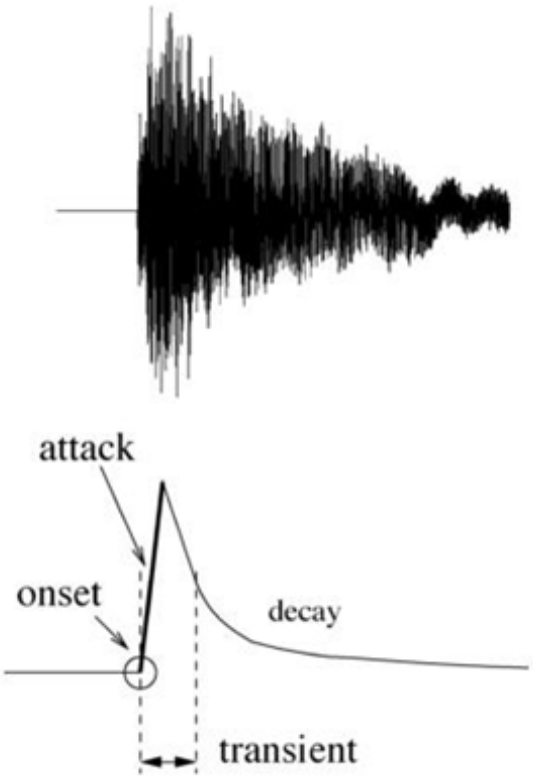

Figure 1.1: 'Onset', 'Transient', 'Attack' and 'Decay'

For realistic situations, it is hard to get an ideal audio signal. In the real world, the signal is polyphonic, which means there might be different sound sources appearing at the same time interval and might contain some noise from the outside environment. So it is difficult to detect the onset locations directly with quantitative time varying in the transient region [6]. Because of that, there are different approaches to find onsets in an audio signal.

Approaches to onset detection can be of different domains as time domain, frequency domain, phase domain and complex domain [4]. In time/temporal domain the onset detection approach is based on the energy of the signal. Here the relative change in energy is considered in building the detection function [5][7]. In the frequency domain the high-frequency content across a signal spectrum is observed and their magnitudes are used in building the detection function [5]. In phase domain, the phase change is observed across fast Fourier transformed frames[5][8]. In complex domain functions, different detection domains are combined to build a single detection function [5][9].

1.2 Research Problem and Research Question

As discussed in section 1.1 there is a massive requirement for the radio broadcast monitoring. For the monitoring purposes, there are several cheap and non-reliable audio analysis

approaches are available in the current environment. Some of those approaches include reading the attached meta-data, asking for the broadcast report from broadcast stations, having a human observer to listen and monitor the content played in a particular broadcast channel etc. Since there is large number of radio channels exist in a country this is highly impractical and will lead to erroneous solutions.

Today there exist some commercially available software solutions for the radio broadcast monitoring process. Some of them are as follows.

1.2.1 Digital Radio Tracker (DRT)

Digital radio tracker [10] is a radio airplay monitoring service which tracks radio airplay of songs on more than 5000+ radio stations around the United States of America and worldwide. The radio stations include major FM terrestrial, college, commercial, non-commercial and internet radio stations. DRT reports allow anyone to get the airplay detection information about a particular song. The significant feature in DRT is that it does not need any special encoding or fingerprinting in the audio streams for the monitoring purpose. DRT has the ability to keep a log on every song played in their monitoring stations and their database provides a very comprehensive report about the songs played in their monitoring radio stations with a history of 8 weeks.

The major disadvantage of this solution is that this is not available for the Sri Lankan radio broadcast context. Not a single broadcast channel in Sri Lanka is being listed in their database. And also this only monitors the songs which are being broadcasted and is not a unified solution for the whole broadcast content monitoring.

1.2.2 ACRCloud Broadcast Monitoring Service

ACRCloud broadcast monitoring service [11] is a web-based broadcast monitoring service which is designed for media monitoring and analysis agencies, labels, broadcasters, media operators, content owners to monitor and measure content's performance and to protect copyright. It is basically designed as an automatic content recognition platform based on acoustic fingerprinting technology. It allows the content owners/ users to upload their content and ingest live stream for their content identification and monitoring.

ACRCloud is also not available in Sri Lankan broadcast context. Since this requires content upload for the monitoring purposes this will not address a global context for broadcast monitoring. Only the uploaded content will be monitored.

1.2.3 BeatGrid Media Monitor

'BeatGrid media' [12] is a software application which enables the users to identify the content in streams and files. The media monitoring application helps in monitoring services with thousands of radio and TV stations. It facilitates advertising verification for broadcast monitoring firms, broadcasters and advertising agencies and brands. It matches any audio or video content within a short period of time and it is scalable to more than thousands of channels. Their broadcast monitoring technologies provide real-time competitor insights of TV and radio advertising. Furthermore 'BeatGrid' monitoring platform facilitates in air-play monitoring for content copyright management by compiling the music hit lists based on most aired songs and artists.

BeatGrid is also not applicable to the Sri Lankan broadcast context as they do not facilitate the Sri Lankan broadcast stations. Since the channel monitoring is done via an audio fingerprinting approach that would need a large content upload for the system for it to get the fingerprints of actual content.

According to the above-discussed materials (Section 1.2.1, Section 1.2.2 and Section 1.2.3), we can identify that there is only a few commercially available software for radio broadcast content monitoring. Almost all of them are not for the Sri Lankan broadcast context and are not freely available. Most applications do not support a unified approach for the broadcast monitoring. Therefore it is clear that a unified methodology for the monitoring of Sri Lankan radio broadcast content has a knowledge gap.

As an initial approach for the broadcast monitoring system, it is essential to identify the different content categories in a FM radio broadcast. Therefore the classification of radio broadcast context and onset detection of it can be identified as the research problem which aids in facilitating a unified methodology to identify different content categories in the Sri Lankan broadcast context.

Considering the aforementioned research problem the generated research questions are as follows.

1.2.4 Research Questions

How can we classify a radio broadcast context into a set of pre-identified content categories?

As discussed, one of the major requirements is to come up with a classification model for the radio broadcast content classification. As a result of that, a new classification model for the Sri Lankan FM radio broadcast context will be implemented by observing the different audio features in it.

How can the onsets of a broadcast context be represented using a unified methodology?

There is no well-established onset detection mechanism for the radio broadcast context. So this research attempts in proposing a unified mechanism to find the onsets of broadcast content.

1.3 Research Aims and Objectives

The main aim of this research is to assist a deep automated analysis for the application levels of the radio broadcast context monitoring process. As the initial step, we attempt to classify the broadcast context into a set of pre-identified content categories and then detect onsets of audio forms. This will lead to the separation of audio events (i.e. news, commercials, songs and other human voice related components) in a public radio broadcast context.

The objectives of the research are as follows.

Objective 1: Classification of broadcast context into different content categories

Public radio broadcast context can be divided into many categories according to the listener's insight. One of the major objectives of this research is to identify some of those content categories in a Sri Lankan broadcast channel. As an initial step, the audio features of those content types will be examined and a classification model will be implemented.

Objective 2: Automated identification of onsets of a radio broadcast context

Onset detection of radio broadcast context plays a major role in commercial application development in broadcast context. The proposed approach will identify the onsets of each content category/type according to the aforementioned classification results.

1.4 Justification of the research

As discussed in Section 1.1 and Section 1.2 it is clear that there is a requirement for a radio broadcast monitoring system. In order to have a proper unified system, it is necessary to identify the relevant broadcast content categories. For the identification of the content categories, the proposed approach uses the audio features in the broadcast stream and the onsets are detected according to the classification results. Through onsets, the audio events (i.e. news, commercials, songs, and other human voice segments) in a radio broadcast context can be separated easily. Chapter 2 will discuss the existing approaches for the onset detection. Once we examine the onset detection approaches it is clear that most of them have addressed it in the domain of music.

So it is clear that there is a knowledge gap in the use of onset detection and classification of content categories in the public radio broadcast context. This research basically targets

on reducing that identified gap.

1.5 Methodology

The major component of this research is finding an onset detection technique which facilitates the onset detection in radio broadcast context for the separation of content types. The proposed onset detection technique follows a classification approach in predicting the onsets in a radio broadcast context. As the initial step, the audio features in a broadcast context were examined and a classification model was designed to get promising accuracies for the content labeling. Semantic rules which were observed in a radio broadcast context is applied in refining the classification results. Finally, the onsets were predicted according to the content changes in the refined classification results. The evaluation model was based on the ground truth data which are being annotated by a human user according to his/her insight on the broadcast content types. Figure 1.2 illustrates the high-level diagram of the proposed approach.

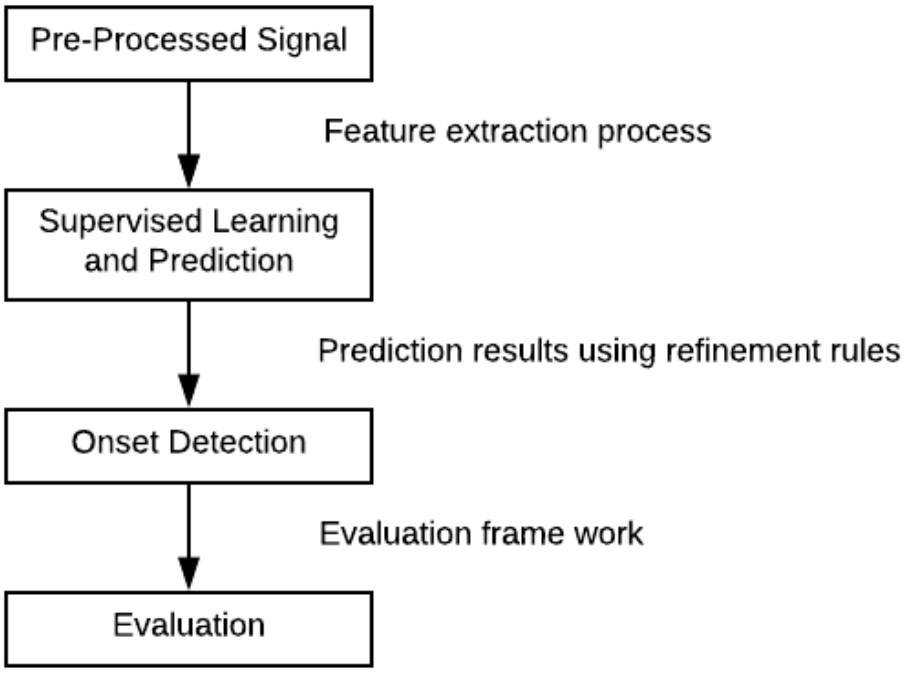

Figure 1.2: Proposed Methodology

1.6 Outline of the Dissertation

This dissertation is organized into six main chapters. A precise introduction to the research is given in Chapter 1. This document proceeds to describe relevant background information

and related work in Chapter 2. Subsequently, in Chapter 3, the proposed methodology will be described including key concerns and design phase. Chapter 4 is dedicated to present the implementation process highlighting dataset, tools, and algorithms with formulas. Chapter 5 will elaborate the evaluation process with all related experiments. Important findings and results will be elaborated in a very precise manner. Final Chapter will explain the conclusion of this research along with the future work.

1.7 Delimitation of Scope

This research targets on the onset detection of the radio broadcast context. A classification approach is followed in detecting the onsets of the radio broadcast context. Sri Lanka's FM broadcast content is analyzed prior to this research and the model will be trained according to the Sri Lankan broadcast context. For the preliminary stage of model construction and evaluation, one broadcast channel is being used.The used channel is known as 'SLBC-Commercial Service'.

Since the FM radio broadcast context is very dynamic in nature the scope of the classification process is narrowed down to a little. As an initial step the scope is limited up to classification of four different content categories as News (i.e. pure news bulleting without any background sound effects), Songs, Radio Commercials and Other pure human voice content (i.e. human voice prominent parts which includes all the phone conversations, radio discussions, radio dramas with voice only parts etc.).

A limited length (maximum 1 hour) of the audio streams will be used in training and testing processes of the model. And also the model will be trained with the aforementioned content categories. All the recordings for the model will be taken in the format of '.wav'.

Jingles (i.e. the small sound effects played in between different broadcast content) are avoided in the initial experimental process and they also considered as radio commercials.

When annotating the dataset the assumption that offset of one content category will be the onset of the other content has been taken into consideration. No silence removals will be done for the audio streams in the pre-processing stage.

1.8 Summary

Extraction of audio events (i.e. news, radio commercials, songs, other human voice components) from a radio broadcast stream can be seen as a major research area which aids in the process of audio monitoring. Even though there are many non-reliable techniques available for the radio broadcast context monitoring there is no unified methodology to detect the content categories of a broadcast context. The main aim of this research is to assist a deep

automated analysis for the application levels of the radio broadcast context monitoring process. As an initial step, this research focuses on the classification and onset detection of radio broadcast context. The knowledge gap in the use of onset detection and classification of radio broadcast context is addressed in this research. The identified research questions focus on the classification of public radio broadcast context into pre-identified content categories and the onset detection of the broadcast content. The proposed approach will be based on audio pre-processing, audio feature extraction and supervised learning approach for the initial classification of the broadcast context, refinement of classification results using a set of observed semantic rules and onset detection using the classification results. The research focuses on a single Sri Lankan radio broadcast channel for the initial model development and evaluation process.

Chapter 2

Literature Review

Auditing the content of audio transmitted by radio stations is of great interest for the government, publicists, musicians and for the managers of radio broadcast stations among others. Monitoring radio broadcast serves several important purposes such as auditing audio marketing campaigns such as advertisements; ranking popular songs, ensure copyrights and royalty payments of music compositions, preventing banned content being broadcasted etc. There are several approaches to this radio broadcast monitoring task such as reading the attached meta-data or simply asking the radio stations for the broadcast report. These solutions are very cheap but they cannot be considered as real audio analysis and therefore they are not reliable [13]. Other approaches such as using human observers are also widely being used in the monitoring process. But they seem impractical and hard to setup with the existence of a large number of radio broadcast channels in a country and that also does not imply real audio analysis and will result in error-prone solutions.

A smarter and more reliable approach to this task will need to classify the broadcast context into pre-identified categories. In the audio classification process, onset detection plays a major role. Many researchers have attempted in different ways in detecting the onset of audio files. Many of them include onset detection in music domain. Relatively fewer attempts have been made in onset detection and classification of radio broadcast context. In the analysis of radio broadcast context, the discrimination between speech/human voice and music/non-speech signals is an important problem. As a result, a lot of research has been conducted in a global context in this area.

The most pioneering research work related to onset detection technique was the Tutorial on Onset Detection by Pablo Bello et al. [5]. In their research, they have stated that the usual way to detect onset is to look for 'transient' regions in the signal. Also, the researchers have stated that if the signal is very percussive, time domain methods are usually adequate and if not, then the spectral methods like phase distributions and spectral differences are adequate. And also they emphasized that a combination of different detection functions might work well with the application levels. Their discussion was basically focused on the

more specific problems of the note onset detection in musical signal. Since the radio broadcast stream consists of so much of variations the direct applications of these methods (detection functions) will not give out good results.

The most common way of onset detection in an audio stream is using an energy based algorithm. Even though it is generally a fast algorithm, its effectiveness becomes low when the transients of the signals are not presented and when the energy of the signals get overlapped with polyphonic mixtures. An alternative to the standard onset detection was proposed by Juan Pablo Bello and Mark Sander [14]. They came up with a phase-based onset detection technique instead of the traditional energy-based detection. Since the phase carries all the timing information of an audio signal and as transients are well-localized in the time domain the proposed approach have made more meaningful results. Researchers have used the statistical measures in building the detection function. Their proposed approach was built upon the phase-based transient / steady-state (TSS) separation and the output data is analyzed by using statistical methods. For the evaluation process, the researcher has used a database of complex real recordings including both percussive and non-percussive instruments. The proposed detection function provided high detection rates for both percussion and non-percussion compositions. According to the evaluated results, the proposed onset detection function will only work for instrumentals. Nature of instrumental compositions will not closely relate to the dynamic fashion of the broadcast context. So having a phase based onset detection function will not give satisfactory results.

The neural network is another approach that has been attempted in onset detection. Jan Schlüter and Sebastian Böck have proposed a novel approach for musical onset detection using a convolutional neural network (CNN) [2] as an alternative to recurrent neural networks (RNN) and hand-designed method. They have used the spectral representation of the wave and onsets were characterized by a swift change of spectral content over time. The use of a separate detector for percussive and harmonic onsets and the combination of results from many minor variations of the same scheme added an extra value to the approach. As mentioned earlier, the researchers have used the spectrogram representation of the wave and they have stated that the onset detection is closely related to the edge detection in images. With the selected musical domain the characterization of onsets through a spectral representation is possible. But the spectrogram representation of a radio broadcast wave would be quite different from the instrumental and musical composition. Hence the identification of onsets from the spectrogram representation would be inefficient. Because of that, this approach may not give the expected results for the onset of different broadcast content categories.

In the process of extracting and classification of radio broadcast context, many researchers have tried to differentiate speech and music from radio streams by using different approaches. Omer Mohsin Mubarak, Eliathamby Ambikairajah, and Julien Epps have attempted in sep-

arating music with speech based on Mel frequency cepstral coefficient (MFCC) and Gaussian mixture model (GMM) as the classifier [6]. Their approach came up with a MFCC-based feature vector and it was computed over 28 critical bands. They have employed two GMMs as one for speech/non-music and the other one for music. They discovered that music and speech gave minimum error rates in a different number of MFCC-based features. In this approach, there was a binary classification as speech and non-speech. For a multi-class classification like distinguishing news, commercials, songs, and other voice categories the direct approach of this cannot be used.

A. Röbel has proposed a new onset detection algorithm based on the classification of spectral peaks into transient and non-transient peaks [15]. In his approach, he has used a statistical model of the classification results to prevent the detection of random transient peaks due to noise. The detection function was developed for a special application where the detection delay should be as short as possible and does not require to find soft onsets. The researcher has followed a two-stage strategy in his approach to onset detection. In the first stage, it classifies the spectral peaks from the DFT spectrum into peaks which belongs to attack transients and non-attack transients. Then in the second step, he employed a statistical model based on the classification results to detect the transient events. The basic approach for the transient detection system was based on the center of gravity (COG) of the time domain signal. He stated that a peak to be detected as a transient COG relevant to that peak is at the far right side of the center of the signal window. Since their algorithm is adapted using drums, plucked string, bells etc. there was a high tendency for their approach to giving high results for those trained compositions. As their training database does not contain a large scope of sound classes; the used threshold values, parameter values, and COG filters can be generalized to other sound classes also.

There have been several attempts at combining signal cues from different detection functions to provide with a more accurate estimation of onsets. In that respect, blackboard modeling, an approach taken from expert systems, has been successfully applied in the field of audio processing. Noberto Degara- Quintela, Antonio Pena, Manuel Sobreira-Seonae and Soledad Torres-Guijarro presented a combination of techniques using a blackboard system for the onset detection process [3]. Their proposed system defines a global sound source separation system where the experts are integrated to. Experts use their own onset detection algorithms and their own analysis strategies. In their approach, they adapted a single blackboard that controls several onset detection methods. They have defined four levels of information as segments, spectra, detection functions and peaks in the blackboard architecture. They and have used the high-frequency content analysis (HFC), the spectral difference (SD) and local energy (LE) calculation for the detection functions. As stated, researchers' proposed approach was successful in multiple lines of reasoning. According to their illustrated onset detection results, the blackboard architecture only detects whether a particular

onset is a hard or soft onset and temporal locations of the onsets cannot be obtained. Since the experimented domain is not specifically mentioned the actual relevancy of the proposed approach for the broadcast content segmentation cannot be stated.

Music note onsets can be classified as 'soft' onsets and 'hard' onsets. Hard onsets show a sudden change in energy while soft onsets show a very gradual change. Even though hard onsets are easily detectable using a time-frequency representation and energy based function, soft onset detection is somewhat hard as sound sources contain noise and oscillations associated with frequency and amplitude modulation [16]. As an approach to distinguish both hard and soft onsets from music pieces Ruohua Zhou and Joshua D. Reiss proposed a methodology for music onset detection by combining energy based and pitch based approaches [16]. Their approach consists of three main stages as time-frequency processing, onset type classification, and detection algorithms. For the time-frequency representation of the music signal, they have used a Resonator Time-Frequency Image (RTFI) which is efficient in computation. For the onset type classification, they have used a 'hardness' measure (HM). If the analyzed signal is above the threshold of HM then it is identified as a hard onset and if not it is considered to be soft onset. Energy based functions are used for the signals identified to be hard onsets and pitch based algorithms are used for the signals with soft onsets. Their approach worked best for the classes of the solo drum, solo brass, and solo wind. The researchers have stated that by combining both energy and phase-based approaches together might work well for other onset detection classes. But the proposed algorithm consumed more running time than other similar approaches.

In many instances, the onsets of musical notes coincide with the amplitude envelope. A segment based onset detection stage for multi-pitch onset detection which consists of a temporal segmentation stage was proposed by Yulong Wan et al. [15]. Their approach deviates a little from the traditional onset detection approaches. The proposed method first detects onsets by a matched filtering on the amplitude envelope and the original audio stream is segmented into small clips using those detected onsets. The pitch energy spectrum is used to detect the onsets in those segmented clips. Finally, they combined all the detected onsets in each segment to get the overall result of the audio file. As stated the temporal segmentation of the audio file effectively reduces the system pressure caused by time-frequency analysis of the whole-file level and more details of the onsets can be obtained by the segmented analysis. According to their findings, the matched filter method was suitable for onset detection of fixed type audio such as piano music and other specific musical instruments. That makes a doubt with the applicability of the proposed method with an audio stream with dynamic changes. The dataset used for the evaluation process of this approach also provides some doubts with the generality of this approach to the other domains like radio broadcast content analysis.

Segmentation of note objects in a real-time context is useful in live performances and audio broadcasting. Even though many music-oriented applications required real-time functionalities only a few have attempted in extracting music objects in real-time [17]. Having that as a motivation Paul Brosseir, Juan Pablo Bello and Mark D.Plumbley proposed an approach for real-time temporal segmentation of note objects in music signals. They have proposed a method for the segmentation of note objects with a very short time of delay. Researchers have used onset detection of notes to find the boundaries of musical notes. Their onset detection approach was combined with an output of a silence detector to produce the onset / offset pairs of the note object. Four onset detection functions based on high-frequency content (HFC), spectral difference, phase deviation and complex domain were used in this proposed approach. A peak picking algorithm with a dynamic thresholding function was used in order to obtain a series of onset times. A silence detector was used to reduce the false positives detected in the low energy areas. Researchers have implemented a small library with a set of processing units: phase vocoder, onset detection functions, and peak pickers for their experiments. Their findings stated that HFC is well suited for the detection of percussive onsets, spectral difference method / complex domain approaches are well suited for tonal / non-percussive onsets. And also they stated that to maximize the number of detections it is good to combine those approaches for a note segmentation algorithm. The experimented dataset included monophonic audio signals of the music domain and high results were gained for the composition of that nature. Radio broadcast recordings have polyphonic and dynamic compositions. So the applicability of the proposed method by these researchers to the broadcast context is uncertain.

Even though there are different onset detection methods, most of them work only in offline mode. The traditional onset detection methods usually use only spectral and/or phase information of a signal. And also many approaches do not employ machine learning techniques and probabilistic information. The approaches presented in [2, 3, 5, 6, 14, 16, 17, 18] usually work in the offline mode as their peak picking algorithms rely on future information to determine the location of the onset. Only a few algorithms were designed to work for online scenarios by aiming to minimize the delay between the onset occurrence and reporting [19]. Having that as a motivation Sebestian Böck, Andreas Arzt, Florian Krebs, and Markus Schedl proposed a novel approach for online real-time onset detection with recurrent neural networks. Their proposed system comprises of three main processing steps as signal pre-processing, neural network onset prediction and peak post-processing. The system takes a discretely transformed audio signal as the input and by three parallel Short-Time Fourier Transforms the discrete signal is transformed into a frequency domain with different window lengths. The processed signal is then input to a recurrent neural network to detect the next occurring onset of the audio stream. A simple post-processing is done to minimize the number of false detections while reporting the onsets. The proposed

approach was successful in achieving performance close to current state-of-the-art offline onset detection algorithms having a zero delay between the onset detection and reporting it.

Many researchers have focused on onset detection and segmentation of content in music compositions. Only a few have paid their concentration of that in radio broadcast context. John Saunders has proposed a novel approach in discriminating speech from music on broadcast FM radio [20]. As stated the proposed algorithm was able to distinguish the two classes as music and speech from broadcast content in real time. Researchers were successful in identifying some of the features which can differentiate speech and music. Some of them are tonality, energy sequences, excitation patterns etc. In their proposed approach they have used the average zero-crossing rate (ZCR) of the time domain waveform with a fixed threshold value to discerning voiced speech. Since there are only two classes they have employed a multivariate Gaussian classifier to decide the class of the test token. Even though the dataset gathered contained different broadcast content categories like talk, commercials and many types of music the paper does not mention any evaluation results. Because of that, the behavior of the proposed approach with the mentioned classes is doubtful.

When considering the FM radio broadcast content radio commercials can be seen as a content type which has a very dynamic and unpredictable behavior. In modern times a larger portion of broadcast content contains advertisements. Shashidhar.G.K et al. have proposed a real-time identification of advertisement segments in radio broadcast [21]. The researchers have identified certain audio features like energy, pitch, duration etc. that present in both advertisements and other audio streams. And also they have observed some other factors related to advertisements in the FM broadcast like the frequency of the occurrence of advertisements, appearance of the custom tune of the radio station before advertisements, pitch and speaking rate of the speaker in advertisements etc. unlike in previous research the researchers have used an ensemble approach with Hidden Markov Model (HMM) and Artificial Neural Network (ANN) for the detection process. The model was constructed with the extracted audio features and concatenating them with the distribution function representing the advertisement density corresponding to the time of delay. The proposed approach has the limitation that the radio DJ utterances were misclassified as advertisements. Since they have concerned on a specific language and a specific channel the approach will not be a generalized solution for all the broadcast channels.

Seneviratna E.D.N.W and Jayaratna K.L have proposed an automated content-based audio monitoring approach for radio broadcasting of Sri Lanka [22, 23]. Their approach consists of a real-time audio recognition algorithm which easily recognizes a song from the broadcast content. Initially, they have identified the song frames from the continuous audio stream using their proposed silent point based algorithm and discard the non-song frames (frames which include radio discussion, news, dramas etc.). By using the method of audio

fingerprinting / content-based identification (CBID) researchers were successful in identifying songs from the radio broadcast context with high accuracies. The research has come up with certain assumptions when deciding the window size for the processing which deviates the solution from a generalized approach.

By examining most of the existing literature we can identify that there are many onset detection approaches which were basically tested and evaluated under the music domain. And also comparatively there are less number of research have been conducted in segmenting the radio broadcast context. Even though there are many approaches to find music over speech, in almost all the approaches they have addressed the instrumental work and not the songs with a human voice. Almost all the approaches were tested on specific datasets which were free from noise and none of them have analyzed in segmenting a continuous stream of audios. Most of the solutions which address radio broadcast domain are also so specific to a certain radio channel and the proposed approaches are not directly applicable to any other cases. Our approach will target on identifying the relevant features to detect the onset of a broadcast context and to find the most suitable/ effective onset detection algorithm for the radio broadcast context segmentations. Moreover, the segmented results will be used in the classification process which will give the relevant class to the segmented content.

Chapter 3

Design

3.1 Introduction

This chapter elaborates the proposed design for the identified research problem. This consists of two major sections as design considerations and design overview. Section 3.2 discusses the facts considered when building the design and in Section 3.3, a detailed description of the design is being discussed with all its steps.

3.2 Design Considerations

3.2.1 Sound may come through both channels (Stereo) or one channel (Mono)

The nature of the input file matters a lot with the accuracy of the model which is being constructed. The FM radio recordings which had been broadcasted on one radio channel are being used to construct this model as well as to test the model.

An FM broadcast recording can come in the format of stereo or mono. In stereo, there can be two or more channels. In stereo channels different parts of the recordings can come in one channel and rest can flow through other channels/sides. In feature extraction, the methodologies will capture only one channel. So if someone introduces the stereo channel to a feature extraction process some of the important features may not get captured by the selected channel in the extraction process. In order to get a good feature vector, the stereo recording should be converted to a mono recording. This process should be done in the pre-processing stage of the audio signal.

3.2.2 Different audio formats

Audio formats can be of many types. There are two different types of audio formats as lossless and lossy. They give out different audio qualities. Lossless audio files keep all the

audio quality of the original source while lossy compress the audio file for space savings.

Wav, AIFF, FLAC, APE are some of the lossless audio file formats. All these formats are uncompressed, which means they are exact copies of the original source audio. For audio editing, it is recommended to use the lossless audio file format as it contains the original audio. MP3, AAC, OGG, WMA are some of the lossy audio file formats. Those formats will save tons of space by compressing the original audio. If someone uses this format for feature extraction there is a high tendency for the loss of valuable features.

As an initial approach, the proposed model will use the recordings of the format '.Wav' to train and evaluate the model.

3.2.3 Audio sample rate

When comes to digital signal processing Nyquist Shannon Theorem is a fundamental theorem which expresses the relationship between the continuous time signal and the discrete time signal. In analog to digital signal conversion, the signal is reproduced to samples. The number of samples per second is known as the sample rate. According to the Nyquist Shannon theorem; if the frequency of an analog signal is f, then the sampling rate must be at least 2f. If the sampling rate is less than 2f some of the highest frequency components in the analog signal will not be represented correctly in the digital output.

The general frequency of a radio broadcast recording will be around 44100Hz. The proposed approach will use the sample rate of 22050 Hz. The frequencies till 11025Hz by covering almost all the high frequency and low-frequency components will be considered in feature extraction processes.

3.2.4 Dynamic variation of the radio broadcast content

Radio broadcast content of Sri Lanka consists of a very dynamic nature. Some of the content types can be identified as news, songs, radio commercials, radio drama and other human spoken words which include phone conversations, radio discussion etc. News can be identified as the general news bulleting in the radio broadcast channels. Within news bulleting, there are different sound effects (i.e. jingles) which are being played. And also there can be radio commercials in between a series of 'News' objects.

If we consider songs there are times where the full song is being played. Sometimes a part of the song is being played instead of the full song. A mix of songs (Radio DJs) or non-stop of songs is being broadcasted. Both old and modern versions of songs are being broadcasted. Some small non-song objects like jingles are included in between song objects. Songs are being played with a series of advertisements, after a series of voice objects etc.

Radio commercials are the most dynamic content type that can be identified in the broadcast context. If we consider Sri Lankan radio stream there are various types of commercials. Some radio commercials are with some background music (not songs). Some of them have no background music and instead they contain some background sound effects. Some of the commercials are advertised by the radio broadcast presenter. There can be advertisements with parts of the songs. That can be either the chorus or melody part of a song. Some commercials start with a song and then lowered its sound to a background music and again at the end, it augments the song. There can be instances where a series of commercials are being played.

When concerning about radio dramas it also got some dynamic behaviors. Radio dramas not solely consist of the human voice. There are instances where the dialogues consist of pure human voice. But there can be instances where background music/song or some other sound effect is being played with the dialogues.

Apart from these the other radio announcements, radio talks, phone calls, radio interviews, magazine programs, radio discussions etc. are all considered to be the human voice. This consists of solely human voice and no background sound effects nor music/songs.

There can be instances where these content types get to mix with noise or some other external effects and all these knowledge has been considered when designing the proposed model.

3.3 Design Overview

As an initial approach the radio broadcast content types were classified into four major categories as news, radio commercials, songs, and voice (i.e. pure human voice with no/minor background sound effects/ voice prominent parts). As mentioned in Chapter 3 there is no established procedure to analyze the dynamic variation of radio broadcast content categories and onset detection of those content categories.

According to the proposed methodology, there are few main state-of-art approaches.They are audio pre-processing, audio windowing, feature extraction and selection, supervised learning model building and training according to the pre-identified categories, evaluation of predicted content categories and finally the onset detection and post-processing. Each stage will be discussed in further sections.

Figure 3.1 shows the overview of the research design.

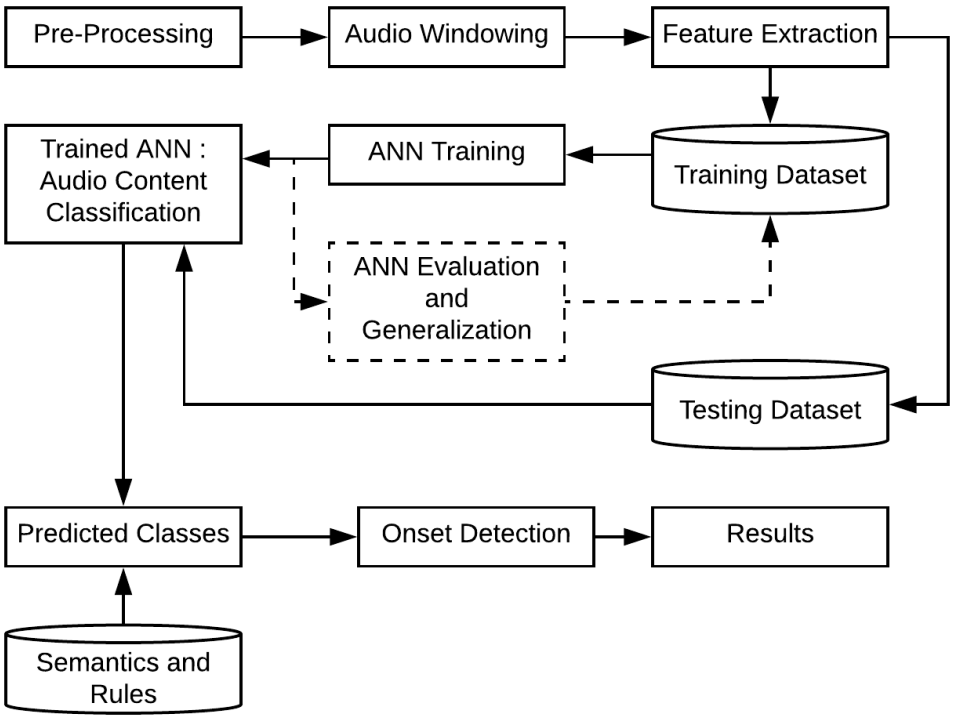

Figure 3.1: Design Overview

3.3.1 Pre-processing

In the preprocessing stage, the acquired signal is converted to the '.wav' format if it is not in the '.wav' format. Since most of the recordings are in stereo format they are being summed to a single channel (i.e. to a mono channel). The sample rate of each and every acquired audio is converted to 22050 Hz to make the consistency for the next stages.

3.3.2 Audio Windowing

Acquired audio signals are then subjected to time-windowing to serve short-term, non-stationary, signal processing. Use of non- overlapping orthogonal windows was selected as the simplest and the less computationally demanding approach. Various window sizes were tested to serve a better audio detection. Smoothing windows (i.e. Hamming, Hanning etc.) were avoided to accelerate the sharp changes and abrupt event detection. By considering the fact that minimum duration for a radio broadcast event is not less than 1- 3s [13], the window size of 2.5s [20, 24] was empirically selected as a good promise for the fine audio detection and classification.

3.3.3 Feature Extraction and Selection

In order to differentiate the pre-identified categories from a given radio broadcast stream, it is necessary to identify the features which can distinguish them from one another. For that feature extraction is very much essential. The selected features were successful in the classification of voice over music/song. Python 'Librosa' library is used in feature extraction process.

Following are the features used in building the model.

Chroma Features

In the context of audio processing, the chroma features or the chromagram is closely related to the twelve different pitch classes. It is also referred to as pitch class profiles. This can be identified as a powerful tool for analyzing the pitch differences of audio context. Chroma features have turned out to be a powerful mid-level feature representation in audio matching.

Mel-Frequency Cepstral Coefficient (MFCC)

In audio processing, the Mel-Frequency Cepstrum (MFC) is a representation of the short-term power spectrum of sound based on the linear cosine transformation of a log spectrum on a non-linear Mel scale of frequency. MFCC are the coefficients which make the MFC. First 13 MFCC features are extracted in-order to build the relevant feature vectors.

Root-Mean-Square (RMS) Energy

The RMS value of a continuous-time waveform is the square root of the arithmetic mean of the squares of the values, or the square of the function that defines the continuous waveform.

Spectral Centroid

In digital signal processing, the 'center of mass' of an audio signal is denoted by Spectral Centroid. The spectral centroid is a good predictor of the 'brightness' of the sound.

$$Centroid = \frac{\sum_{n=0}^{N-1} f(n)x(n)}{\sum_{n-0}^{N-1} x(n)} \tag{3.1}$$

Here, in equation (3.1) x(n) represents the weighted frequency value, or magnitude, of bin number n, and f(n) represents the center frequency of that bin. Each frame of a magnitude spectrogram is normalized and the mean (centroid) is extracted per frame.

Spectral Contrast Features

Spectral contrast is defined as the decibel difference between peaks and valleys in the spectrum.

Spectral-Roll-off

The roll-off frequency is defined as the frequency under which some percentage (cutoff) of the total energy of the spectrum is contained. The roll-off frequency can be used to distinguish between harmonic (below roll-off) and noisy sounds (above roll-off).

Zero Crossing Rate (ZCR)

Zero crossing rate is defined as the rate at which a signal changes its sign from positive to negative or back. Simply it is the rate of sign change along a signal. Generally, ZCRs are used for voice activity detection and also can be used as a primitive pitch detection algorithm.

$$ZCR = \frac{1}{2}\sum_{n=1}^{N} |sign(x[n]) - sign(x[x-1])| \tag{3.2}$$

In equation (3.2) the sign function is 0 for negative arguments and 1 for positive arguments and x[n] denotes the time domain signal of frame t.

Onset Strength

Onset strength function computes the spectral flux onset strength envelope. Spectral flux is the measure of how quickly the power spectrum of a signal is changing and it is generally computed by comparing the power spectrum of a frame with its previous frame.

Tempo

In the audio processing domain, tempo indicates the speed of a given composition. Generally, tempo is measured in beats per minute.

3.3.4 Ground Truth Data Construction

In order to make the ground truth dataset manual annotation of data is required. Sample recordings from a FM radio broadcast will be acquired and it will undergo the previously mentioned pre-processing and feature extraction stages. Time which the content change happens will be marked by listening to the clips carefully. Software named 'Audacity' is used to listen and get the time value of the content change points in the acquired samples. The feature vectors of the ground truth dataset were annotated accordingly with the content change time values. Annotated class labels will be as follows.

- 'News' - News frames
- 'Song' – Song frames
- 'Advert' - Radio commercials / advertisements

- 'Voice' – Voice prominent content in radio discussions, radio interviews, phone calls, radio programs etc.

3.3.5 Artificial Neural Network Construction and Training

Since the proposed approach is based on the supervised learning model it is necessary to train a neural network which is capable of predicting the audio content classes with some high accuracy rates. The Neural network model in 'Keras' library is used in constructing the multi-layer perceptron (supervised learning) model. The configuration of the neural network was done by trial and error method. The neural network structure will be discussed in the next chapter.

The neural network was fine-tuned with the ground truth data to get higher results for the test cases.

3.3.6 Semantics and Rules

Some set of rules have been proposed in-order to refine the prediction results of the ANN. Rules were generated according to the behavior of the selected content types of the FM broadcast stream.

Following are the derived rules.

- A single song frame cannot appear in between two news frames (i.e. a scenario like 'News' 'Song' 'News' cannot appear). They should be transformed into 'News' frames
- A single advertisement frame cannot appear in between two news frames (i.e. a scenario like 'News' 'Advert' 'News' cannot appear). They should be transformed into 'News' frames
- If there's a series of continuous song frames appearing at the middle of the prediction results, then the time duration of that block should be at least 40s. That means there should be at least 16 blocks of continuous song frames if that to be considered as a song
- Song frames can only be refined to advertisement frames
- If there is a series of song frames and with some non-song frames at the middle and if the whole series can add up to at least to 40s then the misclassified frames can be replaced with song frames.

The prediction results are being refined using these rules to enhance the accuracies of the prediction. Refined results are carried forward to onset detection.

3.3.7 Onset Detection

Onsets were identified by analyzing the refined results of the classification stage. The results were considered in a sequential manner and the places which the content change experiences were considered as onsets. Onsets were detected according to an error rate of 2.5s (frame length).

3.3.8 Evaluation

The classification results and predicted onsets were evaluated using the ground truth details of the tested audio clips. Confusion matrices were generated with respect to predicted classification results, refined classification results and detected onsets of the test clips.

3.4 Summary

This chapter presented a detailed description of the proposed design of this research. It elaborated a precise description of the design considerations for the proposed design. Design considerations basically discussed the nature of the radio broadcast stream, the way it affects to the proposed design and the changes to be done to the audio stream for the flexibility of the proposed design. And also this chapter discussed the major steps in the approach including the audio preprocessing, feature extraction, supervised neural network model tuning and onset detection.

Chapter 4

Implementation

4.1 Introduction

This chapter describes the implementation details of the proposed design which was described in Chapter 3. Section 4.2 describes the software tools and libraries used for the implementation. Section 4.3 discusses the implementation details of the system processes.

4.2 Software Tools

The proposed solution was implemented using Python 3.5 with the use of Python Librosa library for audio feature extraction and Python Keras library for the neural network modeling.

4.2.1 Python Librosa

'Librosa' is a python package for audio and music signal analysis. At a high level,Librosa provides implementations of a variety of common functions used throughout the field of music information retrieval. Librosa provides some audio processing submodules which facilitate the audio feature extraction process. Some of the used sub-modules for this research are librosa.beat, librosa.core, librosa.feature, and librosa.onset.

4.2.2 Python Keras

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow. It provides a set of neural network models including RNN, CNN with a modular, user - friendly and easy extensible manner.

The basic neural network model (Sequential model) is applied to this research as an initial setup.

4.3 System Processes

4.3.1 Aquiring the Audio Stream

To acquire the recorded FM broadcast stream a special device named 'Frashback 8' has been used. It was developed by Sonifex Proprietary Limited Company and it offers the ability to replay audio at a remote workstation from logging recorder. It has the capabilities of capturing FM and AM radio broadcast and saving them to '.wma' files.

'Flashback 8' has the ability to record sixty-four stereo line channels, thirty-two stereo FM stations, thirty-two AM stations, four DAB/DAB+ digital radio ensembles and up to thirty-two internet radio streams.

System requirements for 'Flashback 8' are as follows.

- Processor - Pentium 500MHz or faster
- Memory - 128MB minimum
- Operating System – Windows XP or later
- Multimedia audio card supporting 16 bit 48kHz stereo replay
- Network - Ethernet network adapter and drivers supporting TCP/IP protocol

The recordings are of the '.wma' format.

4.3.2 Pre-processing

As discussed in the Chapter 4.3.1 the radio broadcast streams are recorded using the aforementioned device. The recorded files are converted to the '.wav' format for further processing.

As the first step of pre-processing the stereophonic audio files are converted to the monophonic files and as the next step, they have been downsampled to the sample rate of 22050 Hz. The design considerations for the conversion of stereo to a mono signal and the chosen sample rate are discussed in the Chapter 3.2.2 and Chapter 3.2.3 respectively.

Python Librosa library is used in stereo to mono conversion as well as for the downsampling of the audio file. 'Librosa.load' function is used to load the '.wav' stereo file and when 'mono=true' the library converts the file to a mono channel at the moment of loading. The audio file will load as a series of floating points. 'Librosa.resample' function is used to downsample the audio file to 22050Hz.

```
y_old,sr = librosa.load(filename, mono=True, sr = None);
y = librosa.resample(y_old,sr,22050);
sr = 22050;
```

4.3.3 Audio Windowing

Non-overlapping windows are used as the windowing function. As discussed in Chapter 3.3.2 the window size of 2.5s is used as the windowing function. No smoothing windows are being used.

```
frame_duration = 2.5;
frame_length = int(math.floor(frame_duration*sr));
N = len(y);
num_frames = int(math.floor(N/frame_length));
```

4.3.4 Feature Extraction and Selection

All the features are extracted using the python 'Librosa' library. Relevant feature extraction functions can be illustrated as follows.

```
chroma_feature_vec = librosa.feature.chroma_stft(y=y,
sr=sr, n_fft = frame_length, hop_length=frame_length);

mfcc_feature_vec = librosa.feature.mfcc(y=y, sr=sr,
n_mfcc=13, n_fft = frame_length, hop_length = frame_length);

rms_feature_vec = librosa.feature.rmse(y=y,
n_fft = frame_length, hop_length = frame_length);

spectral_centroid_vec = librosa.feature.spectral_centroid(y=y,
sr=sr, n_fft = frame_length, hop_length= frame_length);

spectral_contrast_vec = librosa.feature.spectral_contrast(y=y,
sr=sr, n_fft = frame_length, hop_length = frame_length);

spectral_rolloff_vec = librosa.feature.spectral_rolloff(y=y,
sr=sr, n_fft = frame_length, hop_length = frame_length);

zcr_vec = librosa.feature.zero_crossing_rate(y=y,
frame_length= frame_length,hop_length= frame_length);
```

```
onset_strength_vec = librosa.onset.onset_strength(y=y,
sr = sr, n_fft = frame_length, hop_length= frame_length);

tempo = librosa.beat.tempo(y=y, sr = sr ,
hop_length = frame_length, aggregate= None);
```

The most important parameters passed for the functions are the float representation of the audio file, sampling rate, frame length or the number of samples in an analysis window and the hop length. Since we are using the non-overlapping frames the hop length will be equal to the frame length.

Each feature will give out the corresponding vectors as the output. Table 4.1 illustrates the composition of each feature per frame.

Table 4.1: Composition of extracted features

Feature	**Number of values per frame**
Chroma Features	12
MFCC Features	13
RMS Feature	1
Spectral Centroid	1
Spectral Contrast	7
Spectral Roll-off	1
Zero Crossing Rate	1
Onset Strength	1
Tempo	1
Total number of features	38

After feature extraction, the feature vector with all 38 features will be created and they are being saved as comma separated files (i.e. '.csv' format). Evaluation of the selected features will be discussed in Chapter 5.2

4.3.5 Ground Truth Data Collection

As discussed in the Chapter 3.3.4 ground truth feature vectors were annotated with the relevant labels.

The mid value of the frame time will be considered as the global time representation of the frame. For an instance, if the starting value of a frame is 30.00s and the end value of the same frame is 32.50s then the global time value of that particular frame will be 31.25s.

When annotating the ground truth dataset we get the time values which we experience the content change in the audio file. The frame which the selected time value falls will be taken as the corresponding frame.

For an instance suppose there is a content change at the time value 29.76. Figure 4.1 shows the onset location with respect to the frames. According to Figure 4.1, it is between 27.50s and 30.00s. That means it belongs to 'Frame 3'. Therefore new content 'Content 2' will start from 'Frame 3'.

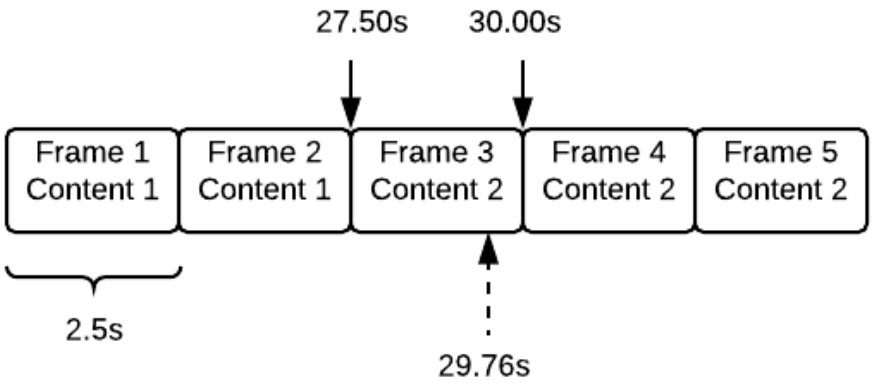

Figure 4.1: Audio Feature Vector Annotation

4.3.6 Supervised Learning Model

Python Keras library has been used in building the neural network model.

The model comprises with the input layer with 38 input nodes, 6 hidden layers with 200 neurons each and one output layer with 4 output neurons. Since there are 38 features the input layer comprises of 38 neurons. As mentioned in Section 3.3.4 there are four categories of broadcast content. So the output layer is designed with four output neurons. The network structure is designed by the trial and error method.

The rectified linear units activation function is used in every layer except the output layer. Softmax activation function is used in the output layer.

Before introducing to the network both the training and testing set data values were normalized. Normalization function can be illustrated as follows.

```
dataset_train = np.loadtxt(filename_train, delimiter=",",
                dtype=object);
dataset_test = np.loadtxt(filename_test, delimiter=",",
                dtype=object);

# Normalizing training and testing datasets
```

```
train_data = dataset_train[1:,1:39].astype(np.float);
norm_data_train = (train_data - np.mean(train_data))/
                  np.std(train_data);

test_data = dataset_test[1:,1:39].astype(np.float);
norm_data_test = (test_data - np.mean(train_data))/
                 np.std(train_data);
```

Code structure for the neural network is illustrated in Appendix A.

4.3.7 Refinement of Prediction Results

Predicted results were refined using the rules defined in the Chapter 3.3.6. The refinement rules are based on the broadcast channel observations.

In the first phase of refinement, the single song frames and single advertisement/radio commercial frames in between two news frames are being changed to 'News' frames. With observation, there cannot be a single song frame in between two news frames and there cannot be a single advertisement/radio commercial frame in between two news frames. With refinement rules, they have transformed into 'News' frames. The refined output is passed to the second phase of the refinement process.

In the second phase of refinement, the song frames are refined. If the difference between two detected song frames is less than 5 then the intermediate content is transformed into song frames. The refined output is passed to the third phase of refinement.

In the third phase, the song frames are refined according to the observation that a song should last at least 40s (16 frames) of duration. If not, the song frames are refined to the advertisement/radio commercials frames ('Advert' frames).

Code fragments for the refinement rules are illustrated in Appendix A.

4.3.8 Onset Detection

According to the refined results from the broadcast content classification, onsets are predicted. When analyzing the classification output sequentially, if there is a content change in those results, then there can be an onset between those two frames with the error rate of (+/-) 2.5s. Figure 4.2 illustrates the onset detection methodology followed in the proposed model.

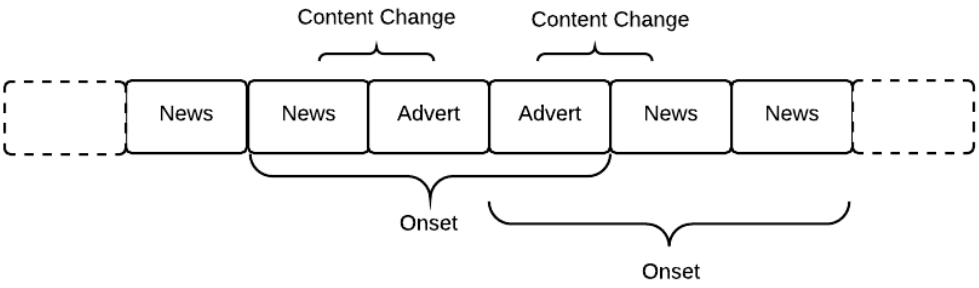

Figure 4.2: Onset Detection Methodology

4.4 Summary

This chapter elaborates a detailed description of the implementation details of the proposed approach. The software tools and libraries which were employed for the model construction are described in the first part of this chapter. Secondly, the implementation details of the major components in the proposed design including the signal acquiring, pre-processing, feature extraction and supervised learning model, classification result refinement, and onset detection are discussed in detail. Relevant code fragments for this chapter are illustrated in Appendix A.

Chapter 5

Results and Evaluation

5.1 Introduction

This chapter discusses how the results are obtained from the previously discussed implementation details. There is no generalized model to evaluate the results gained for the onset detection and classification of content in radio broadcast context. The proposed evaluation model will be discussed in Section 5.2 and Section 5.3 will elaborate the evaluation results.

5.2 Evaluation Model

The evaluation process of the model will be based on ground truth data. Radio broadcast recordings of the selected Sri Lankan broadcast channel ('SLBC- Commercial Service') will be taken and it will be manually annotated according to the identified content changes. The frames were annotated as 'News', 'Advert', 'Song' and 'Voice'. Chapter 3.3.4 gives a detailed description about the annotated categories. A software tool named 'Audacity' is being used in the annotation process.

Accuracies for the test results will be obtained with respect to confusion matrices. The statistical values obtained from the classification results before refinement and after refinement will be used in the evaluation of the classifier. Similarly, a confusion matrix for the onsets will be obtained with respect to the ground truth annotations. The proposed onset detection technique depends solely on the classification accuracies. Therefore the training dataset plays an important role in the classification output.Training dataset contains the manual annotation of content categories of the selected radio broadcast channel.

5.3 Training Dataset

The training dataset comprises of the total of $20,000$ manually annotated feature vectors generated from the FM broadcast recordings of the Sri Lankan broadcast channel 'SLBC-

Commercial Service'. Table5.1 illustrates the composition of the training dataset.

Table 5.1: Training dataset composition

Frame Type	**Number of annotated frames**	**Percentage**
Advert	5000	25%
News	5000	25%
Song	5000	25%
Voice	5000	25%
Total	20000	

An equal number of frames for each content category is chosen to reduce the biases of the classifier for a particular content type.

5.4 Testing Dataset

Testing set contains two recordings of the length of 1 hour from the broadcast channel 'SLBC - Commercial Service'. One recording in the testing set contains all the defined content categories. Table5.2 contains the composition of the testing set I and Table5.3 contains the composition of the testing set II. Statistical measures of the average values of the testing results will be taken to obtain the correct accuracies of the model.

Table 5.2: Testing dataset I composition

Frame Type	**Number of annotated frames**	**Percentage**
Advert	460	31.97%
News	295	20.50%
Song	544	37.80%
Voice	140	9.73%
Total	1439	

Table 5.3: Testing dataset II composition

Frame Type	**Number of annotated frames**	**Percentage**
Advert	135	9.38%
News	10	0.69%
Song	440	30.56%
Voice	855	59.36%
Total	1440	

5.5 Results

5.5.1 Testing Frame Size

To validate the frame length a series of experiments have been carried out. As discussed in Chapter 3.3.2 the frame sizes were selected with the frame lengths of size 1.0s, 1.5s, 2.0s, and 2.5s for the initial testing processes. A broadcast recording clip of duration 1 hour from the channel 'SLBC- Commercial Service' has been used to extract features. Feature vectors were taken to perform the k-means clustering.

R studio software environment has been used for the clustering purposes.Table5.4 - 5.7 elaborates the relevant classification results.

Table 5.4: K-means classification results for a broadcast recording with frame size of 1.0s

	1.0s			
	1	2	3	4
Advert	48	**153**	28	114
News	19	**29**	0	22
Song	782	**1186**	417	448
Voice	**76**	49	66	42

Table 5.5: K-means classification results for a broadcast recording with frame size of 1.5s

	1.5s			
	1	2	3	4
Advert	85	27	**100**	17
News	15	11	**20**	0
Song	293	452	**809**	272
Voice	45	**76**	53	67

Table 5.6: K-means classification results for a broadcast recording with frame size of 2.0s

	2.0s			
	1	2	3	4
Advert	41	**62**	14	55
News	**15**	8	1	11
Song	**410**	242	328	407
Voice	**47**	40	55	23

Table 5.7: K-means classification results for a broadcast recording with frame size of 2.5s

	2.0s			
	1	2	3	4
Advert	8	**48**	42	39
News	2	9	**10**	7
Song	**297**	287	230	295
Voice	41	22	27	**46**

According to the above-illustrated details, the frame length of 2.5s shows promising results for the K-means clustering. Therefore the frame length of 2.5s has been chosen for the feature extraction process in making the training dataset.

The training set for the classification model will be constructed based on the abovejustified frame length.

5.5.2 Training Set Validation

As discussed in Chapter 5.3 training dataset was selected with an equal number of feature vectors from each selected content category. K-means clustering is applied to the whole training dataset to validate the evenness of the feature vectors. Table5.8 illustrates the k-means results of the training dataset.

Table 5.8: K-means classification results for the training dataset

	Training dataset			
	1	2	3	4
Advert	1046	**1994**	1655	305
News	1389	1735	**1758**	118
Song	920	**1775**	1661	644
Voice	**1857**	658	1127	1358

In the training dataset majority of the frames of advertisements/radio commercials and songs hit to the same class. The major reason for that is in Sri Lankan broadcast context mostly the advertisements/radio commercials have the nature of the songs.Most advertisements contain musical parts which have a high tendency to the misclassification with the songs. Therefore both the categories of songs and advertisements/ radio commercials have a high tendency for the misclassification among those two categories.

5.5.3 Feature Selection and Neural Network Validation

The proposed classification model is totally based on the audio features. Chapter 3.3.3 discussed the audio features which have extracted initially from the classification model. Feature selection methods have been employed in order to select the best set of features for the classifier. All the aforementioned features (i.e. 38 features mentioned in the Chapter 3.3.3) were ranked by using the feature selection algorithms named InfoGainAttributeEval and OneREttributeEval [13]. A software tool named 'Weka' has been used in feature ranking process.

Table5.9 illustrates the ranked features from the mentioned feature ranking algorithms.

Table 5.9: Feature Ranking from InfoGainAttributeEval and OneRAttributeEval

Rank	**InfoGainAttributeEval**	**OneRAttributeEval**
1	Spectral Contrast 4	Spectral Contrast 3
2	Spectral COntrast 3	Spectral COntrast 4
3	Spectral Contrast 2	Spectral Centroid
4	MFCC 9	Spectral Contrast 2
5	Spectral Roll-off	MFCC 2
6	Spectral Centroid	MFCC 9
7	MFCC 2	Spectral Contrast 5
8	Spectral COntrast 5	Spectral Roll-off
9	MFCC 7	Spectral Contrast 1
10	MFCC 8	MFCC 8
11	MFCC 13	MFCC 4
12	MFCC 4	MFCC 7
13	MFCC 3	Zero Crossing Rate
14	Spectral Contrast 1	MFCC 13
15	Zero Crossing Rate	Spectral Contrast 7
16	Spectral Contrast 7	MFCC 3
17	MFCC 10	MFCC 10
18	Onset Strength	Chroma Features 6
19	Chroma Feature 6	Onset Strength
20	Chroma Feature 5	Chroma Feature 9
21	Chroma Feature 7	Spectral Contrast 6
22	MFCC 6	Chroma Feature 5
23	Chroma Feature 9	Chroma Feature 4
24	Chroma Feature 10	Chroma Feature 2

25	Chroma Feature 2	Chroma Feature 7
26	Spectral Contrast 6	MFCC 1
27	Chroma Feature 4	MFCC 6
28	Root Mean Square	Chroam Feature 10
29	Tempo	Chroma Feature 8
30	Chroma Feature 8	Root Mean Square
31	Chroma Feature 12	Chroma Feature 12
32	Chroma Feature 1	MFCC 5
33	MFCC 1	Chroma Feature 1
34	Chroma Feature 11	MFCC 12
35	MFCC 12	Chroma Feature 3
36	Chroma Feature 3	Chroma Feature 11
37	MFCC 5	Tempo
38	MFCC 11	MFCC 11

From both the aforementioned algorithms the order of the ranked features is almost the same except for few cases. Therefore for the feature selection the first 20, first 30 and all 38 features from the InfoGainAttributeEval ranking are chosen to decide the best suitable neural network.

The neural network has been trained with trial and error basis. The composition of the test dataset for the network validation is illustrated in Table5.10. A different number of hidden layers with the neuron count of 200 [25] and 300 each with the activation function 'ReLu' has been experimented. The network is trained for 300 epochs. Each network model is being tested with three trials and the averaged values are being taken as the accuracies. Table5.11 discusses the average prediction results for the used test dataset.

Table 5.10: Composition of the dataset used in neural network validation

	Number of feature vectors
Advert	460
News	295
Song	544
Voice	14
Total	1439

Table 5.11: Neural network prediction results for the test dataset

			Training dataset			
			3	4	5	6
38 Features	200 Neurons in each hidden layer	News	75.690	78.002	66.835	**55.107**
		Advert	60.408	60.844	67.631	**68.996**
		Song	59.924	58.946	50.689	**77.145**
		Voice	79.048	77.857	80.179	**80.952**
	300 Neurons in each hidden layer	News	64.310	**73.625**	63.524	52.525
		Advert	65.884	**67.467**	72.926	65.211
		Song	61.857	**58.885**	61.826	69.914
		Voice	80.714	**80.000**	66.667	74.762
30 Features	200 Neurons in each hidden layer	News	78.227	**67.677**	57.351	44.108
		Advert	57.205	**63.464**	73.872	70.524
		Song	32.843	**59.681**	30.637	60.417
		Voice	74.762	**79.524**	77.143	75.714
	300 Neurons in each hidden layer	News	74.523	**55.219**	45.230	43.098
		Advert	58.661	**75.619**	80.349	70.160
		Song	58.333	**65.686**	29.044	81.985
		Voice	76.190	**77.857**	80.714	69.048
20 Features	200 Neurons in each hidden layer	News	45.667	51.403	31.987	26.936
		Advert	71.397	70.670	**79.913**	75.400
		Song	72.794	52.757	**63.419**	43.137
		Voice	46.429	65.000	**66.429**	78.810
	300 Neurons in each hidden layer	News	55.668	**48.373**	28.620	15.937
		Advert	68.122	**74.381**	79.767	73.071
		Song	31.979	**55.821**	57.196	74.877
		Voice	76.190	**70.714**	74.762	70.952

In the above-displayed results, highest average accuracies obtained at each network model is highlighted (written in bold). According to those figures, the most promising results were obtained in the network model with 38 features, 6 hidden layers with 200 neurons in each hidden layer.

Therefore the final network can be considered as a neural network with 6 hidden layers with 200 neurons in each layer with the activation function 'ReLu'.

5.5.4 10-Fold Cross Validation for the Training Dataset

Once the network structure was decided, 10-fold cross validation was carried out for the training set to get an averaged prediction accuracy of the designed neural network. Table5.12 illustrates the prediction results obtained after 10-fold cross validation of the training dataset. According to the averaged accuracy value we can say that the neural network model is capable in predicting the selected content categories with an accuracy of 58%.

Table 5.12: 10-Fold cross validation accuracies

Iteration	**Accuracy Value (%)**
1	48.35
2	63.60
3	61.10
4	57.20
5	58.10
6	54.20
7	61.10
8	58.80
9	62.50
10	57.35
Average accuracy	$58.23(+/-4.24)$

5.5.5 Prediction Result Refinement Using Semantics and Rules

Test datasets in Chapter 5.4 are being used in the evaluation of the model. For each test dataset, the model is executed for four times and the average results have been taken in calculating the accuracy measures.

At first, the classification results are being evaluated. Then the classification results after the refinement of news frames are being evaluated. At last the final refinement results are being analyzed. The overall results of the four iterations are being analyzed to get the confusion matrices.

Test Results for Test dataset I

Initial prediction results of the dataset I for all four iterations with the true positive percentages are displayed in Table5.13. Average prediction results are displayed in Table5.14.

Table 5.13: Classification results for the test dataset I

		News	Advert	Song	Voice	True Positives (%)
Iteration 1	News	**110**	168	4	15	37.0370
	Advert	17	**329**	90	22	71.8341
	Song	0	66	**477**	1	87.6838
	Voice	3	25	0	**112**	80
Iteration 2	News	**151**	126	3	17	50.8418
	Advert	33	**346**	52	27	75.5459
	Song	2	140	**401**	1	73.7132
	Voice	4	16	7	113	80.7143
Iteration 3	News	**177**	104	15	1	59.5960
	Advert	40	**297**	117	4	64.8472
	Song	0	33	**511**	0	93.9338
	Voice	12	46	24	**58**	41.4286
Iteration 4	News	**204**	80	2	11	68.6869
	Advert	80	**304**	50	24	66.3756
	Song	0	137	**407**	0	74.8162
	Voice	9	15	7	**109**	77.8571

Table 5.14: Average classification results for the test dataset I

	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Average Accuracy (%)
News	37.0370	50.8418	59.5960	68.6869	**54.0404**
Advert	71.8341	75.5459	64.8472	66.3756	**69.6507**
Song	87.6838	73.7132	93.9338	74.8162	**82.5368**
Voice	80	80.7143	41.4286	77.8571	**70**

Initial predictions give promising results for the song classification. Even though the results are a bit low for the 'News' in all iterations when compared to other content classes, the average accuracy of 'News' prediction is still above 50%.

Classification results for four iterations after the first refinement phase is illustrated in Table5.15 and similarly, the average accuracy levels are illustrated in Table5.16.

Table 5.15: Classification results after 'News' refinement phase for the test dataset I

		News	Advert	Song	Voice	True Positives (%)

Iteration 1	News	**134**	144	4	15	**45.1179**
	Advert	20	326	90	22	71.1790
	Song	0	66	477	1	87.6838
	Voice	3	25	0	112	80
Iteration 2	News	**182**	95	3	17	**61.2795**
	Advert	37	342	52	27	74.6725
	Song	2	140	401	1	73.7132
	Voice	4	16	7	113	80.7143
Iteration 3	News	**208**	74	14	1	**70.0337**
	Advert	48	289	117	4	63.1004
	Song	2	140	401	1	73.7132
	Voice	4	16	7	113	41.4286
Iteration 4	News	**231**	53	2	11	**77.7778**
	Advert	85	299	50	24	65.2838
	Song	0	137	407	0	74.8162
	Voice	9	15	7	109	77.8571

Table 5.16: Average classification results for the test dataset I after 'News' refinement phase

	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Average Accuracy (%)
News	45.1179	61.2795	70.0337	77.7778	**63.5522**
Advert	71.1790	74.6725	63.1004	65.2838	**68.5590**
Song	87.6838	73.7132	93.9338	74.8162	**82.5368**
Voice	80	80.7143	41.4286	77.8571	**70**

With the refinement process of 'News' frames in the predicted results, the true positives of the 'News' prediction have been increased. When compared the overall accuracies with the previous prediction accuracies (i.e. before refinement process) the accuracies obtained after the refinement process has a considerable increment.

Once the refinement process for 'News' frames has been completed the result set is passed to the song refinement phase.Table5.17 illustrates the classification results after the song refinement phase. That can be seen as the final classification results and similarly, the average accuracies in this phase are illustrated in Table5.18.

Table 5.17: Classification results after 'Song' refinement phase for the test dataset I

		News	Advert	Song	Voice	True Positives (%)
Iteration 1	News	134	148	0	15	45.1179
	Advert	19	338	80	21	73.7991
	Song	0	3	**541**	0	**99.4485**
	Voice	2	20	6	112	80
Iteration 2	News	182	98	0	17	61.2795
	Advert	36	342	53	27	74.6725
	Song	0	74	**470**	0	**86.3971**
	Voice	2	14	11	113	80.7143
Iteration 3	News	208	88	0	1	70.0337
	Advert	45	264	147	2	57.6419
	Song	0	0	**544**	0	**100**
	Voice	7	48	27	58	41.4286
Iteration 4	News	231	55	0	11	77.7778
	Advert	85	320	29	24	69.8690
	Song	0	73	**471**	0	**86.5809**
	Voice	7	20	4	109	77.8571

Table 5.18: Average classification results for the test dataset I after 'Song' refinement phase

	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Average Accuracy (%)
News	45.1179	61.2795	70.0337	77.7778	**63.5522**
Advert	73.7991	74.6725	57.6419	69.5809	**68.5590**
Song	99.4485	86.3971	100	86.5809	**93.1066**
Voice	80	80.7143	41.4286	77.8571	**70**

When analyzing all the results in all the phases it is clear that the classifier is good at identifying 'Song' frames over other frames. All the other categories give lower accuracy levels when compared to the accuracy values obtained for 'Song' frames. That is basically of the high misclassification rates of those categories. When analyzing the results it is clear that most of the time the 'News' and 'Voice' frames have been misclassified with advertisement/radio commercial frames and most of the time the 'Advert' frames have misclassified mainly with 'Song' frames. This is basically because of the dynamic nature of the advertisements/ radio commercials in the Sri Lankan broadcast context. Radio commercials in Sri Lankan broadcast context contain voice parts as well as song/ music parts. The annotations were based on the human knowledge. Human has the capabilities to identify the

broadcast advertisements/ commercials over a song, news segment or other voice segments of radio discussions, phone conversations etc. But the trained classifier solely focuses on the audio features and it has no knowledge base in identifying one content category over another category. Therefore this leads the way for the higher misclassification rates. Even though there are misclassification rates, the average accuracies of all true predictions are above 50% which can be seen as an acceptable value.

Once the classification results are obtained the onsets of the broadcast context are determined. Whenever there is a content change in the classification results the changed frame will be taken as an onset. Table5.19 illustrates the confusion matrices for the one to one onset detection for every iteration. Table5.20 depicts the accuracy measures of one-to-one onset detection for test dataset I.

Table 5.19: Confusion Matrix for one-to-one onset detection for test dataset I

		Predicted onset	Predicted non-onset
Iteration 1	Actual onset	16	24
	Actual non-onset	147	1253
Iteration 2	Actual onset	19	21
	Actual non-onset	153	1246
Iteration 3	Actual onset	8	32
	Actual non-onset	138	1261
Iteration 4	Actual onset	20	20
	Actual non-onset	159	1240

Table 5.20: Accuracy measures for one-to-one onset detection for test dataset I

	Precision	Recall	Specificity	False Negative Rate	Accuracy
Iteration 1	9.8160	40	89.4925	60	88.1168
Iteration 2	11.0465	47.5	89.0636	52.5	87.9083
Iteration 3	5.4795	20	90.1358	80	88.1862
Iteration 4	11.1732	50	88.6347	50	87.5608
Average	**9.3788**	**39.375**	**89.3317**	**60.625**	**87.9430**

According to the summarized results of the onset detection in the test dataset I the true

positive rate is at a considerable low level. That is because the prediction results were unable to identify the exacts onset frames which coincide with the actual annotations of onset. If the predicted onsets were considered with an error rate of (+/-) 2.5s from either side of the frame then the accuracy levels of correct onset predictions will be higher. Table5.21 illustrates the onset values with the (+/-) 2.5s error rate of each iteration and Table5.22 illustrates the accuracy measures for the aforementioned prediction results.

Table 5.21: Confusion Matrix for onset detection with (+/-) 2.5s error rate for test dataset I

		Predicted onset	Predicted non-onset
Iteration 1	Actual onset	25	15
	Actual non-onset	138	1261
Iteration 2	Actual onset	25	15
	Actual non-onset	147	1252
Iteration 3	Actual onset	14	26
	Actual non-onset	132	1267
Iteration 4	Actual onset	26	14
	Actual non-onset	153	1246

Table 5.22: Accuracy measures for onset detection with (+/-)2.5s error rate for test dataset I

	Precision	Recall	Specificity	False Negative Rate	Accuracy
Iteration 1	15.3374	62.5	90.1358	37.5	89.3676
Iteration 2	14.5349	62.5	89.0636	37.5	88.7423
Iteration 3	9.5890	35	90.5647	65	89.0202
Iteration 4	14.5251	65	89.0636	35	88.3947
Average	**13.4966**	**56.25**	**89.8142**	**43.75**	**88.8812**

When the onsets were detected with an error rate of (+/-) 2.5s the overall accuracy measures have been improved.

Test Results for Test dataset II

Table5.23 illustrates the average accuracy results for the initial prediction; Table5.24 illustrates the average accuracy values of the classification results after refinement phase I and Table5.25 illustrates the accuracies final classification results for the test dataset II.

Table 5.23: Average initial classification results for the test dataset II

	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Average Accuracy (%)
News	20	20	10	20	**17.5**
Advert	75.5556	75.5556	73.3333	56.2963	**70.1852**
Song	31.1364	39.5455	51.1364	66.1334	**46.9886**
Voice	60.5386	64.5199	20.6089	42.5059	**47.0433**

Table 5.24: Average classification results after 'News' refinement phase for the test dataset II

	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Average Accuracy (%)
News	20	20	10	20	**17.5**
Advert	75.5556	75.5556	72.5926	56.2963	**70**
Song	31.1364	39.5455	51.1364	66.1334	**46.9886**
Voice	60.5386	64.5199	20.6089	42.5059	**47.0433**

Table 5.25: Average final classification results for the test dataset II

	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Average Accuracy (%)
News	20	20	10	20	**17.5**
Advert	88.8889	90.3704	96.2963	56.2963	**82.9630**
Song	33.4091	53.4091	63.1818	73.4091	**55.8523**
Voice	60.5385	64.5199	20.6089	42.5059	**47.0433**

Observations of the composition of this clip conclude that it contains only a few number of 'News' labels, large number of 'Voice' labels and a moderate amount of 'Song' and 'Advert' frames. The overall accuracy of 'News' frames is very low in this case. It might be because of the nature of the news content in the audio clip. If that clip contains a news bulleting which belongs to an hourly news bulleting scenario then there might be these sort of misclassifications with other content categories. As mentioned before the misclassification rates of 'Song' and 'Advert' frames within those two classes can be higher in Sri Lankan broadcast context. So a well-précised training dataset will be needed in avoiding the misclassifications between these two classes.

Table5.26 elaborates the one-to-one onset detection in the test dataset II and Table5.27 elaborates the accuracy measures of one to one onset detection.

Table 5.26: Confusion Matrix for one-to-one onset detection for test dataset II

		Predicted onset	Predicted non-onset
Iteration 1	Actual onset	8	5
	Actual non-onset	452	974
Iteration 2	Actual onset	9	4
	Actual non-onset	415	1011
Iteration 3	Actual onset	5	8
	Actual non-onset	237	1189
Iteration 4	Actual onset	7	6
	Actual non-onset	365	1061

Table 5.27: Accuracy measures for one-to-one onset detection for test dataset II

	Precision	Recall	Specificity	False Negative Rate	Accuracy
Iteration 1	1.7391	61.5385	68.3030	38.4615	68.2418
Iteration 2	2.1226	69.2308	70.8976	30.7692	70.8826
Iteration 3	2.0661	38.4615	83.3801	61.5385	82.9743
Iteration 4	1.8817	53.8462	74.4039	46.1539	74.2182
Average	**1.9524**	**55.7692**	**74.2461**	**44.2308**	**74.0792**

Even though the classification results are not very much promising the recall/ true positive rate lies above 50% as an average score. The clip contains only a few actual onsets. But the classification results predicted a larger number of values for onsets. The misclassification between content categories paved the way for a low precision value. If the approximated onsets values are considered, then the average recall value will get higher.

Table5.28 illustrates the confusion matrix for onset detection with (+/-) 2.5s error rate in each iteration and Table5.29 illustrates the accuracy measures for the aforementioned prediction results.

Table 5.28: Confusion Matrix for the onset detection with (+/-)2.5s for test dataset II

		Predicted onset	Predicted non-onset
Iteration 1	Actual onset	8	5
	Actual non-onset	452	974
Iteration 2	Actual onset	9	4
	Actual non-onset	415	1011
Iteration 3	Actual onset	5	8
	Actual non-onset	237	1189
Iteration 4	Actual onset	7	6
	Actual non-onset	365	1061

Table 5.29: Accuracy measures for onset detection with (+/-)2.5s error rate for test dataset II

	Precision	Recall	Specificity	False Negative Rate	Accuracy
Iteration 1	2.6087	92.3077	68.5835	7.6923	89.3676
Iteration 2	2.8302	92.3077	71.1080	7.6923	71.2030
Iteration 3	2.8926	53.8462	83.5203	46.1539	83.2523
Iteration 4	2.1505	61.5385	74.4741	38.4614	74.3572
Average	**2.6205**	**75**	**74.4215**	**25**	**74.4267**

5.6 Discussion

According to the aforementioned classification details, it is visible that the 'News' frames do not get misclassified as 'Song' frames in a majority and 'Song' frames do not get misclassified as 'News' frames in the majority. The constructed model is successful in differentiating the 'News' frames from 'Song' frames. There were so many situations where 'Song' frames got misclassified as 'Advert' and vice versa. That is basically because of the dynamic nature of the radio commercials (i.e. 'Advert' frames). A more precise technique which aids in identifying the structural similarities of these selected content categories [26, 27] would help in improving the classification results. And also building a knowledge base on top of the classification model would do better in the classification of the content categories.

When examining the onset detections it is visible that in many instances the content change from 'Voice' to 'Song' is identified with high accuracies. But at times the classification results may not display them as a change from 'Voice' to 'Song' so accurately. High accuracies can be obtained if the onset change is observed with (+/-) 2.5s of error rate. At many instances, one to one onset detection does not happen frequently. It is clearly visible that the predicted onsets have moved one frame up or down from the actual onset values. Therefore in the detection of the onsets of the radio broadcast content, the onsets with the (+/-) 2.5s error rate should be considered. To reduce the false positives of the predicted onsets the classification model should further tuned to get good classification results for the identified content categories.

After evaluating the above discussed two test scenarios (Chapter 5.5.5 and Chapter 5.5.5) the average accuracies of the model can be illustrated as follows. Table5.30 elaborates the overall classification results of the proposed classification model; Table5.31 elaborates the one-to-one onset detection accuracies of the proposed model and Table5.32 elaborates the onset detection accuracies of the model with (+/-) 2.5s of error rate.

Table 5.30: Overall classification results of the proposed model

	Test Case I	Test Case II	Average Prediction Accuracies (%)
News	63.5522	17.5	40.5261
Advert	68.9956	82.9630	75.9793
Song	93.1066	55.8523	74.4795
Voice	70	47.0433	58.5217

Table 5.31: Overall one-to-one onset detection acuracies of the proposed model

	Precision	Recall	Specificity	False Negative Rate	Accuracy
Test Dataset I	9.3788	39.3750	89.3317	60.6250	87.9430
Test Dataset II	1.9524	55.7692	74.2461	44.2308	74.0792
Average	**5.6656**	**47.5721**	**81.7889**	**52.4279**	**81.0111**

Table 5.32: Overall accuracies of onset detection of the proposed model with (+/-) 2.5s

	Precision	Recall	Specificity	False Negative Rate	Accuracy
Test Dataset I	13.4966	56.2500	89.8142	43.7500	88.8811
Test Dataset II	2.6204	75.000	74.4215	25.000	74.4267
Average	**8.0586**	**65.6250**	**82.1178**	**34.3750**	**81.6539**

5.7 Summary

This chapter discusses the experimental results and findings with regard to the classification of radio broadcast content into pre-identified categories and onset detection of it. Experiments to validate the frame size, feature selection, and network structure organization have been carried out prior to the evaluation process. According to the experimental results, the frame size of length 2.5s has been chosen for the further processing. Thirty eight (38) features were chosen from the feature selection process and a neural network with 6 hidden layers with 200 neurons in each layer had been chosen as the network architecture for the classification model. The proposed model was evaluated with two test datasets. The proposed model was successful in predicting the identified content categories with the approximated accuracies of 41% for news, 76% for radio commercials, 75% for songs and 59% for other voice-related segments. The proposed approach for onset detection was successful in predicting the onsets with an error rate of (+/-) 2.5s with approximately 82% of accuracy level. Improvements to the classifier will lead to good accuracy levels in the proposed approach.

Chapter 6

Conclusions

6.1 Introduction

This chapter discusses the conclusions which can be arrived at the research questions, aims, and objectives, limitations that can be seen in the current research and the implications for further research.

6.2 Conclusions about Research Questions

The main aim of this research is to assist a deep automated analysis for the application levels of the radio broadcast context monitoring process. In aid of this task, an initial approach was taken in identifying the content categories in a broadcast context and identifying the onsets of those identified content categories in a broadcast stream.

The proposed approach focused on classifying the broadcast content into news, advertisements/ radio commercials, songs and other human voice content which includes radio discussions, radio dramas, phone interviews etc. In the classification model, the main target was to identify a set of features and design a good network structure which helps in classifying the above-identified content categories with promising accuracies. The proposed classification model was successful in identifying news, advertisements/radio commercials, songs and other voice content with accuracies of 41%, 76%, 75% and 59% respectively. According to the classification results which were discussed in Chapter 5.5.4, the proposed classifier is successful in identifying the song frames over news frames. The dynamic nature of the Sri Lankan radio broadcast context affected for the misclassification rates of songs with advertisements/ radio commercials and other voice content. A structure analysis methodology for the above-identified content categories would improve the classification results by reducing the misclassification rates.

The proposed onset detection methodology is solely based on the above classification results. Onsets were characterized by the positions where the classification results experience

a content change. The proposed model was successful in detecting the onsets of (+/-) 2.5s of error rate with an accuracy of 82%. More improvements to the classification model will help in reducing the false positive rate for onset detection of the model.

6.3 Conclusions about Research Problem

Monitoring the radio broadcast context plays an important role in every country's mass media and broadcast acts. So there exists a huge requirement in the world for such reliable monitoring applications. For such applications, it's a requirement to identify the different content categories that are being broadcast in the public radio broadcast context. As discussed in Chapter 1 and Chapter 2 it is clear that there is no unified methodology for the radio broadcast content classification and onset detection in radio broadcast context. This research focuses on identifying the content categories in public radio broadcast context and identifying the onsets of each content category. Onset detection is not basically addressed in the radio broadcast domain. The proposed approach contributes in identifying a methodology for the onset detection in the broadcast context and this research contributes with a model which can address the classification of radio broadcast content into pre-identified content categories and finding the onsets in them.

6.4 Limitations

This research comprises a model which helps in the classification process of the radio broadcast context into pre-identified content categories and a methodology to detect the onsets in a broadcast content. The major limitation in this model is that it only trained for the selected broadcast channel (i.e. SLBC- Commercial Service'). The classifier should be trained for the other broadcast channels in order to make the model so generalized to other content categories. Since the onset detection is completely based on the classification results, every misclassification in the classifier output will lead to the false results of the onset detection. The used refinement rules are based on the observations of the selected radio channel. The network is being trained in each and every iteration. Therefore some accuracy levels might change from iteration to iteration.

6.5 Implecations for Further Research

The main component of the proposed model is the classifier. The classifier output has a huge weight in the accuracies of the detected content categories as well as onsets.More improvements to the classifier should be done in future work. For an instance, a classifier like a CNN can be used for the classification process. More experiments should be done in

the identification of more features which can give promising results for the classification of broadcast content categories. A more comprehensive training set which can detect the content categories in many broadcast channels should be developed in the generalization of this approach. Well-mannered annotation procedure should be followed in making the training dataset. A knowledge base can be built on top of the classifier in advancing the accuracy results. And also the experiments to identify the structural similarities of the content categories can be performed in improving the classification results. Some metadata like time can be used in the refinement process of the classification results and some more semantic rules should be identified when generalizing this approach to other broadcast channels.

References

[1] C. R. S. Celebrating Radio: Statistics / World Radio Day 2015, 2018. [Online]. Available: http://www.diamundialradio.org/2015/en/content/celebrating-radiostatistics.html

[2] J. Schluter and S. Bock, "Improved musical onset detection with convolutional neural networks," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[3] Q. NDegara, A. Pena Giménez, M. Sobreira Seoane, and S. Torres Guijarro, *Knowledge-based Onset Detection in Musical Applications*, 2008.

[4] [Online]. Available: https://en.wikipedia.org/wiki/Onset_(audio)

[5] J. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. Sandler, "A tutorial on onset detection in music signals," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 1035–1047, 2005.

[6] O. Mubarak, E. Ambikairajah, and J. Epps, "Analysis of an mfcc-based audio indexing system for efficient coding of multimedia sources," *Proceedings of the Eighth International Symposium on Signal Processing and Its Applications, 2005.*

[7] X. Rodet and F. Jaillet, "Detection and modeling of fast attack transients," 2018. [Online]. Available: http://hdl.handle.net/2027/spo.bbp2372.2001.105

[8] C. Duxbury, J. P. Bello, M. Davies, and M. Sandler, "A combined phase and amplitude based approach to onset detection for audio segmentation," in *Digital Media Processing For Multimedia Interactive Services*. World Scientific, 2003, pp. 275–280.

[9] C. Duxbury, J. P. Bello, M. Davies, M. Sandler *et al.*, "Complex domain onset detection for musical signals," in *Proc. Digital Audio Effects Workshop (DAFx)*, vol. 1. Queen Mary University London, 2003, pp. 6–9.

[10] [Online]. Available: http://digitalradiotracker.com

[11] [Online]. Available: https://www.acrcloud.com

[12] [Online]. Available: http://www.beatgridmedia.com

[13] R. Kotsakis, G. Kalliris, and C. Dimoulas, "Investigation of broadcast-audio semantic analysis scenarios employing radio-programme-adaptive pattern classification," *Speech Communication*, vol. 54, no. 6, pp. 743–762, 2012.

[14] J. P. Bello and M. Sandler, "Phase-based note onset detection for music signals," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 5. IEEE, 2003, pp. V–441.

[15] A. Röbel, "Onset detection in polyphonic signals by means of transient peak classification," in *MIREX Online Proceedings (ISMIR 2005)*, 2005, pp. 1–1.

[16] R. Zhou and J. D. Reiss, "Music onset detection combining energy-based and pitch-based approaches," *Proc. MIREX Audio Onset Detection Contest*, 2007.

[17] P. Brossier, J. P. Bello, and M. D. Plumbley, "Real-time temporal segmentation of note objects in music signals," in *Proceedings of ICMC 2004, the 30th Annual International Computer Music Conference*, 2004.

[18] Y. Wan, X. Wang, R. Zhou, and Y. Yan, "Multi-pitch onset detection via temporal segmentation and segmental analysis."

[19] S. Böck, A. Arzt, F. Krebs, and M. Schedl, "Online real-time onset detection with recurrent neural networks," in *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx-12), York, UK*, 2012.

[20] J. Saunders, "Real-time discrimination of broadcast speech/music," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 2. IEEE, 1996, pp. 993–996.

[21] S. G. Koolagudi, S. Sridhar, N. Elango, K. Kumar, and F. Afroz, "Advertisement detection in commercial radio channels," in *Industrial and Information Systems (ICIIS), 2015 IEEE 10th International Conference on*. IEEE, 2015, pp. 272–277.

[22] E. Senevirathna and K. Jayaratne, "Automated content based audio monitoring approach for radio broadcasting," 2013.

[23] G. Abayawickrama, V. Wijesuriya, and K. Jayaratne, "Web-based audio monitoring and analytics system for securing royalty payments of sri lankan song artists," Ph.D. dissertation, University of Colombo School of Computing, 2017.

[24] L. Lu, H.-J. Zhang, and H. Jiang, "Content analysis for audio classification and segmentation," *IEEE Transactions on speech and audio processing*, vol. 10, no. 7, pp. 504–516, 2002.

[25] Z. Kons and O. Toledo-Ronen, "Audio event classification using deep neural networks." in *Interspeech*, 2013, pp. 1482–1486.

[26] J. Foote, M. L. Cooper, and U. Nam, "Audio retrieval by rhythmic similarity." in *ISMIR*, 2002.

[27] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, vol. 1. IEEE, 2000, pp. 452–455.

Appendices

Appendix A

Code Listings

A.1 Audio feature extraction and feature vector construction

```
from __future__ import print_function
import librosa
import matplotlib.pyplot as plt
import numpy as np
import librosa.display
import math

filename = '<filename>';

y_old,sr = librosa.load(filename, mono=True, sr = None);
y = librosa.resample(y_old,sr,22050);
sr = 22050;
frame_duration = 2.5;
frame_length = int(math.floor(frame_duration*sr));
N = len(y);
num_frames = int(math.floor(N/frame_length));

chroma_feature_vec = librosa.feature.chroma_stft(y=y,
sr=sr, n_fft = frame_length,hop_length=frame_length);

mfcc_feature_vec = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=13,
n_fft = frame_length, hop_length = frame_length);

rms_feature_vec = librosa.feature.rmse(y=y,
```

```
n_fft = frame_length, hop_length =frame_length);

spectral_centroid_vec = librosa.feature.spectral_centroid(y=y,
sr=sr, n_fft =frame_length, hop_length= frame_length);

spectral_contrast_vec = librosa.feature.spectral_contrast(y=y,
sr=sr, n_fft =frame_length, hop_length = frame_length);

spectral_rolloff_vec = librosa.feature.spectral_rolloff(y=y,
sr=sr, n_fft = frame_length,hop_length = frame_length);

zcr_vec = librosa.feature.zero_crossing_rate(y=y,
frame_length= frame_length,hop_length= frame_length);

onset_strength_vec = librosa.onset.onset_strength(y=y,
sr = sr, n_fft = frame_length,hop_length= frame_length);

tempo = librosa.beat.tempo(y=y, sr = sr ,
hop_length = frame_length, aggregate=None);

frame_times = np.empty([num_frames,1]);

for k in range(0,num_frames):
        frame_start = int(k*frame_length);
        frame_end = int(frame_start+frame_length -1);

        frame = np.array(y[frame_start:frame_end+1])

        frame_start_time = np.true_divide(frame_start,sr);
        frame_end_time = frame_start_time +
                        np.true_divide(frame_length,sr);

        frame_start_time = float(format(round(frame_start_time,1)));
        frame_end_time = float(format(round(frame_end_time,1)));

        frame_times[k] = frame_start_time+(frame_end_time
                        -frame_start_time)/2;

#Constructing the feature vector
```

```
feature_vec = np.empty([num_frames,38]);

for k in range(0,num_frames):
        feature_vec[k,0] = rms_feature_vec[0,k];
        feature_vec[k,1] = spectral_centroid_vec[0,k];

    feature_vec[k,2] = spectral_contrast_vec[0,k];
        feature_vec[k,3] = spectral_contrast_vec[1,k];
        feature_vec[k,4] = spectral_contrast_vec[2,k];
        feature_vec[k,5] = spectral_contrast_vec[3,k];
        feature_vec[k,6] = spectral_contrast_vec[4,k];
        feature_vec[k,7] = spectral_contrast_vec[5,k];
        feature_vec[k,8] = spectral_contrast_vec[6,k];

        feature_vec[k,9] = spectral_rolloff_vec[0,k];
        feature_vec[k,10] = zcr_vec[0,k];

        feature_vec[k,11] = mfcc_feature_vec[0,k];
        feature_vec[k,12] = mfcc_feature_vec[1,k];
        feature_vec[k,13] = mfcc_feature_vec[2,k];
        feature_vec[k,14] = mfcc_feature_vec[3,k];
        feature_vec[k,15] = mfcc_feature_vec[4,k];
        feature_vec[k,16] = mfcc_feature_vec[5,k];
        feature_vec[k,17] = mfcc_feature_vec[6,k];
        feature_vec[k,18] = mfcc_feature_vec[7,k];
        feature_vec[k,19] = mfcc_feature_vec[8,k];
        feature_vec[k,20] = mfcc_feature_vec[9,k];
        feature_vec[k,21] = mfcc_feature_vec[10,k];
        feature_vec[k,22] = mfcc_feature_vec[11,k];
        feature_vec[k,23] = mfcc_feature_vec[12,k];

        feature_vec[k,24] = tempo[k];

        feature_vec[k,25] = chroma_feature_vec[0,k];
        feature_vec[k,26] = chroma_feature_vec[1,k];
        feature_vec[k,27] = chroma_feature_vec[2,k];
        feature_vec[k,28] = chroma_feature_vec[3,k];
        feature_vec[k,29] = chroma_feature_vec[4,k];
        feature_vec[k,30] = chroma_feature_vec[5,k];
```

```
feature_vec[k,31] = chroma_feature_vec[6,k];
feature_vec[k,32] = chroma_feature_vec[7,k];
feature_vec[k,33] = chroma_feature_vec[8,k];
feature_vec[k,34] = chroma_feature_vec[9,k];
feature_vec[k,35] = chroma_feature_vec[10,k];
feature_vec[k,36] = chroma_feature_vec[11,k];
feature_vec[k,37] = onset_strength_vec[k];
```

A.2 Neural Network structure for the classification model

```
#replacing labels with integers
Y_train[Y_train=='Advert'] = 0;
Y_train[Y_train=='News'] = 1;
Y_train[Y_train=='Song'] = 2;
Y_train[Y_train=='Voice'] = 3;

# Converting the labels to categorical labels (hot labels);
Z_train= keras.utils.to_categorical(Y_train, num_classes=4);

# create model
model = Sequential();
model.add(Dense(200, input_dim=38, activation='relu'));
model.add(Dropout(0.25));
model.add(Dense(200, activation='relu'));
model.add(Dropout(0.25));
model.add(Dense(200, activation='relu'));
model.add(Dropout(0.25));
model.add(Dense(200, activation='relu'));
model.add(Dropout(0.25));
model.add(Dense(200, activation='relu'));
model.add(Dropout(0.25));
model.add(Dense(4, activation='softmax'));

# Compile model
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer='rmsprop',
metrics=['accuracy']);

# Fit the model
```

```
try:
        model.fit(X_train, Z_train, epochs=300, batch_size=200);
except ValueError as err:
        print ('error',err);

# evaluate the model
scores = model.evaluate(X_train, Z_train);
print("\n%s: %.2f%%" % (model.metrics_names[1], scores[1]*100));

# prediction
res= model.predict(X_test);
results=np.matrix.round(res);

categorical_labels_pred = [];
pred_labels=[];

for i in results:
        categorical_label_value =np.argmax(i,-1)
        categorical_labels_pred.append (categorical_label_value);

        if(categorical_label_value == 0):
                pred_labels.append([categorical_label_value,'Advert']);
        elif(categorical_label_value == 1):
                pred_labels.append([categorical_label_value,'News']);
        elif(categorical_label_value == 2):
                pred_labels.append([categorical_label_value,'Song']);
        elif(categorical_label_value == 3):
                pred_labels.append([categorical_label_value,'Voice']);

np.savetxt("test_results.csv", pred_labels, delimiter=",",fmt='%s');
```

A.3 Refinement process for 'News' frames

```
#Refinement
refinement_songs_advert_between_news = [];

for k in range(0,len(pred_labels)-1):
        cur_val = pred_labels[k][1];
        if(k==0):
```

```
                refinement_songs_advert_between_news.append(cur_val);
        else:
                prev_val = pred_labels[k-1][1];
                next_val = pred_labels[k+1][1];
                if(prev_val == 'News' and next_val == 'News' and
                cur_val =='Song'):
                        refinement_songs_advert_between_news.
                        append('News');
                elif(prev_val == 'News' and next_val == 'News'
                        and cur_val == 'Advert'):
                        refinement_songs_advert_between_news.
                        append('News');
                else:
                        refinement_songs_advert_between_news.
                        append(cur_val);

refinement_songs_advert_between_news.append(
                        pred_labels[len(pred_labels)][1]);
```

A.4 Refinement process for 'Song' frames

```
#Refinement for songs
def count_song_frames(rel_array):
        song_frame_indexes=[];
        for k in range(0, len(rel_array)):
                cur_val = rel_array[k];
                if(cur_val == 'Song'):
                        song_frame_indexes.append(k);
        return song_frame_indexes;

song_frame_indexes_count_1 =
count_song_frames(refinement_songs_advert_between_news);

refinement_songs_1 = refinement_songs_advert_between_news[:];
for k in range(0,len(song_frame_indexes_count_1)-1):
        cur_val_index = song_frame_indexes_count_1[k];
        next_val_index = song_frame_indexes_count_1[k+1];
        value_diff = next_val_index - cur_val_index;
        if(value_diff <= 5 and value_diff>1):
```

```
                for i in range (0,value_diff):
                        refinement_songs_1[cur_val_index+i+1]= 'Song';

song_frame_indexes_count_2 = count_song_frames(refinement_songs_1);

refinement_songs_2 = refinement_songs_1[:];
song_frame_counter=1;
count_start_ptr = 0;
for k in range(0, len(song_frame_indexes_count_2)-1):
        cur_val_index = song_frame_indexes_count_2[k];
        next_val_index = song_frame_indexes_count_2[k+1];
        value_diff = next_val_index - cur_val_index;
        if(value_diff == 1):
                song_frame_counter = song_frame_counter+1;
        else:
                if(song_frame_counter<16):
                        for i in range(count_start_ptr,k+1):
                                index = song_frame_indexes_count_2[i];
                                before_ref_val =
                refinement_songs_advert_between_news[index];
                                if(before_ref_val == 'Song'):
                                        refinement_songs_2[index] =
                                        'Advert';
                                else:
                                        refinement_songs_2[index] =
                                        before_ref_val;
                                        count_start_ptr = k+1;
                                        song_frame_counter = 0;

#Refinement at the end of the song_frame_indexes_count_2 list
if(song_frame_counter<16):
        for i in range(count_start_ptr,k+1):
                index = song_frame_indexes_count_2[i];
                before_ref_val =
                        refinement_songs_advert_between_news[index];
                if(before_ref_val == 'Song'):
                        refinement_songs_2[index] = 'Advert';
                else:
                        refinement_songs_2[index] = before_ref_val;
```

```
    count_start_ptr = k+1;
        song_frame_counter = 0;

last_val_index = song_frame_indexes_count_2[-1];
before_ref_val = refinement_songs_advert_between_news[last_val_index];
if(before_ref_val == 'Song'):
        refinement_songs_2[last_val_index] = 'Advert';
else:
        refinement_songs_2[last_val_index] = before_ref_val;
```

A.5 Onset labeling and one-to-one onset detection

```
#Label the predicted onsets as onsets and Non_onsets
final_results = refinement_songs_2[:];
predicted_onsets_and_nononsets = [];
predicted_onsets_and_nononsets.append('Non_onset');
for k in range (1, len(final_results)):
        cur_val = final_results[k];
        prev_val = final_results[k-1];
        if(cur_val!= prev_val):
                predicted_onsets_and_nononsets.append('Onset');
        else:
                predicted_onsets_and_nononsets.append('Non_onset');

#Label the actual onsets as onsets and Non_onsets
actual_onsets_and_nononsets = [];
actual_onsets_and_nononsets.append('Non_onset');
for k in range (1, len(Y_test)):
        cur_val = Y_test[k];
        prev_val = Y_test[k-1];
        if(cur_val!= prev_val):
                actual_onsets_and_nononsets.append('Onset');
        else:
                actual_onsets_and_nononsets.append('Non_onset');

#Onset mapping -one to one
TP_indexes = [];
FP_indexes = [];
```

```
FN_indexes = [];
TN_indexes = [];

for k in range(0,len(actual_onsets_and_nononsets)):
        if(actual_onsets_and_nononsets[k]=='Onset' and predicted_
                onsets_and_nononsets[k]=='Onset'):
                TP_indexes.append(k);
        elif (actual_onsets_and_nononsets[k]=='Onset' and predicted_
                onsets_and_nononsets[k]=='Non_onset'):
                FN_indexes.append(k);
        elif (actual_onsets_and_nononsets[k]=='Non_onset' and predicted_
                onsets_and_nononsets[k]=='Onset'):
                FP_indexes.append(k);
        elif (actual_onsets_and_nononsets[k]=='Non_onset' and predicted_
                onsets_and_nononsets[k]=='Non_onset'):
                TN_indexes.append(k);
```

A.6 Onset detection with (+/-) 2.5s error rate

```
#Onset mapping - onsets at either sides of the predicted list
approximated_onsets = TP_indexes[:];
selected_actual_onsets = TP_indexes[:];
for k in range(1, len(predicted_onsets_and_nononsets)-1):
        if(k not in approximated_onsets):
                cur_pred_val = predicted_onsets_and_nononsets[k];
                cur_act_val = actual_onsets_and_nononsets[k];
                prev_act_val = actual_onsets_and_nononsets[k-1];
                next_act_val = actual_onsets_and_nononsets[k+1];

        if(cur_pred_val == 'Onset' and cur_act_val=='Onset'):
                if(k not in selected_actual_onsets):
                        approximated_onsets.append(k);
                        selected_actual_onsets.append(k);
        elif(cur_pred_val == 'Onset' and prev_act_val == 'Onset'):
                if(k-1 not in selected_actual_onsets):
                        approximated_onsets.append(k);
                        selected_actual_onsets.append(k-1);
        elif(cur_pred_val == 'Onset' and next_act_val == 'Onset'):
                if(k+1 not in selected_actual_onsets):
```

```
approximated_onsets.append(k);
selected_actual_onsets.append(k+1);
```