# Classification of Voice Content in Public Radio Broadcasting Context

G.A.G.S.Karunarathna

# Classification of Voice Content in Public Radio Broadcasting Context

G.A.G.S.Karunarathna
Index No : 14000601

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 (SCS4124)

University of Colombo School of Computing

Sri Lanka

January, 2019

# 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 : G.A.G.S.Karunarathna

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

This is to certify that this dissertation is based on the work of Ms. G.A.G.S.Karunarathna 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

Mass media has acquired a global character because of the rapid development of information technology. This technological advancement undoubtedly impacts the traditional mass media such as newspapers, television broadcasting, and radio broadcasting where the important changes have occurred in its production and distribution chains. With the evolution of mass media technology, content analysis of radio broadcasting emerged as a major research area which facilitates to automate the radio broadcasting monitoring process.

This dissertation focuses on the problem of automating the radio broadcasting monitoring process in Sri Lanka. A proper content classification is required to monitor radio broadcasting content automatically. In this research, more attention goes to the voice dominant content classification of radio broadcasting by employing a multi-class Support Vector Machine(SVM). Multi-class SVM implements as a compound of binary SVM classifiers. This study comparably investigates the performance of "One Vs. One" and "One Vs. All" methods which are known as two conventional ways to build multi-class SVM.

One of the most substantial measures in creating such classification is selecting the optimal feature sets for each binary SVM classifier independently. For that, time domain features, frequency domain features, cepstral features, and chroma features are manually analyzed. The two multi-class SVM models are trained based on the selected features. These models are capable of classifying five voice dominant classes such as news, conversations, advertisements without jingles, radio drama and religious programs with accuracies of $85\%$ and $83\%$ respectively for "One Vs. One" and "One Vs. All" models. Therefore "One Vs. One" model is selected as the soundest multi-class SVM classifier for this study.

# Preface

Audio pattern analysis and signal processing are used for extract information from radio broadcasting context in order to generate a new knowledge. Voice content analysis and processing in broadcasting context is a novel research study in Sri Lankan radio broadcasting context in order to automate the radio broadcasting monitoring process. There are several machine learning approaches to classify multiple audio classes such as Neural Networks, Deep learning models, Hidden Markov Model, Gaussian Mixture Model etc. Most of them are using the same set of features to classify multi-classes. This dissertation explores a Support Vector Machine (SVM) methodology which uses unique set of features for each class to discriminate that class instead of using all features.

‘Sri Lanka Broadcasting Corporation’ (SLBC) audio recordings are used as the primary data source. Annotating data for the training and testing was made by me. Then feature patterns are observed manually by myself and extracted the features which show a good discriminating pattern for each class. In Chapter 3, the design purely relies on the observed feature patterns of each audio class. In the classification phase, multi-class SVM is implemented using two approaches to select the most reliable approach which fits into the problem domain. The implementation details exposed in Chapter 4 are entirely my own work. The results in Chapter 5 are exclusively depends on the experiments carried out by me with the guidance of 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 allowing me to undertake this work and providing me with continuous guidance, supervision and inevitable suggestion throughout the research.

My utmost gratitude goes to Dr. M.I.E.Wickramasinghe, senior lecturer of University of Colombo School of Computing and Mr. W.V.Welgama, senior lecturer of University of Colombo School of Computing for providing feedback to improve my research study throughout the proposal defense and the interim evaluations. With a special mention to Dr. H.E.M.H.B.Ekanayake as the final year computer science project coordinator.

This thesis is dedicated to my loving family who has provided me through moral support in my life. I am also grateful to my colleagues who have supported me along the entire university 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 |
| APE | Monkey’s Audio |
| DAB | Digital Audio Broadcasting |
| DAGSVM | Directed Acyclic Graph Support Vector Machine |
| DNN | Deep Neural Network |
| DRT | Digital Radio Tracker |
| DWT | Discrete Wavelet Transform |
| FFT | Fast Fourier Transformation |
| FM | Frequency Modulation |
| GMM | Gaussian Mixture Model |
| HMM | Hidden Markov Model |
| KNN | K Nearest Neighbour |
| LPC | Linear Predictor Coefficients |
| MFCC | Mel-Frequency Cepstral Coefficient |
| MFE | Mel Filter bank Energies |
| RBF | Radial Basis Function |
| RBM | Restricted Boltzmann Machine |
| RMS | Root Mean Square |
| SLBC | Sri Lanka Broadcasting Cooperation |
| SMO | Sequential Minimization Optimization |
| SVM | Support Vector Machine |
| WMA | Windows Media Audio |
| ZCR | Zero Crossing Rate |

# Chapter 1

# Introduction

## 1.1 Background

People use radio as a communication medium over many decades. After its invention, the radio still reaches its broad stage as a dynamic and amiable communication device. The difference between radio and other information sources such as televisions, computers, and smartphones is people listen to the radio without any technological advancement. The only prerequisite to using a radio is not to have hearing disabilities. According to the statistic found in 2015, more than half of the population has car radios [1]. Most of the women and men are tend to listen to the radio at work. The youth seems to be moving from traditional radio to online radio. Therefore, unlike the rest of the communication mediums radio plays an important role in the society.

In a radio transmission, radio station and the listeners are the two endpoints. Radio stations broadcast unidirectional wireless signals over space to the multitudes of individual listeners equipped with radio receivers. The broadcasting content consists of a sequence of various kind of content categories such as songs, advertisements, news, interviews, conversations, and radio dramas. The audience of a radio channel always relies on the content category that the radio stations are broadcasting (i.e. musical programs may have many more listeners than a political conversation). Therefore, for a particular radio station, the number of listeners can be varied on a program to program. Thus, in order to grasp the audience to a program, the program should be performed well, and fit into the audience. Hence, for the purpose of measuring the performance of a broadcast program, radio stations need to monitor the broadcasting content regularly. Broadcast content monitoring helps to verify when and where the broadcast content placed, protect copyrights by knowing precisely how the content is being used, and measure performance across other broadcast channels. Therefore, broadcast content monitoring is a necessary thing for radio stations.

Furthermore, the stakeholders of radio channels also need to monitor the broadcast content for different purposes such as legal assessments, and economic aspects. Authorized people in mass media and information corporations need to track the FM channels regularly to ensure whether the broadcasting contents are in order to the rules and regulations and has a diversity of available programs. Singers and composers need to monitoring songs to make sure that loyalty of the payments, and the copyrights. Advertising agents are keen on the frequencies of the broadcast advertisements which makes a huge impact on the company revenue. Several political parties also keep an alert on their name referencing in the radio broadcasting content, especially on the news and political discussions. Similarly, most of the stakeholders monitor the radio broadcasting content for analyzing the content.

In the monitoring of radio broadcasting, both manual monitoring and automated monitoring are used. A human being listens to the radio and track down the relevant information is known as the manual monitoring. In automated radio monitoring processes, well-trained machine monitors the radio content behalf of a man. It saves the time, money and manpower. Most of the time, well-developed countries use automated radio monitoring process. As a developing country, Sri Lanka has not yet established such a technology to monitor radio broadcasts. Since there is a large number of radio channels in Sri Lanka, manual monitoring is not practical. Unfortunately, the mechanisms used in developed countries can not be substituted for FM channels in Sri Lanka, because of the language differences, different pronunciations, and accent. Hence, having an automated radio broadcasting monitoring process will add a new fortune to the Sri Lankan radio broadcasting context.

As an initial step to build a automated monitoring process, identifying different content classes (i.e. songs, advertisements, news, interviews, conversations, and radio dramas) in radio broadcasting content is essential. When analyzing the current situation of the above mentioned problem, classifying broadcast context for onset detection is recognized as the closest research work [2]. Onset detection is the mechanism which is used to identify the places where the content changes are happening in a musical note or other sound streams. This research proposed a unified methodology to automate radio broadcasting monitoring which detects onsets of radio broadcasting context with the assist of the classification of the broadcasting content. The research distinguish songs, commercial advertisements with jingles, news, and other contents in a radio stream. However, the issue with this unified method is, it is unable to identify voice dominant content classes in broadcasting context. Hence, classification of different voice dominant content classes has identified as a further enhancement of the research.

## 1.2 Research Problem and Research Question

As discussed in section 1.1 there is an essential requirement of automating the radio broadcasting monitoring process in Sri Lankan radio broadcasting context. Even though there are few commercially available software tools exist for radio monitoring such as Digital Radio Tracker (DRT), ACRCloud broadcast monitoring, BeatGrid etc., they are not working properly for Sri Lankan broadcasting context because of the language differences, different accents, and pronunciations [2]. Therefore, having an observer to listen to the radio content, reading the attached meta-data, asking for the broadcast report from broadcast stations are some of the poor mechanisms used currently. But they are less reliable. Also, consume time and money as well. In order to overcome this problem, automating the radio broadcasting monitoring process is a necessity.

The existing automated technique with onset detection identifies news, radio commercials with jingles, songs and other voice content with accuracies of $41\%$, $76\%$, $75\%$, and $59\%$ respectively [2]. However, voice dominant contents such as advertisements without jingles, radio dramas, conversations, and religious programs are not explicitly identified. Although the news is being identified, the accuracy of the classifying news should be improved. Consequently, the classification of different voice dominant contents in the radio broadcasting stream is identified as the knowledge gap in between the requirement and existing solutions. Hence, classifying these voice dominant contents are identified as the research problem which is going to address throughout this research. Based on the research problem, the research questions are as follows,

**Question 1: Is it possible to come up with an automated method to classify voice dominant contents in a radio broadcasting context into a set of pre-defined content categories?**

Altogether there are five pre-defined voice categories. All are news, advertisements without jingles, radio dramas, conversations, and religious programs. Since there are five pre-defined classes, a supervised learning approach is suggested. The two most important aspects of this research are to identify the unique features for each class to classify classes and to explore the most appropriate supervised learning approach to this problem domain. Since each class has its unique set of features, a binary classifier seems to be the most applicable choice to distinguish each of them individually from the rest of classes. In the machine learning context, SVM is the best choice for binary classification. Moreover, as described in Section 2.3.4, SVM performs comparatively better in audio classification problems [3-7].

**Question 2: How do multi-class Support Vector Machine and its variations affect the results of the classification?**

As mentioned in the conclusion of Chapter 2, multi-class SVM is been chosen as the classification model. There are three variations in multi-class SVM as One-Vs-One, One-Vs-All, and DAGSVM (Directed Acyclic Graph SVM). Since the DAGSVM is a hierarchical graph, there is a high probability to propagate the misclassifications of upper levels to lower levels. This might be lead to an error training situation while training the model. Hence DAGSVM is rejected at the very first step. Therefore, this research attempts to evaluate the most accurate method to this domain by comparing the performances of One-Vs-One and One-Vs-All models.

## 1.3 Aim and Objectives

The ultimate goal of the research is to identify the voice dominant content categories for automating the context of radio broadcasting in Sri Lanka. The objectives that need to be fulfilled to reach the aim are as follows,

**Objective 1: Identify the most appropriate features to build a classification model**

The human brain can understand the distinction between the different voice prominent audio classes without its meaning. That implies there are sensible discriminating features among these audio classes. Since the accuracy of the classification model tightly coupled with the selected features, a proper feature selection phase should be required for each class individually.

Hence, identifying proper features would be done in manually by analyzing the feature patterns of the frequency spectrum of each audio class. Moreover, the knowledge come through related works also would be helpful when selecting the features.

**Objective 2: Identify an appropriate learning approach to classify voice content in an efficient and effective way**

As stated in the literature, machine learning based approaches certainly benefited for most of the real world audio classification problems. Machine learning is a vastly expanded area including more learning approaches such as neural networks, deep learning, HMM (Hidden Markov Model), SVM etc. Hence, before employing an approach, analyze the problem domain properly is a must thing. As explained in Section 2.4, an appropriate learning method is decided by considering the applicability of each approach.

## 1.4 Justification of the Research

When analyzing the background of the audio content classification, a considerable amount of works were found related to radio broadcasting context. Most of them have done by the machine learning community. Real-time discriminating on broadcast speech/music [3], identify speaker role in broadcasting content [4, 5], identifying radio commercials [6], and identifying environmental sounds in broadcasting news [7] are some of the existing evidence for audio signal processing and classification in radio broadcasting. Even though such mechanisms exist, they do not support to Sri Lankan broadcasting context. As stated in Section 1.2, even though there is a unified methodology to address this problem, it is not capable of voice dominant content identification. Thus, distinguish different voice contents are identified as the knowledge gap between the requirement and the existing knowledge. Accordingly, this research mainly focuses on overcoming the identified knowledge gap with the assistant of a proper machine learning approach.

Automating manual supervision of FM channel will be added a novel value to the public radio broadcasting context in Sri Lanka. Moreover, this voice content classification strategy can be easily adapted to a broader range of audio events detection problems and different industry level applications as well. Hence, this will make this research significant.

## 1.5 Methodology

The methodology consists of four main stages as feature analyzing, pre-processing, classification, and evaluation. Analyze the dataset is the first step. This research specifically focuses on the analysis of time series and frequency series of audio signals. A quantitative interpretation of audio data is required for the analysis to identify the most suitable features for distinguishing each class separately. In the preprocessing phase, remove silence from relevant data, split audio clips into frames, and annotating data are carried out.

Since the dataset consists of a set of pre-defined classes (i.e. news, advertisements without jingles, radio dramas, conversations, and religious programs), a supervised learning approach is been proposed initially for the classification. The dataset is trained according to the selected classification approach. As the final step, evaluating the trained model under different criteria is considered in this research. In the evaluation process, manually annotated ground truth data are taken to evaluate the proposed approach.

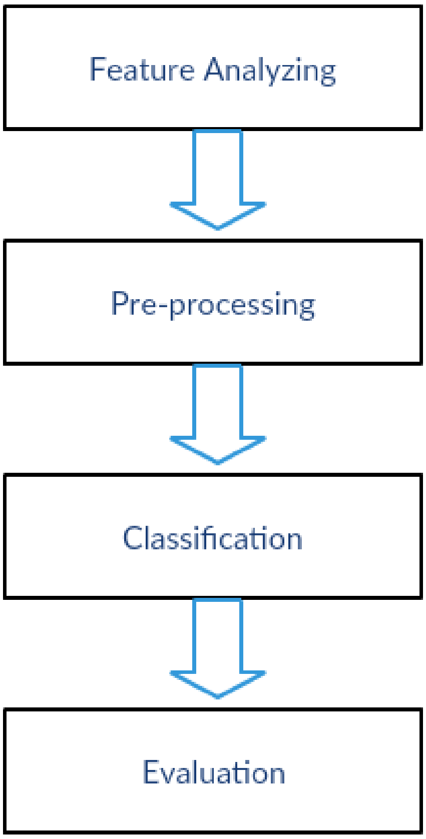


Figure 1.1: Proposed Methodology

## 1.6 Outline of the Dissertation

This dissertation is organized into six chapters as follows. This chapter is given a precise introduction to the research. Chapter 2 explores previous works related to this research and compare and contrast the different approaches used previously. Chapter 3 demonstrates the design of the proposed methodology. Chapter 4 provides the implementation details including the dataset, tools, and formulas used in the research. Chapter 5 presents the evaluation results. The proposed methodology is concluded along with future works in Chapter 6.

## 1.7 Delimitation of Scope

Voice dominant audio content classification in public radio broadcasting context is the major concern of this research project. News, advertisements without jingles, radio drama, religious program, and conversations are the five voice classes that consider in the classification. Songs, music and other contents that do not belong to voice content are not considered during the work.

In some cases, we identify radio events such as telephone conversations and live interviews by listen to its meaning. Since a classification model unable to interpret the meaning of the audio, this kind of events are not considered separately. Therefore single speaker programs, multi-speaker programs, and telephone conversations are categorized under the same category as conversations. Furthermore, under the advertisement category, advertisements with music and jingles are not considered.

In order to represent all the Sinhala FM channels 'Sri Lanka Broadcasting Corporation' (SLBC) is chosen as the primary data source. The total number of data used for the research is, 5 hours and 45 minutes lengthened.

During the preprocessing, silence removal is done only over the news and conversations. The reason for the silence removal is explained in Chapter 3. Due to the time limitations, we do not focus on implementing any industry level application or any device based on the results of the classification.

## 1.8 Conclusion

Initially, this chapter has precisely introduced the background details of the research domain. As seen in the background, radio content classification can be taken as a major research area which aids the radio broadcasting monitoring process. However, there exists a knowledge gap to classify voice dominant content categories according to the Sri Lankan radio channels. Therefore, the research questions are built through the identified problem and presented with the aim and objectives. Subsequently, the novelty of the research was justified. The proposed methodology was briefly described along with data analyzing, pre-processing, classification and evaluation phases. Finally, the dissertation was outlined according to the following chapters, and the scope of the research was stated. Based on the introduction chapter, the dissertation can proceed with a detailed description of the research.

# Chapter 2

# Literature Review

## 2.1 Introduction

The research focuses on automating the manual radio broadcasting monitoring process in Sri Lanka. Since this research belongs to audio signal processing and classification, a considerable amount of related works can be examined. As seen in the related works, algorithmic approaches and machine learning approaches are the traditional approaches that are followed by the researchers for audio classification.

## 2.2 Algorithmic Approaches

Lie Lu, Stan Z. Li and Hong-Jiang Zhang [8] proposed an algorithm which is able to classify an audio stream into speech, music, environment sounds and silence. High zero-crossing rate ratio, low short-time energy ratio, and spectrum flux are used to perform a fast pre-classification of speech and non-speech. Silence detection is performed based on short-time energy and zero-crossing rate (ZCR) in a one-second window. Threshold values of band periodicity, spectrum flux, and noise frame ratio are obtained to discriminate music from the environment sounds. LSP distance analysis is used to apply refinements over the proposed algorithm. The result of this research has some misclassifications between music and environment sound due to the overlaps in the distribution of the features. The sound of a crowd such as the shouting/ cheering has misclassified as music because of the similarity between the vocal tract periodicity of the shouting/ cheering and songs.

An algorithm for discriminating speech from music on broadcast FM radio based on ZCR of the time domain waveform is proposed by John Saunders [3]. This technique emphasized, the characteristics of speech such as limited bandwidth, alternate voiced and unvoiced sections, energy contour between high and low levels are well capable of separating speech from music. The proposed algorithm is indirectly unified with the multivariate Gaussian classifier which uses the amplitude, pitch and periodicity estimates of the waveform for the detection process. The researcher has reported an average accuracy of $98\%$.

Whittaker et al [4] proposed an algorithmic approach for speaker's role identification in radio broadcasting context. This approach classified anchor, journalist and guest programmer by considering lexical features, features from the surrounding context and the duration features (word frequency). First, the structure of the news program is constructed according to the speaker role and then carried out the classification process. An algorithm called "BoosTexter" is taken as the boosting algorithm in the proposed approach. Whittaker et al acquired $80\%$ accuracy.

Though the algorithmic approaches show good result, when the number of classes in the classification is increasing it becomes more complex. Identifying the threshold values to discriminate each class is also difficult. In order to avoid these negative side on the algorithmic approaches, the machine learning approaches are used.

## 2.3 Machine Learning Approaches

Among the algorithmic approach and machine learning approach, the machine learning community has done numerous works under both supervised learning and unsupervised learning. The learning approaches associated with supervised learning are Artificial Neural Networks (ANN), Deep Neural Networks (DNN), Hidden Markov Model (HMM), and Support Vector Machine (SVM). Unsupervised learning approaches are K Nearest Neighborhood (KNN), and Gaussian Mixture Model (GMM). Since our problem domain refers the supervised learning approaches, more attention goes to ANN, DNN, HMM, and SVM.

### 2.3.1 Artificial Neural Network

The most recent and close work which has addressed the same problem is, "Classification of public radio broadcast context for onset detection" conducted by C. Weeratunga [2]. In this approach, the onset detection mechanism along with a classification model is proposed to predict four classes (i.e. songs, voice-related segments, news, and radio commercials). A supervised neural network model with 38 extracted features has included in the classification framework. Radio commercials, songs, news, and other voice contents are classified with accuracies of $76\%$, $75\%$, $41\%$, and $59\%$ respectively. The output of the classification provided a huge weight to the accuracy of the onset detection mechanism. Currently, it has $82\%$ accuracy for onset detection with respect to prior mentioned audio classes in radio broadcasting context. In order to automate the radio broadcasting monitoring process, the existing onset detection method should be improved. Therefore as a further step, we do focus on classifying voice dominant contents in radio broadcasting events.

Another supervised neural network approach has used by M. Kashif Khan et al [9] to classify speech and music. In this model, the percentage of ”Low-Energy” Frames, RMS of a Low-Pass response, spectral flux, mean and variance of the Discrete Wavelet Transform (DWT), difference of maximum and minimum Zero Crossings, Linear Predictor Coefficients (LPC) are selected as features. As the classification framework, multilayer perceptron neural network and backpropagation learning algorithm is used. The experimental results have shown the overall accuracy is $96.6\%$, with music has classified correctly $100\%$.

A bit different research work is done by R. Kotsakis, G. Kalliris, C. Dimoulas [10]. In this research, various audio pattern classifiers in the broadcast-audio semantic analysis are investigated using radio program-adaptive classification strategies with supervised ANN system. The dataset consists of nine different voice classes including two main speakers, phone calls, and five music classes. 35 features are extracted for the classification. Finally, the researchers has observed that the methodology is applicable only for the radio-programs with a single main speaker. Otherwise, it is misclassified with multiple voices of the male speakers since there are more similarities in their voices. In the evaluation, Kotsakis et al found ANN and KNN classifiers quite effective than tree complex and SMO methods.

### 2.3.2 Deep Neural Network

IBM research team [11] suggested a deep neural network as a solution for classifying audio events. Four pre-defined classes as a crowd of people, cars/ road noises, applause/ yelling/ cheering, various kinds of music recorded in outdoor are chosen as the audio classes. A total number of 192 features are applied to the process. First 64 features represent the mean of the MFCC over the entire segment, the second set of 64 features represent the standard deviation (STD) and the last set of 64 features calculated by first taking the STD of the log spectrum over the frame and then applying the Mel-spaced filter banks. The DNN classifier consists of a multi-layer feed-forward perceptron network with back propagation learning algorithm. Introduced a new scaling factor to the network in the reconstruction step of the RBM training is made this research strong. Though GMM has the worst performance, the overall performance of the DNN classifier achieved the best in most of the classes, except for the music class where the SVM performs better.

### 2.3.3 Hidden Markov Model

Same as Neural Networks and Deep Learning approaches, the Hidden Markov Model is also shown high performance in radio broadcasting content classification problems. HMM is taken to radio commercial classification in radio broadcasting by G.Koolagudi et al [6]. In this work, the researchers input MFCC features to the classification model. Ensemble system comprises with ANN along with HMM is used as the classification model. Unlike ANN, HMM works on a different logic in speech recognition. HMM trains with probabilistic values of state transitions and output the highest probabilistic value as the result. As the consequences, some situations where ANN failed (i.e. background music follows an advertisement), HMM performed well.

Another work related to HMM has conducted by Yang Liu [5], identify the roles of speakers in radio broadcasting news contents. Well-structured news content is used in this research which highlights the speaker role sequences. Accuracy of $80\%$ is obtained and they found the beginning and the end of the sentences in the voice of the speaker is a good cue for role identification.

### 2.3.4 Support Vector Machine

SVM basically designs for binary classification problems. As extensions, multi-class SVM obtains by compromising set of binary SVM classifiers. There are 3 main designs for multi-class SVM as One-Vs-All, One-Vs-One, and DAGSVM [12]. The main advantage of SVM when compared to other machine learning approaches is that SVM perform much better in many cases because it finds the best hyperplane/s that separates all data into different classes, no matter even the dataset is small [13]. Aurino et al [14] have proposed a One-class SVM based approach to detect anomaly events that are considered as abnormal sounds in the environment like a gunshot, screaming and broken glass. The proposed methodology consists of two stages. At the first stage, the researchers introduced a new mechanism called “Majority Voting and Rejection” to classify short time frames into predefined classes. At the second stage, aggregated the results of the first stage into longer time frames and reclassified.

In the work of Bouril, A. et al [15], 3000 phonocardiograms from 9 locations of the body of both adults and children were taken to identify normal and abnormal heart sounds using SVM. Here, 74 features of time and frequency domain were considered. The SVM model was utilized by a Gaussian Kernel where it allows three different classifications; -1 for the normal heartbeat, 0 for ambiguous sounds due to noise and 1 for abnormal heartbeat sound. In this research, a binary SVM is chosen to be effective in normal and abnormal classification.

Audio based event detection in office live environments using optimized MFCC features with SVM model has implemented by Kucukbay et al [16]. The complexity of this work is indoor environmental sounds generally located in the background with lower energy which makes the detection process difficult. Sixteen classes such as alert beeping, clear throat, keyboard and switch on/off sounds were classified. One-Vs-All multi-class SVM is used as the classifier. K-fold cross-validation has been used for testing purposes. Martin Morato et al [17] conducted a case study on feature sensitivity for audio event classification using One-Vs-All multi-class SVM. Same as the above [16] sixteen classes have been differentiated using MFCC features and MFE features. To do that 2.5s frame length was used with 1s overlaps. 44.1 kHz was considered as the sampling frequency. Wang, J.et al [18] have used a frame based multi-class SVM classifier to differentiate fifteen audio classes including male, female recognition. A frame based classifier segmented one audio file into several frame sizes and trained the classifier for each. Even though this method improves accuracy from $13.9\%$, the pre-processing and training time is very high.

The same team that prior mentioned in [8], Lie Lu, Stan Z. Li and Hong-Jiang Zhang have proposed a method called hierarchical binary support vector machine for employing an audio segmentation and classification [19]. Here the researchers furthermore considered five pre-defined classes as silence, music, background sound, pure speech and non-pure speech including speech over music and speech over the noise. Since the feature distribution of audio data is so complicated and different classes may have overlapping or interwoven areas, the researchers recognized different audio classes that cannot be linearly separated. Hence a kernel based SVM is used to classify audio contents. In the evaluation, it has shown the accuracy of the SVM based method is better than the method based on KNN and GMM. But the major disadvantage in this approach is misclassifications of upper levels can be propagated to the classifiers in lower level.

The rapid development of broadcasting FM channels demands the news content classification of broadcasting context. Vavrek, J. et al [7] proposed a same methodology as in [19]. The SVM is designed as a binary classification hierarchical tree to address the complexity of the multi-class classification problem. This hierarchical classification strategy is used a particular feature set for each SVM binary classifier. Therefore, the F-score feature selection algorithm is used to obtain features which able to select optimal features for each SVM. The drawback of this work is the error of upper levels of the tree were propagated to the bottom levels. To prevent from that, misclassifications of upper levels are not considered.

The work of Zhu, Y., Ming, Z. , Q. Huang [20] is classified six audio classes using clip based SVM method. Here, the researchers classified pure speech, music, silence, environmental sounds, speech with music, and speech with environmental sounds. The unique property of clip based SVM is, the mean and standard deviation in one audio clip is computed to get clip-based features. The key finding of this work is, the researchers found that the performance of SVM shows good results in similar cases than Decision Trees, KNN, and Neural Networks.

## 2.4 Conclusion

This chapter focus on investigating extensive details of related works including the performance and drawbacks of each approach. The potentials of these approaches vary from problem domain to domain. While reviewing the literature, the best approach that is compatible with our problem domain was observed as follows. Since the dataset consists of a set of pre-defined classes, a supervised learning approach is proposed for the classification. Therefore unsupervised classifiers such as KNN and GMM were eliminated. As mentioned before, a unique set of features for each class has identified. Hence, if all the features are input to the classification model together, it will reduce the accuracy of the model because of some irrelevant features input to the classification of some classes. Therefore, ANN model was rejected. In other hands, HMM was rejected in view of the fact that the sequence of the audio events appearing is not beneficial to our problem. Due to the concern of input different features for different classes, SVM is selected as the classification model.

Since SVM's are originally designed for binary classification, the multi-class SVM builds as a compound of binary SVM classifiers. As we already identified specific features for each class, we can input only the relevant features separately in the case of using a multi-class SVM model because it holds multiple binary SVM models. Accordingly, multi-class SVM is chosen as the most suitable classifier which fits into our problem domain. As it is a composition of several binary SVMs, multi-class SVM can be designed as one of the following methods [12],

- One Vs. One
- One Vs. All
- Dynamic Acyclic Graph SVM (DAGSVM)

One-Vs-All constructs N number of binary SVM models where it has N number of classes. Every single binary SVM is trained with all of the data in the one class with positive labels and rest with negative labels. The decision function which has the largest value is taken as the predicted class. One-Vs-One constructs $\frac{N(N-1)}{2}$ number of binary SVM models where each one is trained only for two classes and a class is predicted using the ”Max-winning” strategy. Same as One-Vs-One, DAGSVM also constructs$\frac{N(N-1)}{2}$ number of binary SVM models where each one is trained for two classes. These binary SVMs are structured as a top to bottom hierarchical tree where it has $(N-1)$ number of leave nodes. It starts at the root node, then a binary decision function is evaluated, and it moves to either left or right depending on the output value of the previous node.

Since the DAGSVM is a hierarchical graph, the misclassifications of upper levels can propagate to lower levels [21]. This will lead to an erroneous situation. Hence DAGSVM was rejected at the very first step. One-Vs-One and One-Vs-All both have benefits as well as limitations [12]. It depends on the application domain. Hence this research attempts to obtain the most reliable method by modeling the multi-class SVM in both ways.

# Chapter 3

# Design

## 3.1 Design Overview

This chapter elaborates the overview of the research design for the proposed methodology. Section 3.2 described how the data is structured. Figure 3.1 shows a high-level diagram of the design of the methodology. As shown in the diagram, the design has five main components as feature analyzing, data pre-processing, feature extraction, the design of the classification model and evaluation. Section 3.3 describes all the details of analyzing the audio signals and the features that are selected. Section 3.4 explains the data preprocessing component with all considerations and construction steps. The feature extraction process is outlined in Section 3.5. Section 3.6 illustrates the design of the classification models and section 3.7 briefly mentions the evaluation plan.

## 3.2 Dataset

'SLBC' audio recordings are the primary data source of this research. At the beginning of the research, 35 hours of audio recordings are annotated by listening to them manually. When observe the data, one limitation of the dataset is that in the religious programs (i.e. Paritta-Sutta Chants) 1 hour and 10 minutes of the same recording plays every time. Therefore, it is unable to find more data for religious programs. Hence, the other classes also restricted to 1 hour and 10 minutes lengthened total recordings to avoid the proportional bias of the dataset. This causes to restrict the whole dataset to 5 hours and 50 minutes. Initially, the whole dataset is divided as $60\%$ and $40\%$ for training and testing purposes respectively as depicted in Figure 3.2.

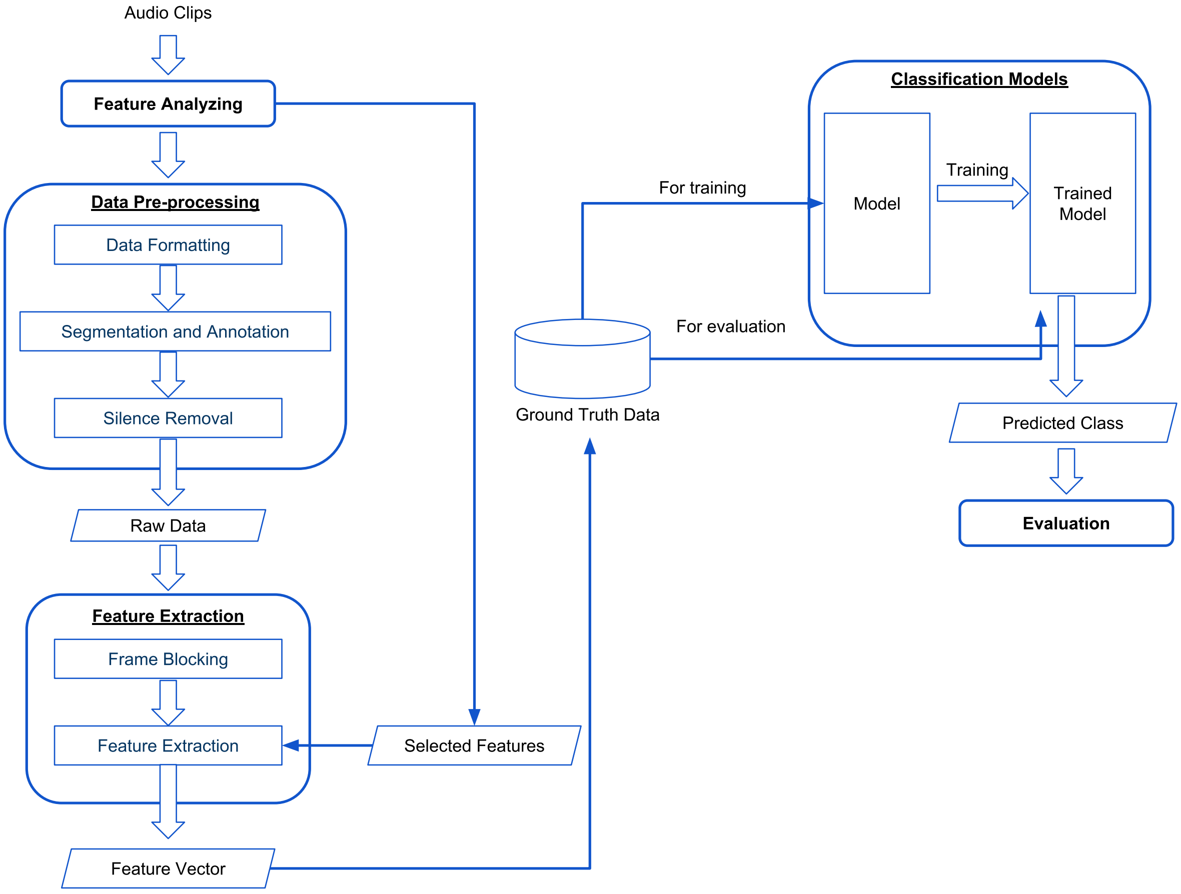


Figure 3.1: Design Overview

### 3.2.1 Training Dataset

According to the dividing factor, the training dataset is approximately 3 hours and 30 minutes lengthen. It consists of 2490 frames with 5s of frame sizes. As shown in Figure 3.2, training dataset is again divided into $70\%$ for train the classifiers and the rest $30\%$ for get the training accuracies of each classifier. Table 3.1 illustrates the arrangement of the training dataset.

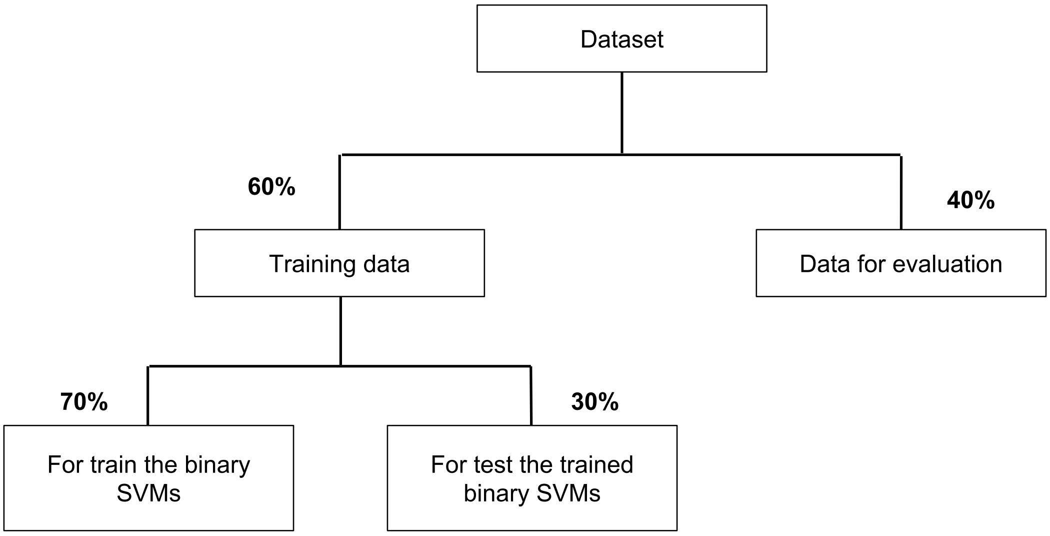


Figure 3.2: Dataset

Table 3.1: Arrangement of training dataset

| **Class** | **Number of frames** | **Percentage** |
|---|---|---|
| News | 495 | 20% |
| Conversations | 498 | 20% |
| Advertisements | 488 | 20% |
| Drama | 508 | 20% |
| Religious Programs | 501 | 20% |

### 3.2.2 Testing Dataset

According to the dividing factor, 2 hours and 20 minutes of a sharp dataset is belong to the testing dataset. The important fact is that this testing data is never being in the training dataset. This includes 1600 number of total frames with 5s of frame sizes. Table 3.2 illustrates the arrangement of the testing dataset.

Table 3.2: Arrangement of testing dataset

| Class | Number of frames | Percentage |
|---|---|---|
| News | 320 | 20% |
| Conversations | 325 | 20% |
| Advertisements | 316 | 20% |
| Drama | 316 | 20% |
| Religious Programs | 323 | 20% |

## 3.3 Feature Analyzing

Features are the only measurable unit of audio data. Features are used to capture the most useful information in the dataset. In a classification, identifying the most appropriate features is essential for differentiate one class from another. In order to select the appropriate features, the data should be analyzed well.

Since this research compare and contrast two multi-class SVM models, the feature selection should be carried out two times for each model. As illustrates in Table 3.3, the number of binary classifiers that are used for each multi-class SVM model as follows. For each binary classification, we must identify the characteristics according to the categories we classify.

Table 3.3: Binary classifiers of multi-class SVM models

| Multi-class model | Binary Classifiers | Identical classes |
|---|---|---|
| One-Vs-One | SVM 1 | News Vs. Advertisements |
| | SVM 2 | News Vs. Conversations |
| | SVM 3 | News Vs. Radio drama |
| | SVM 4 | News Vs. Religious program |
| | SVM 5 | Advertisements Vs. Conversations |
| | SVM 6 | Advertisements Vs. Radio drama |
| | SVM 7 | Advertisements Vs. Religious program |
| | SVM 8 | Conversations Vs. Radio drama |
| | SVM 9 | Conversations Vs. Religious program |
| | SVM 10 | Radio drama Vs. Religious program |
| One-Vs-All | SVM 1 | News Vs. others |
| | SVM 2 | Advertisements Vs. others |
| | SVM 3 | Conversations Vs. others |
| | SVM 4 | Radio drama Vs. others |
| | SVM 5 | Religious program Vs. others |

### 3.3.1 One-Vs-One: Feature Analyzing

According to Table 3.3, features for ten class pairs should be analyzed manually to select the best subset of features. For that, ten audio clips are prepared as Figure 3.3 with class pairs where one class is 15 minutes lengthened.

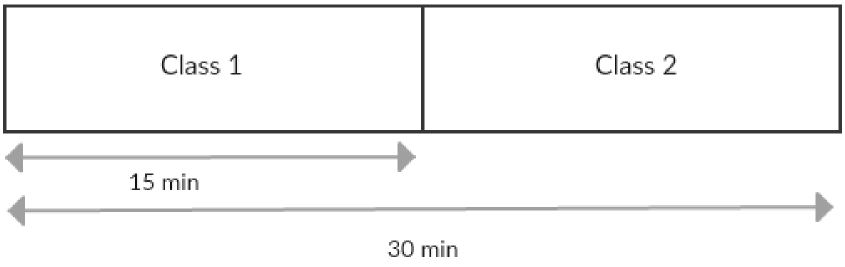


Figure 3.3: Audio clip structure designed for analyze features of two classes

Also, for each pair, a total of 34 features are observed. Figure 3.4 shows one of the obtained spectrum of energy feature which is selected to distinguish advertisements and religious programs.

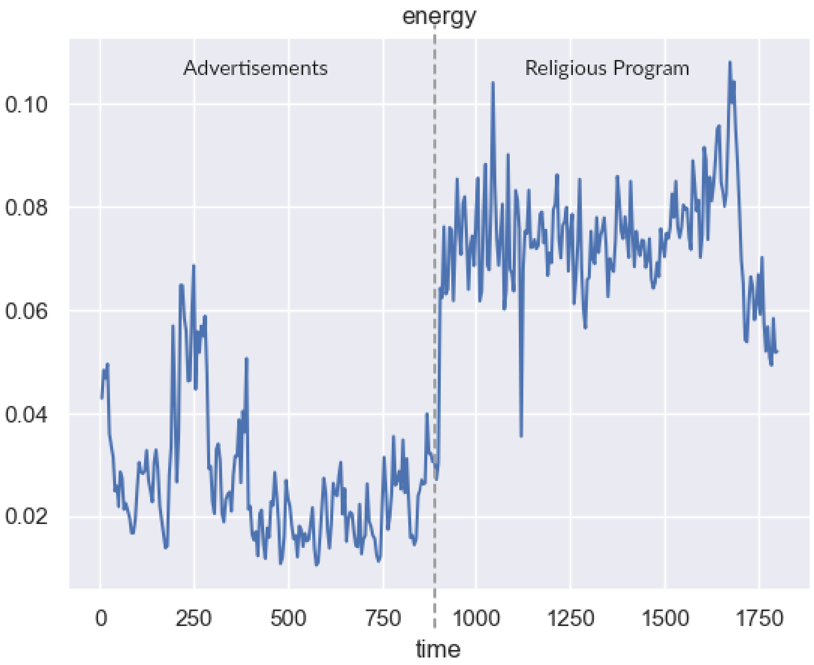


Figure 3.4: Energy Feature for distinguish advertisements and religious programs

Using these obtained spectrums, feature patterns are analyzed for each pair of classes and altogether 24 features are selected. The feature patterns for each selected feature are included in

Appendix B. The selected features are shown in Table 3.4 with its rank obtained by calculating the importance of the features. All the features are not necessarily applied for each pair (i.e. Energy feature shows a good discriminating pattern for religious programs. Therefore, instead of using energy feature for distinguishing all class pairs, it is used for all the class pairs which have religious programs). Likewise, the relevant features for each pair are selected. Removing unnecessary features causes to reduce the dimensions of the classification model.

### 3.3.2 One-Vs-All: Feature Analyzing

As shown in Table 3.3, One-Vs-All method needs only five binary classifiers because it constructs the number of binary SVMs equal to the number of classes. Each classifier allocates for the classification of one class. Here we analyze relevant features for distinguishing a class from the rest. For that, an audio clip is prepared as Figure 3.5 including all the classes where one class has 25 minutes.

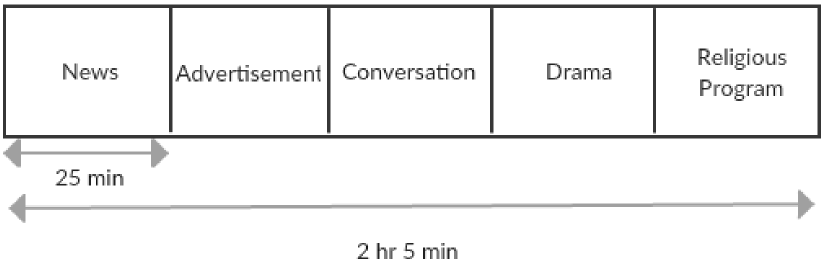


Figure 3.5: Audio clip structure designed for analyze features of one class from the rest

Using this 2 hours and 5 minutes lengthened sample audio clip, 34 feature patterns are analyzed. Figure 3.6 shows a pattern obtained from the frequency spectrum of Spectral Entropy against the five classes.

Table 3.4: Selected features for One-Vs-One model with ranks

| | **Features** | **SVM number** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| 1 | ZCR | | | | | | | | | | 7 |
| 2 | Energy | | | 7 | 3 | | | 4 | | 1 | 3 |
| 3 | Energy entropy | 1 | | | | | 3 | | | 5 | 4 |
| 4 | Spectral centroid | | 2 | 3 | 4 | 2 | 1 | 3 | | | 6 |
| 5 | Spectral spread | | | 4 | 1 | | 6 | 1 | | 2 | 1 |
| 6 | Spectral entropy | | 1 | | | 1 | 4 | | 1 | | |
| 7 | Spectral flux | | | | | 6 | | | | | |
| 8 | Spectral rolloff | | 4 | | | 4 | | | 3 | | |
| 9 | MFCC 1 | | | | | | | | | 7 | |
| 10 | MFCC 2 | | 3 | | | 5 | | | | | |
| 11 | MFCC 3 | | | | 5 | | | 2 | | 4 | 2 |
| 12 | MFCC 4 | 3 | 5 | 1 | | | 2 | | 4 | 8 | |
| 13 | MFCC 5 | | | 6 | | | 7 | | | | |
| 14 | MFCC 6 | | | 6 | | | | | | | |
| 15 | MFCC 7 | | | | | | 5 | 5 | | | |
| 16 | MFCC 8 | 6 | | | | | 7 | | 6 | | 8 |
| 17 | MFCC 9 | 2 | | 2 | 2 | | | 6 | | 3 | |
| 18 | MFCC 10 | | 7 | | | 3 | | 7 | | | |
| 19 | MFCC 11 | | | | 6 | | | | | 6 | 5 |
| 20 | MFCC 12 | | | 5 | | | | | 2 | | |
| 21 | MFCC 13 | | | | | | | | 5 | | 9 |
| 22 | Chroma vector 1 | | | | | | | | | | |
| 23 | Chroma vector 2 | 5 | | | | | | | | | |
| 24 - 30 | Chroma vector 3-9 | | | | | | | | | | |
| 31 | Chroma vector 10 | 7 | | | | | | | | | |
| 32 | Chroma vector 11 | 4 | | | | | | | | | |
| 33 | Chroma vector 12 | | | | | | | | | | |
| 34 | Chroma std | | | | | | | | | | |

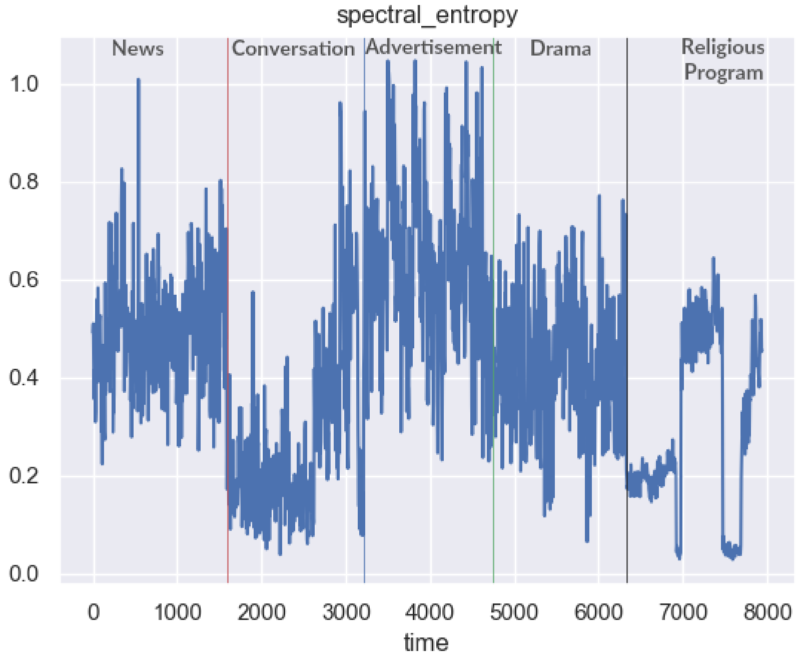


Figure 3.6: Frequency spectrum of Spectral Entropy against the five classes

Finally, 24 features are manually selected out of all observed features. Table 3.5 lists the selected features with the rank obtained by computing the importance of the features. All the features are not certainly used for each class.

## 3.4 Data Pre-processing

Data pre-processing phase helps to create raw data from audio files. Pre-processing prepares the data in a consistent way before extract the features. Data pre-processing stage consists of data formatting, segmentation and labeling, and silence removal that are discussed in Sections 3.4.1, 3.4.2, and 3.4.3 respectively.

### 3.4.1 Data Formatting

There are several considerations to discuss in data formatting such as the file types, selecting the channel (i.e. stereophonic or monophonic), and data resampling etc. There are two audio file types as lossless and lossy. WAV, AIFF, APE are the examples for lossless file types. MP3, AAC, WMA are some of the lossy file types. Lossless keeps the original audio signal as it is and lossy compresses the audio signal to save the space. Hence there is a high tendency to loss important features in the audio signal when we use lossy files. Therefore in order to keep the quality, WAV

Table 3.5: Selected features for One-Vs-All model with ranks

| | **Features** | **SVM number** | | | | |
|---|---|---|---|---|---|---|
| | | **1** | **2** | **3** | **4** | **5** |
| 1 | ZCR | | | 4 | | 7 |
| 2 | Energy | 5 | | | | 2 |
| 3 | Energy entropy | | 8 | 13 | | 8 |
| 4 | Spectral centroid | | | 2 | | 5 |
| 5 | Spectral spread | | | 5 | | 1 |
| 6 | Spectral entropy | | 4 | 1 | | |
| 7 | Spectral flux | | | 12 | | 9 |
| 8 | Spectral rolloff | | 5 | 3 | | |
| 9 | MFCC 1 | | 2 | | | |
| 10 | MFCC 2 | | 1 | 6 | | |
| 11 | MFCC 3 | | | 7 | 4 | 3 |
| 12 | MFCC 4 | 1 | | | 3 | |
| 13 | MFCC 5 | | 10 | 9 | | |
| 14 | MFCC 6 | | | | | 10 |
| 15 | MFCC 7 | | | 10 | | 11 |
| 16 | MFCC 8 | | 9 | | 2 | |
| 17 | MFCC 9 | 2 | 7 | | 6 | 4 |
| 18 | MFCC 10 | | 6 | 8 | | |
| 19 | MFCC 11 | 3 | | | 7 | 6 |
| 20 | MFCC 12 | | 3 | | 1 | |
| 21 | MFCC 13 | | | | 5 | |
| 22- 23 | Chroma vector 1-2 | | | | | |
| 24 | Chroma vector 3 | 4 | | | | |
| 25 - 27 | Chroma vector 4-6 | | | | | |
| 28 | Chroma vector 7 | | | 11 | | |
| 29 - 31 | Chroma vector 8-10 | | | | | |
| 32 | Chroma vector 11 | 6 | | | | |
| 33 | Chroma vector 12 | | | | | |
| 34 | Standard deviation of Chroma vector | | | | | |

file type is chosen.

The radio broadcasting contents come through stereophonic channels or monophonic channels. The stereophonic channel keeps a combination of different audio units comes from different channels while monophonic channel keeps the audio units come from a single channel. Even though stereophonic channels keep different audio units from different channels, features only from one channel are captured in feature extraction instead of all. Therefore when using stereophonic channels in audio processing, important features might not be captured. Accordingly, the monophonic channel is chosen as the channel type to generate good feature vectors.

In analog to digital conversion, the signals reproduce samples. The number of samples per second is known as the sample rate. The alliance between the sample rate $F_s$ and sampling period $T_s$ is as follows.

$$F_s = \frac{1}{T_s} Hz \tag{3.1}$$

For this work, 44.1 kHz is selected as the sample rate. The reason is that FM radio has a bandwidth of 15 kHz approximately. Bandwidth is the difference between the highest and lowest frequencies carried in an audio stream. According to Nyquist Shannon theorem, the highest frequency is half of the sample rate. Practically, the highest frequency for a radio stream is in between 22050 Hz - 20000 Hz because the highest audible frequency of a human is 20000 Hz [22]. Thus, 44100 kHz is the logically best choice for radio sampling.

### 3.4.2 Segmentation and Labeling

In this stage, the WAV files are segmented into different class divisions by manually listening to the audio recordings and labeled them. For the segmentation, the “Audacity” tool is used. As mentioned in Section 1.7, when labeling the files single speaker programs and telephone conversations are labeled as conversations. Table 3.6 shows annotated class label with the time duration of a segment in each class.

### 3.4.3 Silence Removal

Silence is the places where the amplitude of an audio signal is zero. Silence is an important feature to recognize speaker roles because there are identical patterns in the poses of a speaker. However, silence factor affects imperfectly to this classification because when we split data into small frames

Table 3.6: Class segmentation and labeling

| **Label** | **Radio events** | **Time duration of a segment** |
|---|---|---|
| ‘news’ | News programs | [0.5-4] min |
| ‘conversations’ | Single speaker programs<br>Multi-speaker programs<br>Telephone conversation | [1-7] min |
| ‘advertisements’ | Advertisements without jingles/ songs | [0.2-2] min |
| ‘radio drama’ | Radio drama | [1-5] min |
| ‘religious programs’ | Paritta - Sutta Chants | [1-17] min |
| ‘other’ | Songs<br>Musics<br>Jingle | [1-12] min |

there can be frames with only silence. As a consequence, training data with the silence frames will affect the overall classification results. As identified in Section 5.3.5, news and the conversations are the two content categories that found silence phases frequently. Hence, we remove silence only from news and conversations. For that, we use a silence detection function in ”librosa” library. Figure 3.7 shows an example for news segment with silence.

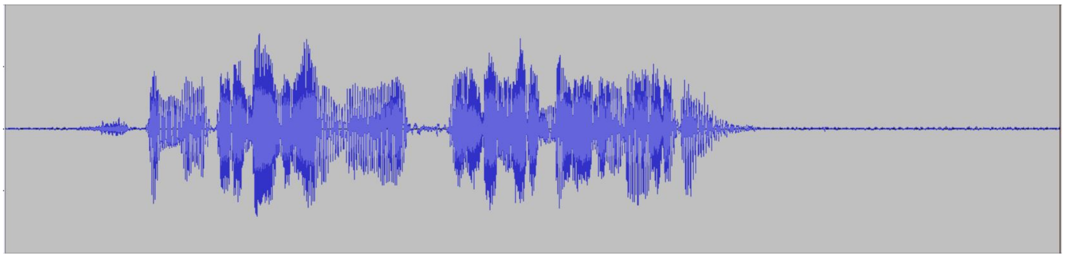


Figure 3.7: News content with silence

## 3.5 Feature Extraction

In audio signal classification, feature extraction is the most important component. Feature extraction process consists of two stages as frame blocking and feature extraction. Extracted features are expressed as feature vectors.

### 3.5.1 Frame Blocking

An audio signal is being as a longer wave in time domain. Frame blocking concept is needed for the purpose of extracting the frequencies of short time signals of a longer signal. In frame blocking longer signal split into short frames. The accuracies of the classification model vary according to the selected frame length. Therefore, different frame lengths such as 0.25s, 2.5s, 3s, 4s, and 5s are tested to get a solid frame length. The obtained results are shown in Section 5.3.3. By considering the accuracies of each, 5s of frame length is selected for framing. Then, segment the raw data into frames and pass to the feature extraction process.

### 3.5.2 Features Extraction

Framed raw data is input to the feature extraction process to manipulate raw data and transform them into feature vectors by extracting feature values. “pyAudioAnalysis” python library is used to extract the features [23]. After extracting features, each frame is represented by a vector consist of various feature values. Table 3.7 lists all the features that are used in both One-Vs-One and One-Vs-All models for extracting features.

Table 3.7: Extracted features

| **Index** | **Feature** | **Description** |
|---|---|---|
| 0 | Zero Crossing Rate | The rate of the sign changes from positive to negative of a signal during a particular time frame. |
| 1 | Energy | The sum of squares of the signal values, normalized by the respective frame length. |
| 2 | Energy Entropy | The entropy of normalized energies of subframes. It represents a measure of abrupt changes in a signal. |
| 3 | Spectral Centroid | The center of gravity (center of the mass) of the spectrum. |
| 4 | Spectral Spread | The second central movement of the spectrum. |
| 5 | Spectral Entropy | The entropy of the normalized spectral energies for a set of subframes. |
| 6 | Spectral Flux | The squared difference between normalized magnitudes of the spectra of the two successive frames. |
| 7 | Spectral Rolloff | The frequency below some percentage of the magnitude distribution of the spectrum is concentrated. |
| 8-20 | MFCCs | Mel Frequency Cepstral Coefficients is a representation of a short term power spectrum of a sound. These frequency bands are not linear but distributed according to the Mel-scale. |
| 21-32 | Chroma Vector | Chromagram is closely related to 12 different pitch classes of Western type music. Mainly used to find pitch differences in an audio data. |

## 3.6 Designs of the Classification Models

Our goal is to investigate the multi-class classification problem of voice dominant audio contents in radio events. As concluded in the literature review, multi-class SVM classifier is selected as the best approach which fits into our problem domain beneficially. To acquire higher performance, we decided to build multi-class SVM in two ways parallelly and compare which one is the most reliable. Henceforth, this section demonstrates the designs of both One-Vs-One and One-Vs-All methods which are going to implement.

### 3.6.1 One Vs. One Model

In this approach, N(N-1)/2 number of binary SVMs are implemented to classify N number of classes. Therefore, we design ten binary SVMs where each SVM classifies a pair of classes as follows.

SVM 1 - News Vs. Advertisements

SVM 2 - News Vs. Conversations

SVM 3 - News Vs. Radio drama

SVM 4 - News Vs. Religious program

SVM 5 - Advertisements Vs. Conversations

SVM 6 - Advertisements Vs. Radio drama

SVM 7 - Advertisements Vs. Religious program

SVM 8 - Conversations Vs. Radio drama

SVM 9 - Conversations Vs. Religious program

SVM 10 - Radio drama Vs. Religious program

A SVM classifies $i^{th}$ and $j^{th}$ classes, for a data point $D = (x_t, y_t)$ as follows,

$$\text{if } (w^{ij})^T \phi(x_t) + b^{ij} \geq 1 - {\varepsilon_t}^{ij} \text{ ; } y_t = \text{class i} \tag{3.2}$$

$$\text{if } (w^{ij})^T \phi(x_t) + b^{ij} \leq -1 + {\varepsilon_t}^{ij} \text{ ; } y_t = \text{class j} \tag{3.3}$$

$(w^{ij})^T\phi(x_t) + b^{ij}$ called as decision boundary where $w^{ij}$ is the weight vector, $x_t$ is the input vector, $b^{ij}$ is the bias, and data $x_t$ is mapped to a higher dimensional space by the function $\phi$. The motivation behind the SVM is maximizing the decision boundary/ margin between two groups of data (i.e. SVM finds the decision boundary that is furthest away from any data points). The maximized decision boundary for $i^{th}$ and $j^{th}$ classes acquired by minimizing the magnitude of $w^{ij}$. Hence, to find the maximum margin, the magnitude of $w^{ij}$ should be minimized as in the Equation 3.4. When the data is non-linearly separable, $C\sum_t {\varepsilon_t}^{ij}$ is introduced as the penalty terms to reduce the number of training errors.

$$min_{w^{ij},b^{ij},\varepsilon^{ij}} \frac{1}{2}(w^{ij})^T(w^{ij}) + C\sum_t {\varepsilon_t}^{ij} \tag{3.4}$$

One-Vs-One model builds ten binary SVMs to classify five classes. Since there are ten decision boundaries, the relevant class for a particular data point is identified using a voting strategy. If a $(w^{ij})^T\phi(x_t) + b^{ij}$ says the data point belongs to $i^{th}$ class, then vote for the $i^{th}$ class. Otherwise, vote for the $j^{th}$ class. Then the class with the largest vote is taken as the predicted class. This is called as “Max Winning” strategy.

### 3.6.2 One Vs. All Model

One-Vs-All method construct N number of binary SVMs where it has N number of classes to classify. Therefore five SVM classifiers are designed as follows.

SVM 1 - News Vs. Rest

SVM 2 - Conversations Vs. Rest

SVM 3 - Advertisements Vs. Rest

SVM 4 - Radio drama Vs. Rest

SVM 5 - Religious program Vs. Rest

Each SVM is trained with the whole dataset where the the data belongs to $i^{th}$ class with positive labels and rest of the data with negative labels. A SVM solves data point $D = (x_t, y_t)$ for $i^{th}$ class according to the following equations Equation 3.5 and Equation 3.6.

$$\text{if } (w^i)^T\phi(x_t) + b^i \geq 1 - {\varepsilon_t}^i \text{ ; } y_t = \text{class i} \tag{3.5}$$

$$\text{if } (w^i)^T\phi(x_t) + b^i \leq -1 + {\varepsilon_t}^i \text{ ; } y_t \neq \text{class i} \tag{3.6}$$

To find the maximum margin, the magnitude of $w^i$ should be minimized as in the Equation 3.7 where $C$ is the constant used to reduce training error.

$$min_{w^i,b^i,\varepsilon^i}\frac{1}{2}(w^i)^T(w^i) + C\sum_t {\varepsilon_t}^i \tag{3.7}$$

One-Vs-All model implements five binary SVMs to classify each class individually. After training five classifiers, the class of a data point $x$ is predicted by finding the decision boundary which has the maximum value. The Equation 3.8 gives the prediction function for data point $x$.

$$\text{class of x = max}((w^i)^T\phi(x) + b^i) \text{ where i = 1...N, N is the number of classes} \tag{3.8}$$

## 3.7 Evaluation

As the evaluation procedure, manual evaluation with ground truth data is chosen. $40\%$ data from the whole dataset is taken as the ground truth data for the evaluation. All the data are annotated manually. Since we have two models, they are comparably evaluated under different criteria.

One-Vs-One model is evaluated using “max winning” strategy. One-Vs-All model is evaluated using the maximum value acquired by the decision functions. In order to measure the accuracies of each class, confusion matrices are formed. Then necessary refinements are applied. The evaluation plan is explained in a detailed manner in Chapter 5.

## 3.8 Summary

With this chapter the conceptual overview of the research design has been discussed. In each subsequent sections, the individual components in the research design were outlined with a description of the connectivity between these components. Also, a brief opening of the evaluation plan was given. The justifications to the selections and all considerations were mentioned within components. The implementation of these components will be offered further in Chapter 4.

# Chapter 4

# Implementation

## 4.1 Introduction

This chapter addresses the implementation details of the proposed design. Section 4.2 outlines the tools that used in the process. Section 4.3 establishes the implementation details of data analyzing, preprocessing, feature extraction, and two multi-class SVM classifiers.

## 4.2 Software Tools

This section discusses the software tools that are mainly involved for the implementation. For the data segmentation, and labeling "Audacity" tool is used. For data analyzing, pre-processing, feature extraction, and the implementations, 'Python 3.6.5' is used as the programming language with extensions of 'PyAudioAnalysis', 'Librosa', 'scikit-learn', and 'matplotlib' libraries.

### 4.2.1 Audacity

Audacity is a free and open-source digital audio editor and recording application software, available for Windows, macOS/OS X and Unix-like operating systems. Audacity is used as a tool for audio segmentation and labeling.

### 4.2.2 PyAudioAnalysis

'PyAudioAnalysis' is an open source python library which provides a wide range of audio analysis features such as content visualization, feature extraction, segmentation and clustering, audio regressions etc. PyAudioAnalysis is already used in the audio analysis research areas such as build smart home functionalities through sound detection, emotion recognition, and music classification.

### 4.2.3 Librosa

Librosa is also an open source python library used for audio processing. It provides the functionalities through APIs for reading audio signals, framing and windowing, feature extraction and sequential modeling.

### 4.2.4 Scikit-learn

Scikit-learn is a simple and efficient library used in python for machine learning and data mining. This is an open source library which provides functionalities such as model selection, dimensionality reduction, clustering, and classification.

### 4.2.5 Matplotlib

'Matplotlib' is a two-dimensional plotting library in python which provides publication quality figures. This is a numerical extension of 'Numpy' library. For a simple plotting 'pyplot' module provides interfaces which are very close to Matlab.

## 4.3 Implementation Details

### 4.3.1 Pre-processing

In the data preprocessing stage, original audio clips are read and pre-processed according to the pre-requisites. As the first step, stereophonic data converts to monophonic. Then resample the data by taking the sample rate as 44100 Hz. In order to pre-process the reading data, 'librosa' *load()* function is used. When read the WAV files the *load()* function converts it into monophonic by enabling *mono* = *True* and resample the audio stream by using *resample()* function in 'librosa'.

**Code 1** Data reading and resampling

```python
import librosa as lb

sample_rate = 44100
x0, sr = lb.load(path, mono=True, sr=None)
x = lb.resample(x0, sr, sample_rate);
```

Then use 'librosa' *split()* function for remove silence in news and conversations. This function removes the data points which are below to pre-defined threshold (in decibels) considering as silence and returns the time intervals which are not considered as silence.

**Code 2** Silence removing

```
def remove_silence(x):
    intervals = lb.effects.split(x, top_db=40, frame_length=
        c.frame_size, hop_length=32)
    for ind in range(len(intervals)):
        start_ind, end_ind = intervals[ind][0], intervals[ind][1]
        if start_ind < len(x):
            abs_seg = abs(x[start_ind:end_ind]).mean()
            abs_audio = abs(x).mean()
            if abs_seg < abs_audio - 0.05:
                x[start_ind:end_ind] = 0
    return x

if cls == "news" or cls == "conversations":
first = 0
interval = lb.effects.split(x)
xt = np.zeros_like(x)
print(interval)
for ind, (start, end) in enumerate(interval[first:]):
    xt[start:end] = remove_silence(x[start:end])
    x = xt
```

### 4.3.2 Feature Extraction

As explained in Section 3.5.1 pre-processed data are split into frames. The length of a frame is 5s. Since the sample rate is 44100 Hz, the number of samples per a window is calculated as *frame_length* × *sample_rate*. Then these samples pass into the feature extraction process.

**Code 3** Feature extracting

```
import pyAudioAnalysis.audioFeatureExtraction as fe

sample_rate = 44100
frame_length = 5
win = int(math.floor(frame_length*sample_rate))
F, f_names = fe.stFeatureExtraction(x,sample_rate,win,win)
```

There are mainly four types of features in the audio analysis as time domain features, frequency domain features, cepstral features, and chroma features. Energy distribution and energy entropy are belong to time domain features. Spectral features are belong to frequency domain features. Chroma vector contains chroma features, and MFCC features are belong to cepstral features. “PyAudioAnalysis” library extract all these types of features.

**Code 4** Create feature vector

```
import pyAudioAnalysis.audioFeatureExtraction as fe
import numpy as np

no_time_spectral_features = 8
no_mfcc_features = 13
no_chroma_features = 12
feature_vec = np.zeros((33, 1))

feature_vec[0] = fe.stZCR(x)

feature_vec[1] = fe.stEnergy(x)

feature_vec[2] = fe.stEnergyEntropy(x)

[feature_vec[3], feature_vec[4]] =fe.stSpectralCentroidAndSpread \
    (x, sample_rate)

feature_vec[5] = fe.stSpectralEntropy(x)

feature_vec[6] = fe.stSpectralFlux(x, x_previous)

feature_vec[7] = fe.stSpectralRollOff(x, 0.90, sample_rate)

feature_vec[no_time_spectral_features: no_time_spectral_features + \
    no_mfcc_features, 0] = fe.stMFCC(x, fbank, no_mfcc_features).copy()

feature_vec[no_time_spectral_features  + \
    no_mfcc_features: no_time_spectral_features + no_mfcc_features + \
        no_chroma_features-1] = fe.stChromaFeatures(x, fs)
```

### 4.3.3 One Vs. One SVM Model

One Vs. One multi-class SVM is a compound of binary SVMs. Since the total number of classes are five, ten binary SVMs are implemented for each class pair as explained in Section 3.6.1. Code 5 shows modeling an SVM for classify news and advertisements by input the feature vector which includes news and advertisements.

**Code 5** Model an SVM for One Vs. One

```
svm1 = model1.news_advertisement(feature_news_ advertisement)
```

*sklearn.svm* module in the 'Scikit-learn' python library is used for the implementation of SVMs. Following code fragment shows the implementation of SVM which classifies news and advertisements.

**Code 6** Train an SVM

```
import sklearn.svm as svm

def news_advertisement(feature_vec):
    X = feature_vec[:, :cols-1]
    Y = feature_vec[:, cols-1]
    Y = Y.astype(int)

    x_train, x_test, y_train, y_test = cv.train_test_split(X, Y, test_size=0.3)

    svm_model1 = svm.SVC(kernel='rbf', probability=True)

    kf = cv.KFold(n_splits=10, random_state=None)

    for train_indices, test_indices in kf.split(x_train):
        svm_model1.fit(x_train[train_indices],y_train[train_indices]) \
            .score(x_train[test_indices], y_train[test_indices])

    y_predict = svm_model1.predict(x_test)
```

The input dataset consists of the data from two different classes. Then the input dataset split into 70% and 30% as the training data and testing data respectively. This testing data set is used to measure the training accuracy of the SVM model. Since the dataset is not linearly separable in two-dimensional space, radial basis function (RBF) is used as the kernel function in order to uplift the data onto a higher dimensional feature space where a linear decision boundary is used to separate the classes. Then train the ten models using the training dataset. 10-fold cross-validation is used to prevent from the bias of outliers. For the class prediction in One-Vs-One method, we implement the “Max Winning” strategy as coded in Code 7.

### 4.3.4 One Vs. All SVM Model

One Vs. All multi-class SVM is an integration of five binary SVMs, where each implements for separate individual class from the rest of the classes. For an example, Code 8 shows the SVM model for classify news from the rest.

The implementation of each SVM is the same as the implementation of binary SVM which is shown in Section 4.3.3. The only difference is the feature vectors that are input to the SVM includes all the training data labeled as one class with positive numbers and others labeled as zeros.

**Code 7** Prediction of One Vs. One - ”MAx Winning strategy”

```
def compute_mode(numbers):
    mode = []
    counts = Counter(numbers)
    maxcount = max(counts.values())
    for num,count in counts.items():
        if count == maxcount:
            mode.append(num)
    return mode

def max_win(data):
    y_predict = np.array(model1.predict(data))
    y_predict = np.append(y_predict,(model2.predict(data)),axis=0)
    y_predict = np.append(y_predict,(model3.predict(data)),axis=0)
    y_predict = np.append(y_predict,(model4.predict(data)),axis=0)
    y_predict = np.append(y_predict,(model5.predict(data)),axis=0)
    y_predict = np.append(y_predict,(model6.predict(data)),axis=0)
    y_predict = np.append(y_predict,(model7.predict(data)),axis=0)
    y_predict = np.append(y_predict,(model8.predict(data)),axis=0)
    y_predict = np.append(y_predict,(model9.predict(data)),axis=0)
    y_predict = np.append(y_predict,(model10.predict(data)),axis=0)
    max_val = compute_mode(y_predict)
```

**Code 8** Model an SVM for One Vs. All

```
svm1 = model1.news(feature_news_other)
```

For testing, we follow a different strategy which is mentioned in Section 3.6.2. In this method, the class is predicted by finding the decision boundary which has the largest value for the decision function as in the Code 9. The largest value of the decision boundaries is taking as the class which has the maximum probability of being in the class.

## 4.4 Summary

This chapter presents the utilized software tools and libraries followed by the important characteristics of them. Then, the implementation details of the major components in the proposed design are described in each section including codes of the functionalities. Pre-processing, feature extraction, One Vs. One SVM model, and One Vs. All SVM model are the major components that discussed in the implementation details. Other code fragments of the research are included in Appendix A.

**Code 9** Prediction strategy of One Vs. All

```python
def predict(data):
    prob = []

    y1 = md1.predict(data)
    p1 = md1.predict_proba(data)
    (y1 == 0) if prob.append(p1[0][0]) else prob.append(p1[0][1])

    y2 = md2.predict(data)
    p2 = md2.predict_proba(data)
    (y2 == 0) if prob.append(p2[0][0]) else prob.append(p2[0][1])

    y3 = md3.predict(data)
    p3 = md3.predict_proba(data)
    (y3 == 0) if prob.append(p3[0][0]) else prob.append(p3[0][1])

    y4 = md4.predict(data)
    p4 = md4.predict_proba(data)
    (y4 == 0) if prob.append(p4[0][0]) else prob.append(p4[0][1])

    y5 = md5.predict(data)
    p5 = md5.predict_proba(data)
    (y5 == 0) if prob.append(p5[0][0]) else prob.append(p5[0][1])

    predict = np.array([y1, y2, y3, y4, y5])
    y_predict = predict

    y_predict_ind = np.nonzero(y_predict)[0]
    y_predict = [predict[i] for i in y_predict_ind]
    if len(y_predict) > 1:
        prob1 = [prob[i] for i in y_predict_ind]
        max_prob = prob1.index(max(prob1))
        y = y_predict[max_prob]

    return int(y)
```

# Chapter 5

# Results and Evaluation

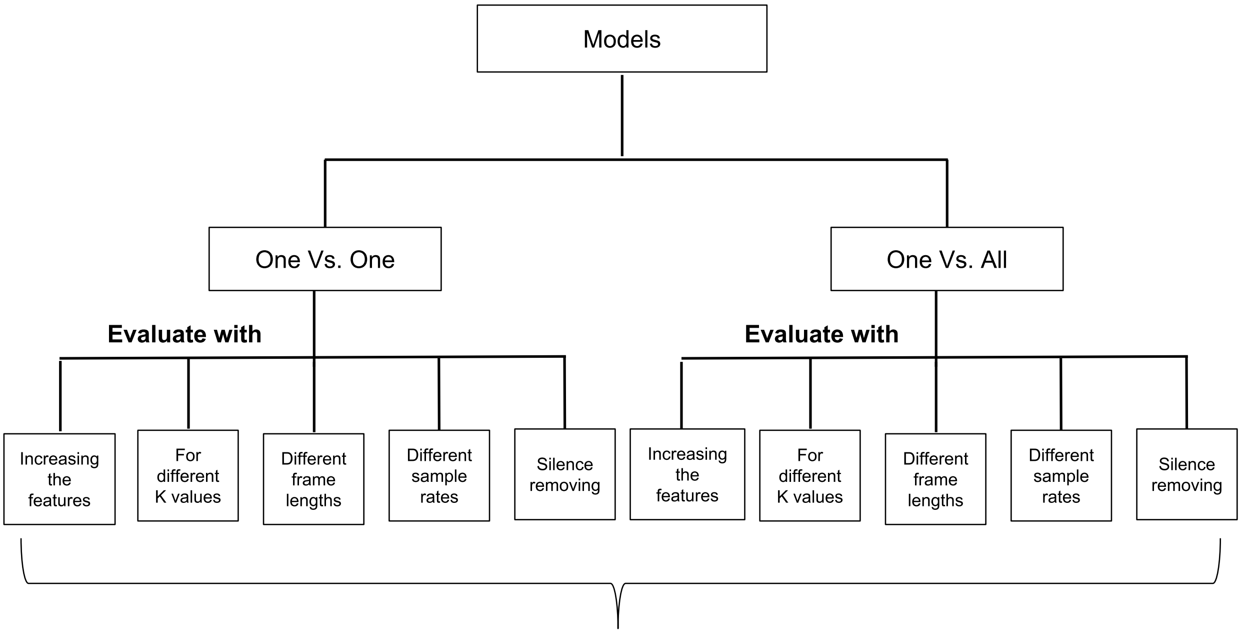
## 5.1 Introduction

This chapter illustrates the success level of the proposed methodology with the results. Section 5.2 presents the evaluation model. Section 5.3 demonstrates the obtained results under different criteria utilizing tables and diagrams. The observations are discussed in Section 5.4. The conclusion of the chapter discussed in Section 5.5.

## 5.2 Evaluation Model

Through the evaluation, we compare and contrast the performance of One-Vs-One and One-Vs-All multi-class SVM models. Ground truth data is required to evaluate the accuracies of the two models. $40\%$ of data from the whole dataset take as the ground truth data which are never being in the training set. To annotate the ground truth data, “Audacity” tool is used. These data are segmented into the news, conversations, advertisements, drama, and religious programs and manually annotate the data.

The models are evaluated under different criteria are as depicted in Figure 5.1. By increasing the features, the performances of the models are evaluated. K-fold cross validation with different K values is used to find the best K value for these models. Additionally, the two models are evaluated using selected frame lengths, selected sample rates, before silence removal and after silence removal. The performances of the two models are presented using graphs and confusion matrices. Then do the necessary refinements to both models. Select the most reliable model by considering the evaluation results. Since the proposed approach is a machine learning model, the precision value is taken to measure performance.



In these experiments, while changing one parameter other parameters are keeping as constants.

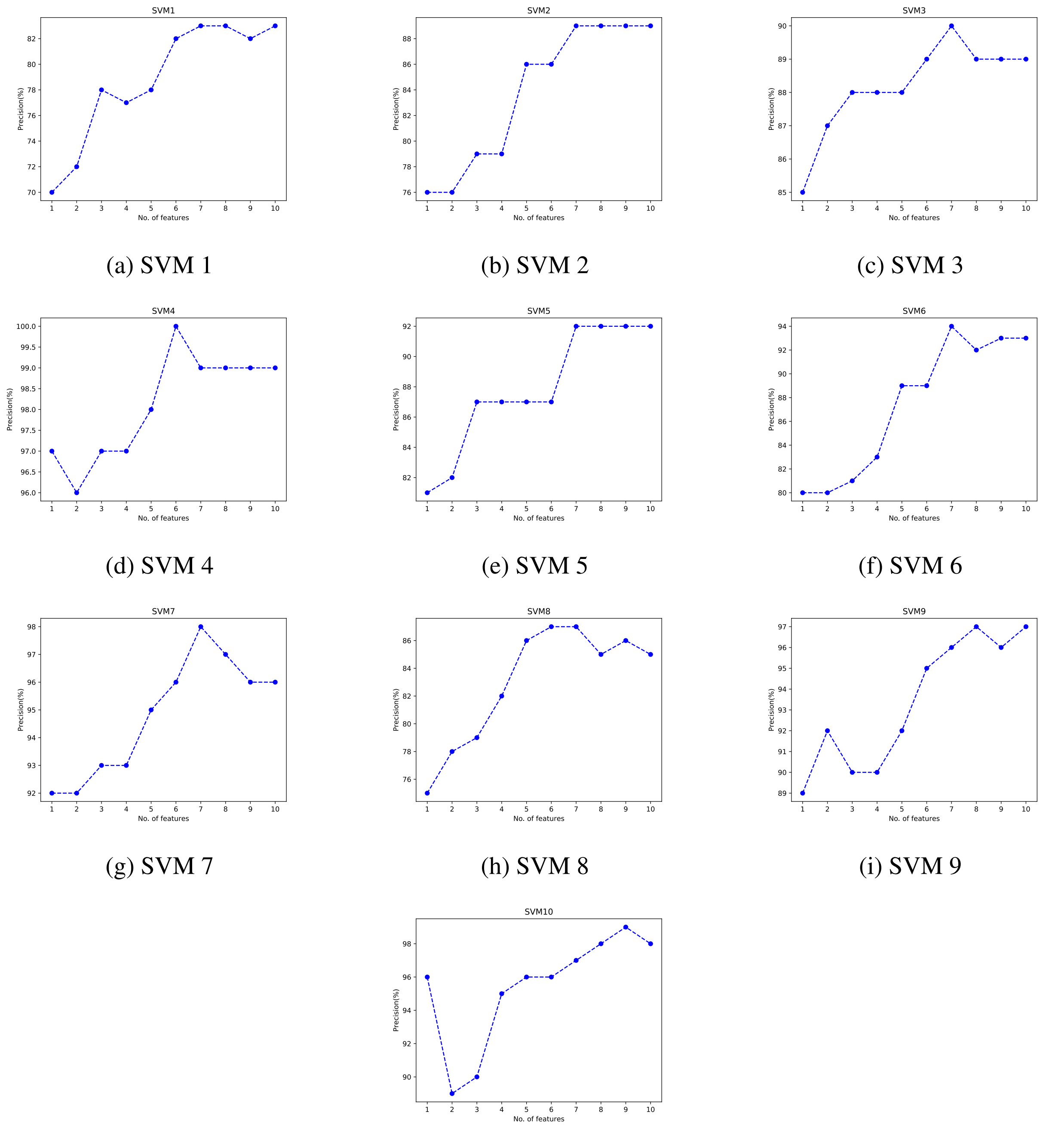
- K value = 10
- Frame length = 5 s
- Sample rate = 44100 Hz
- With silence remove from “news” and “conversations”

Figure 5.1: Evaluation Model

## 5.3 Results

### 5.3.1 Evaluate by Increasing Features

As discussed in Section 3.3, features that selected for each SVM are ranked according to its importance. To get the best sub set of features, the selected features are input to the models iteratively by adding features in each iteration according to the ranking. Figure 5.2 and Figure 5.3 indicates the obtained results for One-Vs-One and One-Vs-All models respectively. The best subset of features for each SVM is selected through this process. According to the graphs shown in Figure 5.2 and Figure 5.3, the obtained features are lists in Table 5.1.



(a) SVM 1

(b) SVM 2

(c) SVM 3

(d) SVM 4

(e) SVM 5

(f) SVM 6

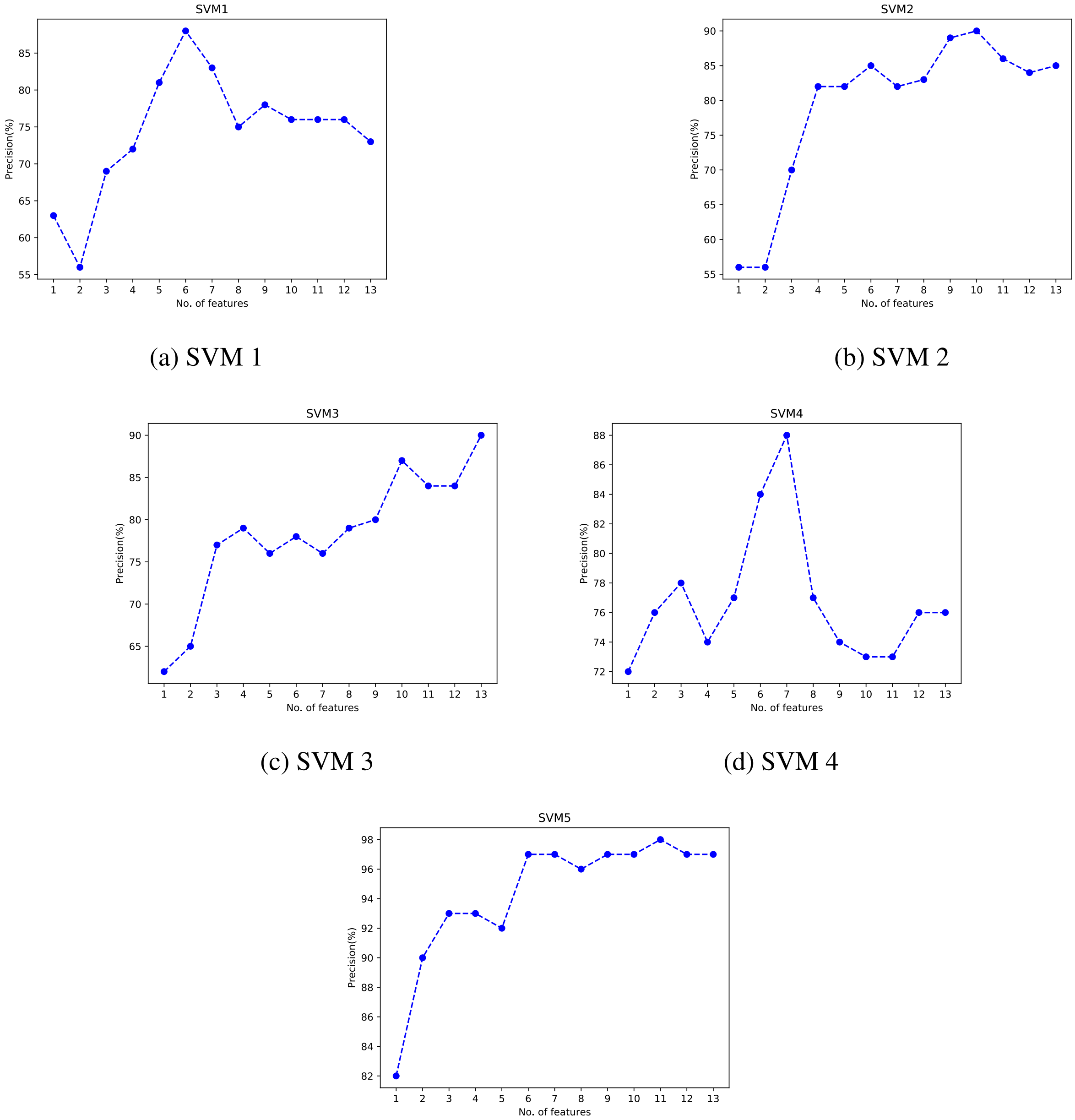
(g) SVM 7

(h) SVM 8

(i) SVM 9

(j) SVM 10

Figure 5.2: Increasing features of binary SVMs in One Vs. One model



(a) SVM 1

(b) SVM 2

(c) SVM 3

(d) SVM 4

(e) SVM 5

Figure 5.3: Increasing features of binary SVMs in One Vs. All model

Table 5.1: Best subset of features for each SVM

| | Binary SVM | No. of features | Best subset of features from the selected features |
|---|---|---|---|
| **One Vs One** | news/ advertisement | 7 | Energy entropy, MFCC 9, MFCC 4, chroma 11, chroma 2, MFCC 8, chroma 10 |
| | news/ conversation | 7 | Spectral entropy, Spectral centroid, MFCC 2, Spectral rolloff, MFCC 4, MFCC 5, MFCC 10 |
| | news/ drama | 7 | MFCC 4, MFCC 9, Spectral centroid, Spectral spread, MFCC 12, MFCC 6, Energy |
| | news/ religious prog. | 6 | Spectral spread, MFCC 9, Energy, Spectral centroid, MFCC 3, MFCC 11 |
| | advertisement/ conversations | 7 | Spectral entropy, Spectral centroid, Spectral flux, MFCC 10, Spectral rolloff, MFCC 2, MFCC 5 |
| | advertisement/ drama | 7 | Spectral centroid, MFCC 4, Spectral spread, Energy entropy, Spectral entropy, MFCC 7, MFCC 8 |
| | advertisement/ religious prog. | 7 | Spectral spread, MFCC 3, Spectral centroid, Energy, MFCC 7, MFCC 9, MFCC 10 |
| | conversations/ drama | 6 | Spectral entropy, MFCC 12, Spectral rolloff, MFCC 4, MFCC 13, MFCC 8 |
| | conversations/ religious prog. | 8 | Energy, Spectral spread, MFCC 9, MFCC 3, Energy entropy, MFCC 11, MFCC 1, MFCC 4 |
| | drama/ religious prog. | 9 | Spectral spread, MFCC 3, Energy, Spectral centroid, Energy entropy, MFCC11, MFCC13,ZCR,MFCC 8 |
| **One Vs All** | news/ other | 6 | MFCC 4, MFCC 9, MFCC 11, Chroma 3, Energy, Chroma 11 |
| | conversations/ other | 10 | MFCC 2, MFCC 1, MFCC 12, Spectral entropy, Spectral rolloff, MFCC 10, MFCC 9, Energy entropy, MFCC 8, MFCC 5 |
| | advertisement/ other | 13 | Spectral entropy, Spectral centroid, Spectral rolloff, ZCR, Spectral spread, MFCC 2, MFCC 3, MFCC 10, MFCC 5, MFCC 7, Chroma 7, Spectral flux, Energy entropy |
| | drama/ other | 7 | MFCC 12, MFCC 8, MFCC 4, MFCC 3, MFCC 13, MFCC 9, MFCC 11 |
| | religious prog./ other | 11 | Spectral spread, Energy, MFCC 3, MFCC 9, Spectral centroid, MFCC 11, ZCR, Energy entropy, Spectral flux, MFCC 6, MFCC 7 |

### 5.3.2 Evaluate with K-Fold Cross Validation for Different K

K-fold cross validation is a process used to evaluate machine learning models. When we define K, the training dataset split into K samples as one sample for validation and others for training. This process does iteratively in K times by changing the validation sample in each iteration. Because of the limited dataset in our classification database, K-fold cross-validation is used to train the model over newcomers. Here we define $K = 2, 3, ..., 10$ in order to evaluate the two models against different K values. The obtained precision values are shown in Table 5.2. Figure 5.4 depicts the variation of the performance when increasing the K value.

Table 5.2: K-Fold cross validation with different K

| **K value** | **One-Vs-One** | **One-Vs-All** |
|---|---|---|
| 2 | 83.73% | 82.30% |
| 3 | 84.42% | 82.59% |
| 4 | 84.61% | 82.16% |
| 5 | 84.90% | 82.27% |
| 6 | 85.08% | 83.19% |
| 7 | 85.13% | 82.86% |
| 8 | 85.11% | 82.83% |
| 9 | 85.15% | 83.34% |
| 10 | 85.73% | 83.37% |

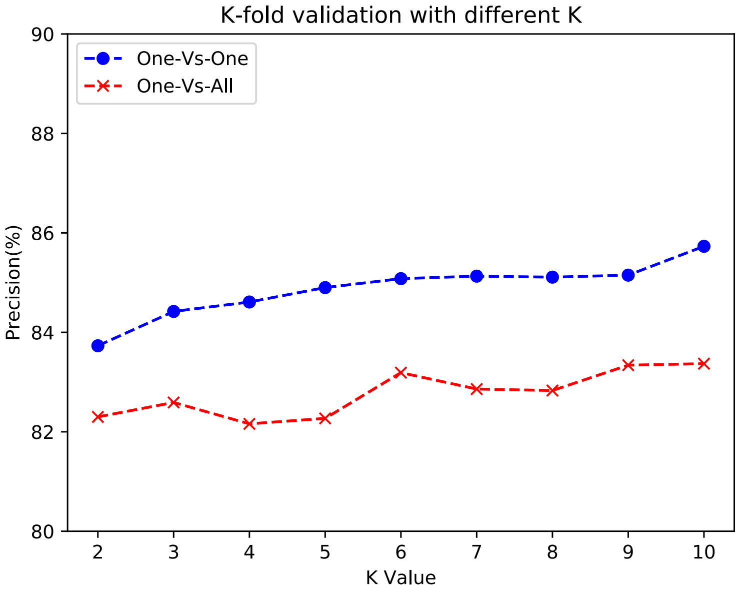


Figure 5.4: Variation of precision value against K

### 5.3.3 Evaluate with Different Frame Sizes

Selecting a correct frame size to extract the features is essential when comes to a classification problem. In the closest work to this research done by C. Weeratunga [2], the researcher has used 2.5 s as the frame size. Other than that, the works that are found in the literature have used different frame sizes such as 25 ms, 0.25 s, and 0.3 s etc. Therefore, the model is evaluated with respect to different frame sizes and reports the results for the chosen frame sizes 0.25 s, 2.5 s, 4 s, and 5s. The reason behind to select 4s and 5s are discussed in Section 5.4. When changing the frame size, other parameters such as K value and sample rate are constants.

Table 5.3 - Table 5.6 shows the obtained results of One-Vs-One model. Table 5.7 - Table 5.10 shows the obtained results of One-Vs-All model.

Table 5.3: Confusion matrix of One-Vs-One model with frame size 0.25s

| | | **Predicted Class** | | | | | Support | Precision(%) |
|---|---|---|---|---|---|---|---|---|
| | | **news** | **conv** | **advert** | **drama** | **relProg** | | |
| **True Class** | **news** | **3034** | 914 | 1270 | 1281 | 86 | 6585 | 48% |
| | **conv** | 1124 | **3290** | 907 | 737 | 539 | 6597 | 55% |
| | **advert** | 1479 | 747 | **2902** | 1222 | 224 | 6574 | 49% |
| | **drama** | 648 | 754 | 843 | **3738** | 468 | 6451 | 50% |
| | **relProg** | 30 | 315 | 59 | 523 | **5698** | 6625 | 81% |
| Average | | | | | | | | 56.45% |

Table 5.4: Confusion matrix of One-Vs-One model with frame size 2.5s

| | | **Predicted Class** | | | | | Support | Precision(%) |
|---|---|---|---|---|---|---|---|---|
| | | **news** | **conv** | **advert** | **drama** | **relProg** | | |
| **True Class** | **news** | **500** | 32 | 82 | 34 | 0 | 648 | 73% |
| | **conv** | 65 | **479** | 39 | 57 | 15 | 655 | 77% |
| | **advert** | 97 | 34 | **450** | 51 | 5 | 637 | 75% |
| | **drama** | 26 | 48 | 29 | **531** | 3 | 637 | 78% |
| | **relProg** | 0 | 30 | 1 | 6 | **618** | 655 | 96% |
| Average | | | | | | | | 79.81% |

Table 5.5: Confusion matrix of One-Vs-One model with frame size 4s

| | | **Predicted Class** | | | | | Support | Precision(%) |
|---|---|---|---|---|---|---|---|---|
| | | **news** | **conv** | **advert** | **drama** | **relProg** | | |
| **True Class** | **news** | **322** | 11 | 54 | 15 | 0 | 402 | 79% |
| | **conv** | 37 | **309** | 22 | 34 | 6 | 408 | 82% |
| | **advert** | 40 | 16 | **299** | 24 | 3 | 382 | 78% |
| | **drama** | 10 | 27 | 9 | **351** | 0 | 397 | 83% |
| | **relProg** | 0 | 14 | 1 | 1 | **390** | 406 | 98% |
| Average | | | | | | | | 83.73% |

Table 5.6: Confusion matrix of One-Vs-One model with frame size 5s

| | | **Predicted Class** | | | | | Support | Precision(%) |
|---|---|---|---|---|---|---|---|---|
| | | **news** | **conv** | **advert** | **drama** | **relProg** | | |
| **True Class** | **news** | **260** | 8 | 43 | 6 | 0 | 317 | 81% |
| | **conv** | 25 | **247** | 14 | 33 | 6 | 325 | 85% |
| | **advert** | 28 | 12 | **242** | 22 | 3 | 307 | 80% |
| | **drama** | 7 | 13 | 6 | **290** | 0 | 316 | 82% |
| | **relProg** | 0 | 7 | 0 | 2 | **314** | 323 | 98% |
| Average | | | | | | | | 85.20% |

Table 5.7: Confusion matrix of One-Vs-All model with frame size 0.25s

| | | **Predicted Class** | | | | | Support | Precision(%) |
|---|---|---|---|---|---|---|---|---|
| | | **news** | **conv** | **advert** | **drama** | **relProg** | | |
| **True Class** | **news** | 1508 | **3632** | 803 | 634 | 8 | 6585 | 61% |
| | **conv** | 295 | **5640** | 366 | 208 | 88 | 6597 | 27% |
| | **advert** | 483 | **3963** | 1571 | 464 | 93 | 6574 | 45% |
| | **drama** | 190 | **4384** | 752 | 923 | 202 | 6451 | 41% |
| | **relProg** | 12 | **3646** | 37 | 8 | 2922 | 6625 | 88% |
| Average | | | | | | | | 52.22% |

Table 5.8: Confusion matrix of One-Vs-All model with frame size 2.5s

| | | **Predicted Class** | | | | | Support | Precision(%) |
|---|---|---|---|---|---|---|---|---|
| | | **news** | **conv** | **advert** | **drama** | **relProg** | | |
| **True Class** | **news** | **209** | 182 | 224 | 33 | 0 | 648 | 71% |
| | **conv** | 21 | **544** | 22 | 39 | 29 | 655 | 45% |
| | **advert** | 58 | 141 | **386** | 27 | 25 | 637 | 57% |
| | **drama** | 7 | **342** | 43 | 212 | 33 | 637 | 68% |
| | **relProg** | 0 | 11 | 0 | 0 | **644** | 655 | 88% |
| Average | | | | | | | | 65.77% |

Table 5.9: Confusion matrix of One-Vs-All model with frame size 4s

| | | **Predicted Class** | | | | | Support | Precision(%) |
|---|---|---|---|---|---|---|---|---|
| | | **news** | **conv** | **advert** | **drama** | **relProg** | | |
| **True Class** | **news** | **321** | 9 | 63 | 9 | 0 | 402 | 75% |
| | **conv** | 53 | **298** | 14 | 32 | 11 | 408 | 87% |
| | **advert** | 43 | 11 | **278** | 46 | 4 | 382 | 76% |
| | **drama** | 10 | 20 | 13 | **353** | 1 | 397 | 80% |
| | **relProg** | 2 | 4 | 0 | 0 | **400** | 406 | 96% |
| Average | | | | | | | | 82.77% |

Table 5.10: Confusion matrix of One-Vs-All model with frame size 5s

| | | **Predicted Class** | | | | | Support | Precision(%) |
|---|---|---|---|---|---|---|---|---|
| | | **news** | **conv** | **advert** | **drama** | **relProg** | | |
| **True Class** | **news** | **265** | 9 | 38 | 5 | 0 | 317 | 73% |
| | **conv** | 38 | **247** | 4 | 34 | 2 | 325 | 85% |
| | **advert** | 44 | 11 | **198** | 51 | 3 | 307 | 80% |
| | **drama** | 13 | 12 | 6 | **283** | 2 | 316 | 76% |
| | **relProg** | 1 | 11 | 0 | 0 | **311** | 323 | 98% |
| Average | | | | | | | | 83.37% |

Figure 5.5 depicts the variation of the performance of One-Vs-One and One-Vs-All against different frame sizes.

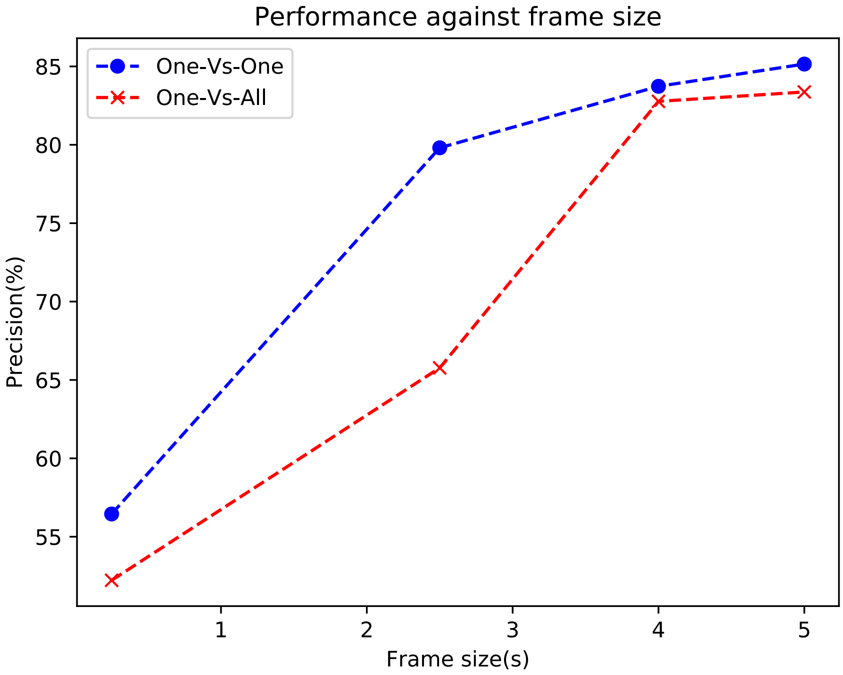


Figure 5.5: Variation of precision value against frame sizes

### 5.3.4 Evaluate with Different Sample Rates

As mentioned in Section 3.4.1, 44100 Hz is selected as the logically best sample rate for our study. In addition to that, 16000 Hz and 22050 Hz were also used as the sample rates in previous works related to radio broadcasting classification. Weeratunga et al [2] proposed 22050 Hz as the sample rate, John Saunders [3] and Vavrek, J. et al [7] proposed 16000 Hz as the sample rates for their studies. Therefore, we evaluate the models with sample rates 16000 Hz and 22050 Hz, and 44100 Hz to find the most solid sample rate for the research. Table 5.11 and Table 5.12 shows the evaluated results of One-Vs-One model for sample rate 16000 Hz and 22050 Hz respectively. Table 5.13 and Table 5.14 shows the obtained results of One-Vs-All model for sample rate 16000 Hz and 22050 Hz respectively. The obtained results of One-Vs-One and One-Vs-All for the sample rate 44100 Hz are same as the results shown in Table 5.6 and Table 5.10.

Figure 5.6 illustrates the changing of performance against different sampling rates for both One-Vs-One and One-Vs-All models.

Table 5.11: Confusion matrix of One-Vs-One model with sample rate 16000 Hz

| | | **Predicted Class** | | | | | Support | Precision(%) |
|---|---|---|---|---|---|---|---|---|
| | | **news** | **conv** | **advert** | **drama** | **relProg** | | |
| **True Class** | **news** | **558** | 115 | 142 | 83 | 0 | 898 | 64% |
| | **conv** | 114 | **587** | 53 | 148 | 4 | 906 | 63% |
| | **advert** | 166 | 68 | **518** | 107 | 25 | 884 | 70% |
| | **drama** | 29 | 134 | 31 | **619** | 69 | 882 | 62% |
| | **relProg** | 0 | 32 | 0 | 34 | **840** | 906 | 90% |
| Average | | | | | | | | 69.73% |

Table 5.12: Confusion matrix of One-Vs-One model with sample rate 22050 Hz

| | | **Predicted Class** | | | | | Support | Precision(%) |
|---|---|---|---|---|---|---|---|---|
| | | **news** | **conv** | **advert** | **drama** | **relProg** | | |
| **True Class** | **news** | **422** | 83 | 98 | 45 | 0 | 648 | 68% |
| | **conv** | 56 | **460** | 29 | 105 | 5 | 655 | 70% |
| | **advert** | 110 | 34 | **434** | 46 | 13 | 637 | 69% |
| | **drama** | 31 | 58 | 69 | **461** | 18 | 637 | 70% |
| | **relProg** | 0 | 22 | 1 | 2 | **630** | 655 | 95% |
| Average | | | | | | | | 74.30% |

Table 5.13: Confusion matrix of One-Vs-All model with sample rate 16000 Hz

| | | **Predicted Class** | | | | | Support | Precision(%) |
|---|---|---|---|---|---|---|---|---|
| | | **news** | **conv** | **advert** | **drama** | **relProg** | | |
| **True Class** | **news** | **258** | 14 | 39 | 6 | 0 | 317 | 75% |
| | **conv** | 29 | **253** | 9 | 31 | 3 | 325 | 84% |
| | **advert** | 42 | 14 | **197** | 51 | 3 | 307 | 78% |
| | **drama** | 13 | 13 | 9 | **279** | 2 | 316 | 76% |
| | **relProg** | 1 | 7 | 0 | 0 | **315** | 323 | 98% |
| Average | | | | | | | | 82.07% |

Table 5.14: Confusion matrix of One-Vs-All model with sample rate 22050 Hz

|  |  | **Predicted Class** |  |  |  |  | Support | Precision(%) |
|---|---|---|---|---|---|---|---|---|
|  |  | **news** | **conv** | **advert** | **drama** | **relProg** |  |  |
| **True Class** | **news** | **263** | 9 | 37 | 8 | 0 | 317 | 75% |
|  | **conv** | 31 | **253** | 4 | 35 | 2 | 325 | 85% |
|  | **advert** | 44 | 13 | **189** | 58 | 3 | 307 | 79% |
|  | **drama** | 12 | 14 | 8 | **281** | 1 | 316 | 74% |
|  | **relProg** | 1 | 9 | 0 | 0 | **313** | 323 | 98% |
| Average |  |  |  |  |  |  |  | 82.18% |

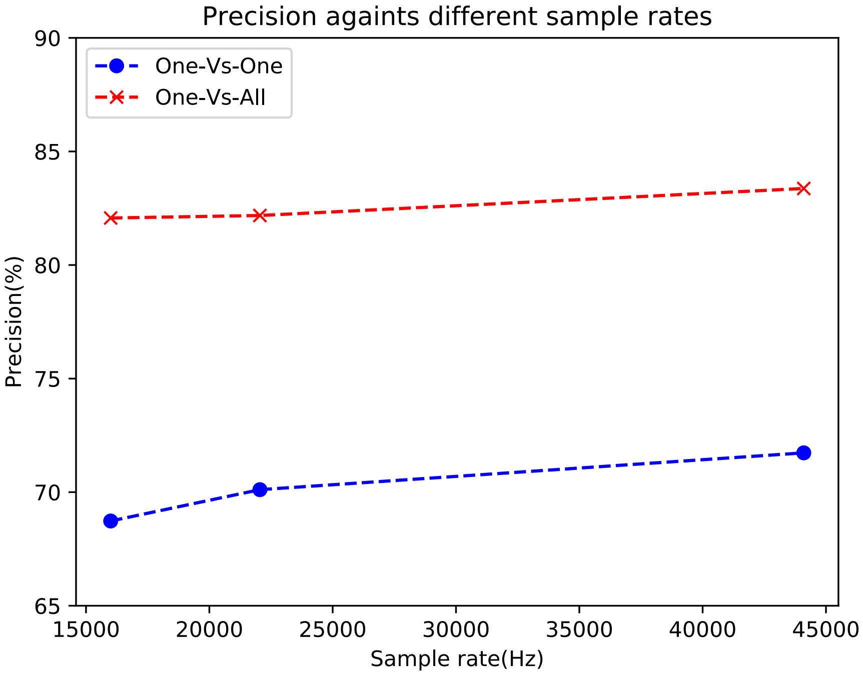


Figure 5.6: Variation of precision value against sample rates

### 5.3.5 Evaluate by Removing Silence

In the data pre-processing phase, silence removal is done only to news and conversation contents because they have very long silence periods within 5s of frames as shown in Figure 3.7. For the evaluating purposes, the model is evaluated without silence removal and with silence removal from all data. Table 5.15 and Table 5.16 shows the confusion matrices of One-Vs-One model without silence removal and after silence removal from all data. Table 5.17 and Table 5.18 shows the confusion matrices of One-Vs-All model without silence removal and with silence removal from all data.

Table 5.15: Confusion matrix of One-Vs-One model without silence removal

| | | **Predicted Class** | | | | | Support | Precision(%) |
|---|---|---|---|---|---|---|---|---|
| | | **news** | **conv** | **advert** | **drama** | **relProg** | | |
| **True Class** | **news** | **257** | 10 | 41 | 9 | 0 | 317 | 79% |
| | **conv** | 29 | **246** | 15 | 32 | 3 | 325 | 82% |
| | **advert** | 28 | 12 | **243** | 21 | 3 | 307 | 80% |
| | **drama** | 7 | 14 | 6 | **289** | 0 | 316 | 82% |
| | **relProg** | 0 | 8 | 0 | 1 | **314** | 323 | 98% |
| Average | | | | | | | | 85.07% |

Table 5.16: Confusion matrix of One-Vs-One model with silence removal

| | | **Predicted Class** | | | | | Support | Precision(%) |
|---|---|---|---|---|---|---|---|---|
| | | **news** | **conv** | **advert** | **drama** | **relProg** | | |
| **True Class** | **news** | **272** | 8 | 29 | 8 | 0 | 317 | 81% |
| | **conv** | 27 | **248** | 15 | 32 | 3 | 325 | 85% |
| | **advert** | 38 | 12 | **232** | 22 | 3 | 307 | 80% |
| | **drama** | 7 | 15 | 6 | **288** | 0 | 316 | 82% |
| | **relProg** | 0 | 8 | 0 | 1 | **314** | 323 | 98% |
| Average | | | | | | | | 85.23% |

Table 5.17: Confusion matrix of One-Vs-All model without silence removal

| | | **Predicted Class** | | | | | Support | Precision(%) |
|---|---|---|---|---|---|---|---|---|
| | | **news** | **conv** | **advert** | **drama** | **relProg** | | |
| **True Class** | **news** | **256** | 11 | 43 | 7 | 0 | 317 | 80% |
| | **conv** | 31 | **255** | 5 | 31 | 3 | 325 | 77% |
| | **advert** | 35 | 13 | **223** | 35 | 1 | 307 | 72% |
| | **drama** | 10 | 18 | 5 | **283** | 0 | 316 | 89% |
| | **relProg** | 1 | 8 | 0 | 0 | **314** | 323 | 97% |
| Average | | | | | | | | 83.2% |

Table 5.18: Confusion matrix of One-Vs-All model with silence removal

| | | **Predicted Class** | | | | | Support | Precision(%) |
|---|---|---|---|---|---|---|---|---|
| | | **news** | **conv** | **advert** | **drama** | **relProg** | | |
| **True Class** | **news** | **265** | 13 | 32 | 7 | 0 | 317 | 83% |
| | **conv** | 33 | **253** | 4 | 33 | 2 | 325 | 78% |
| | **advert** | 48 | 11 | **193** | 52 | 3 | 307 | 62% |
| | **drama** | 15 | 12 | 5 | **282** | 2 | 316 | 89% |
| | **relProg** | 1 | 7 | 0 | 0 | **315** | 323 | 97% |
| Average | | | | | | | | 81.6% |

Figure 5.7 and Figure 5.8 show the changing of the performances of the classification without silence removal and with silence removal in One-Vs-One model and One-Vs-All model respectively.

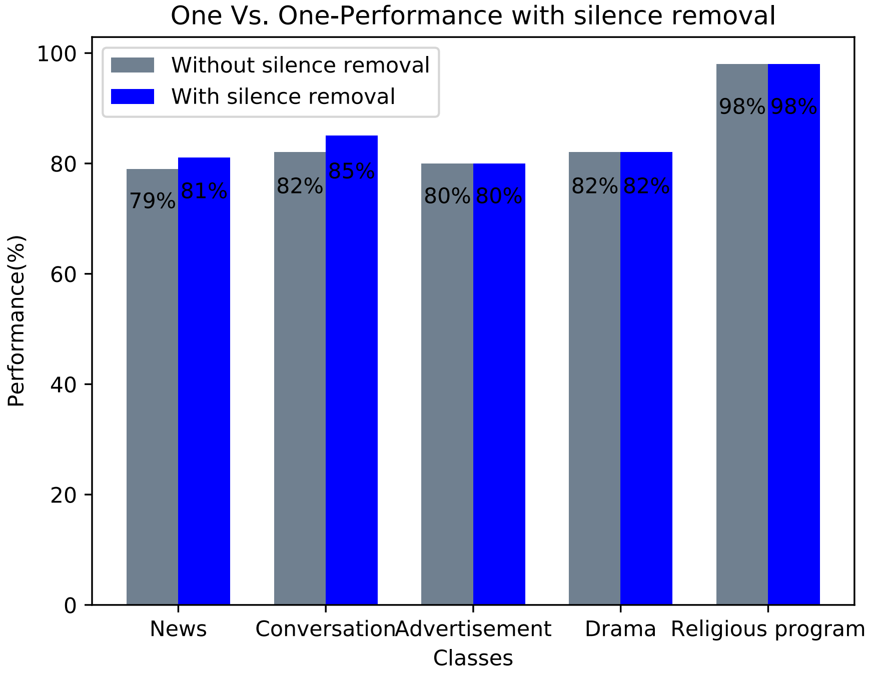


Figure 5.7: Variation of the precision value of One-Vs-One against Silence removal

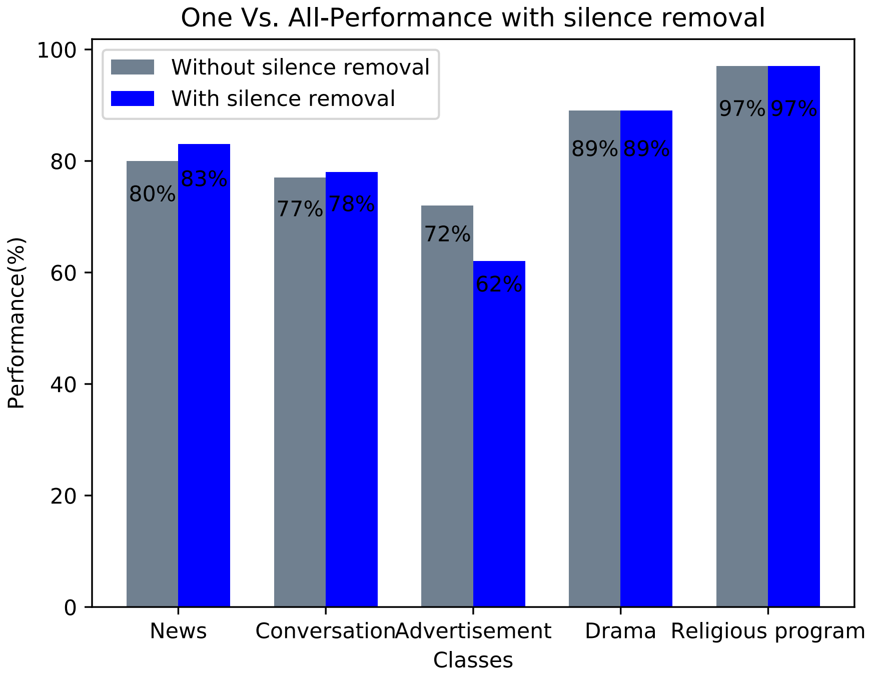


Figure 5.8: Variation of the precision value of One-Vs-All against Silence removal

## 5.4 Discussion

Two multi-class SVM models are evaluated under different conditions, such as increasing features, changing frame sizes, sample rates, K values and silencing. Increasing the features one by one helps to obtain the best subset of features out of the selected features. This method reduces unnecessary dimensions and makes the process faster.

As found in the literature, mostly used frame sizes are 25 ms, 0.25 s and 2.5 s. Unfortunately, in our case non of these are able to achieve satisfiable performance. Therefore, we decided to increase frame lengths to 3s, 4s, and 5s. As a result of that, 5s frame size performed better than the others as shown in Figure 5.5. Therefore, 5s is chosen as the frame size. According to Figure 5.6, 44100 Hz is selected as the sample rate. Table 5.4 depicts the best value for K in K-fold cross-validation as 10. As shown in Figure Figure 5.7 and Figure 5.8, after the removal of the silence from all data, performances increased only in news and conversations. Therefore, we decided to remove silence only from news and conversations. After applying all these justifiable values, the performance of One-Vs-One and One-Vs-All models show respectively in Table 5.19 and Table 5.20 including the accuracies of their binary classifiers.

Table 5.19: Overall results of One-Vs-One model

| **Models** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
|---|---|---|---|---|
| **news/<br>advertisement** | 87% | 87% | 87% | 87% |
| **news/<br>conversation** | 90% | 92% | 91% | 91% |
| **news/<br>drama** | 92% | 92% | 92% | 92% |
| **news/<br>religious program** | 99% | 100% | 100% | 100% |
| **advertisement/<br>conversation** | 91% | 92% | 92% | 92% |
| **advertisement/<br>drama** | 94% | 93% | 94% | 94% |
| **advertisement/<br>religious program** | 99% | 99% | 99% | 99% |
| **conversation/<br>drama** | 85% | 87% | 86% | 87% |
| **conversation/<br>religious program** | 95% | 95% | 95% | 95% |
| **drama/<br>religious program** | 99% | 99% | 99% | 91% |
| **One Vs. One** | **85%** | **85%** | **85%** | **85%** |

As included in Table 5.19, even though ten binary SVMs trained with high training accuracies, after combining the ten models together the accuracy of the alliance becomes lesser. The reason might be the “Max Winning” strategy that uses to predict the classes in One-Vs-One. In “Max winning” strategy when computing the mode, if the maximum number of votes is equal to two classes, then it outputs only one class which appears first in the array. This problem might be a minor effect on the downgrade the final result of One-Vs-One SVM. In our testing dataset, approximately $13\%$ proportion of dataset face this issue when the frame length is 0.25s. But when we increase the frame length, the proportion of identical classes decreased to $4\%$ as shown in Figure 5.9,

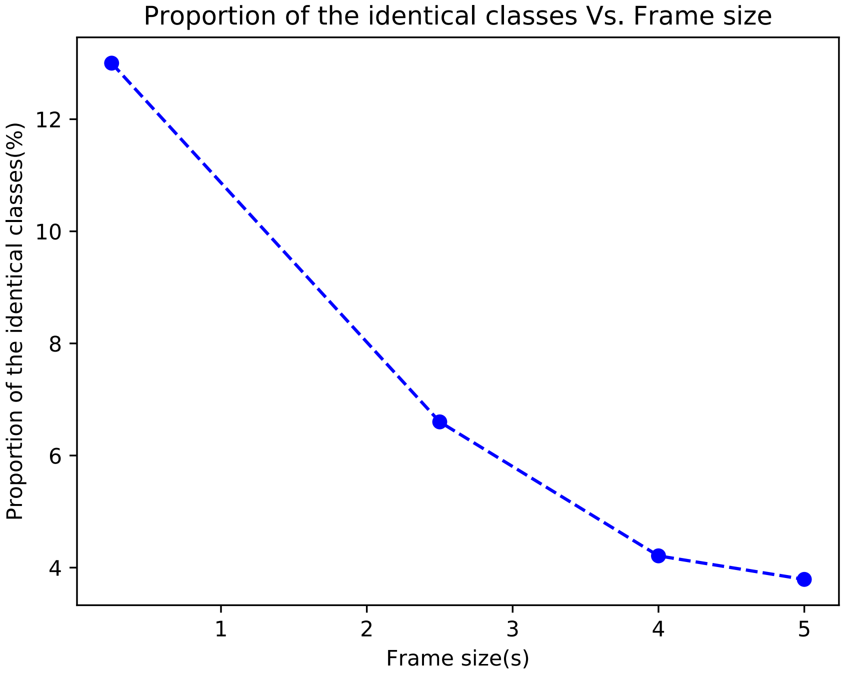


Figure 5.9: Proportion of the identical classes in ”Max Winning” strategy against the frame length

As depicted on Table 5.20, One-Vs-All model shows less accuracy than One-Vs-One model. The disadvantage of using this method is that it takes high training time because each binary classifier individually requires the entire data set for the training. But the binary classifiers in One-Vs-One model requires data only from two classes for training.

When analyzing features for One-Vs-One model all the features are compared against two classes as shown in Figure 3.4. But in One-Vs-All model, the features should be observed one class against four classes as shown in Figure 3.6. Therefore, another disadvantage of One-Vs-All model is that when analyzing features to select the best, it is difficult to observe distinguishable

Table 5.20: Overall results of One-Vs-All model

| **Models** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
|---|---|---|---|---|
| **news/<br>other** | 86% | 82% | 75% | 78% |
| **conversation/<br>other** | 91% | 92% | 78% | 84% |
| **advertisement/<br>other** | 91% | 91% | 79% | 85% |
| **drama/<br>other** | 88% | 82% | 81% | 82% |
| **religious program/<br>other** | 98% | 96% | 97% | 97% |
| **One Vs. All** | **83%** | **83%** | **83%** | **83%** |

features for one class against four classes.

According to Table 5.19 and Table 5.20, the One-Vs-One model achieves 85% of overall precision, and the One-Vs-All model achieves 83% of overall precision. The obtained results of this study guide us to find the most suitable multi-class SVM for this problem domain. According to the performance of these two modes, the One-Vs-One model with 85% of precision, is chosen as the most appropriate model for this research.

Table 5.6 shows the confusion matrix of One-Vs-One model after applying the necessary refinements. When examining the confusion matrix, it is visible that the majority of news frames misclassifies as advertisements but not as religious programs. Conversations misclassified as news, advertisements, and drama in a majority. A considerable amount of advertisements misidentified as news and drama. Religious programs perform well with 98% of precision value and it does not misclassify as news and advertisements. Religious programs can be classified accurately from other classes because of the unique pattern of Paritta-Sutta chants. The misclassifications of news as advertisements and vice versa, conversations as news and drama, and advertisements as drama can be happened because of the static nature of these content categories. In some cases, the meaning the of news, conversations, advertisements without jingles, and radio drama help us to distinguish these classes. Therefore, it is pretty obvious to have these misclassifications because of a supervised learning model unable to interpret the meaning of the input data.

## 5.5 Summary

This chapter reviews the evaluation results and findings with regard to two multi-class SVM models for classification of voice dominant content categories in radio broadcasting. Experiments have been carried out to validate the feature set, K value for cross-validation, frame size, sample rate, and validate the model with silence removal and without silence removal. According to the results, the frame size of length 5s and sample rate of 44100 Hz are chosen with silence removing of news and conversation. For the training processes, 10-Fold cross-validation is applied. By evaluating the results of the two models, the One-Vs-One model is selected as the best classifier. The One-Vs-One model successfully classifies the pre-defined content categories with the accuracies of $78\%$ for news, $85\%$ for conversations, $83\%$ for advertisements, $82\%$ for drama and $98\%$ for religious programs. The overall accuracy of the model is $85\%$.

# Chapter 6

# Conclusions

## 6.1 Introduction

This chapter reviews the conclusions formed after completion of the research. Section 6.2 concludes the research questions stated in Section 1.2 followed by the aim and objectives. In Section 6.3, the contribution of the dissertation for voice dominant content classification in public radio broadcasting context in Sri Lanka is presented. The subsequent sections will review the limitations of this research apparent during the progress of the research as well as the implications for further research.

## 6.2 Conclusions about Research Questions

The main aim of this research is to identify voice dominant content categories to automate the radio broadcasting context in Sri Lanka. For that, a multi-class SVM was proposed to classify five voice dominant content categories in a radio broadcasting context. Two different methods were used to build the multi-class SVM model. They are “One Vs. One” multi-class SVM and “One Vs. All” multi-class SVM.

The novelty of this approach is that instead of feeding all the features together to the model, only selected features were fed separately to each classifier in the model. Hence, before implementing the classification models, the best subset of features for each classifier was identified individually. In One-Vs-One model, always the features were compared against two classes. But in One-Vs-All model, the features were compared one class against four classes. Therefore, identifying features for One-Vs-All model was a bit challenging than identifying features for One-Vs-One model. However, the best features for each classifier were selected as depicted in Table 5.1.

The performance of these two models was evaluated under different criteria. One-Vs-One model successfully classified the pre-defined content categories with the accuracies of 81% for news, 85% for conversations, 80% for advertisements, 82% for drama and 98% for religious programs. The One-Vs-All model successfully classified the categories with the accuracies of 73% for news, 85% for conversations, 80% for advertisements, 76% for drama and 98% for religious programs. The final accuracies of the One-Vs-One and One-Vs-All models are 85% and 83% respectively. Therefore, the One-Vs-One model was chosen as the most appropriate model for this research. When considering the results of both models, a considerable amount of news frames misclassified as advertisements, while the conversations and advertisements misclassified as news and drama. The religious programs performed well than others. The static nature of news, conversations, advertisements, and drama might be the reason for these misclassifications.

## 6.3 Conclusions about Research Problem

As discussed in Chapter 1, having an automated monitoring process for radio broadcasting context in Sri Lanka is the identified requirement. For that, establish an accurate classification method is essential to automate the monitoring process. According to the background review, even though there are classifiers to classify radio broadcasting content, they are not able to classify voice dominant classes in broadcasting content. Hence, this research contributes to identifying a classification model which is able to distinguish the voice dominant content classes in Sri Lankan radio broadcasting context. The proposed classification model is capable to classify news, conversations, advertisements without jingles, radio drama, and religious programs in radio broadcasting contents.

## 6.4 Limitations and Implications for Further Research

The major limitation of this research is that the model is trained and tested only for 'SLBC' radio FM channel. However, this study creates a platform to further generalize this model to all Sri Lankan FM channels. Another limitation is that restricts the data from each class to 1 hour and 10 minutes. The reason is that religious programs were unable to provide data, more than 1 hour and 10 minutes long. Therefore, the data of rest of the classes were also limited nearly to 1 hour and 10 minutes to avoid the proportional bias in the dataset. Therefore, use more data to train the model might be a good choice to increase the performance.

When annotating data, it is very hard to segment data to the point. Occasionally, irrelevant classes have been hit on the boundaries of cutting segments in data. Therefore, the data annotation process was revised again to create a more accurate dataset. Hence, follow a well-structured annotation process is needed to improve the performance.

Altogether, 34 features which belong to time domain features, frequency domain features, cepstral features, and chroma features were observed in this research. In addition to that, identification of more distinguishable features can be lead the model to more accurate results.

# 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] C. Weerathunga, "Classification of public radio broadcast context for onset detection," Master's thesis, 2017.

[3] J. Saunders, "Real-time discrimination of broadcast speech/music," in *icassp*. IEEE, 1996, pp. 993–996.

[4] R. Barzilay, M. Collins, J. Hirschberg, and S. Whittaker, "The rules behind roles: Identifying speaker role in radio broadcasts," in *AAAI/IAAI*, 2000, pp. 679–684.

[5] Y. Liu, "Initial study on automatic identification of speaker role in broadcast news speech," in *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. Association for Computational Linguistics, 2006, pp. 81–84.

[6] 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.

[7] J. Vavrek, E. Vozáriková, M. Pleva, and J. Juhár, "Broadcast news audio classification using svm binary trees," in *Telecommunications and Signal Processing (TSP), 2012 35th International Conference on*. IEEE, 2012, pp. 469–473.

[8] L. Lu, H. Jiang, and H. Zhang, "A robust audio classification and segmentation method," in *Proceedings of the ninth ACM international conference on Multimedia*. ACM, 2001, pp. 203–211.

[9] M. Khan, W. G. Al-Khatib, and M. Moinuddin, "Automatic classification of speech and music using neural networks," in *Proceedings of the 2nd ACM international workshop on Multimedia databases*. ACM, 2004, pp. 94–99.

[10] 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.

[11] Z. Kons, O. Toledo Ronen, and M. Carmel, “Audio event classification using deep neural networks.” in *Interspeech*, 2013, pp. 1482–1486.

[12] C. W. Hsu and C. J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[13] C. Lin, “A comparison of methods for multi-class support vector machines,” *IEEE Transaction onNeural Networks13 (2)*, pp. 415–425, 2002.

[14] F. Aurino, M. Folla, F. Gargiulo, V. Moscato, A. Picariello, and C. Sansone, “One-class svm based approach for detecting anomalous audio events,” in *Intelligent Networking and Collaborative Systems (INCoS), 2014 International Conference on*. IEEE, 2014, pp. 145–151.

[15] A. Bouril, D. Aleinikava, M. S. Guillem, and G. M. Mirsky, “Automated classification of normal and abnormal heart sounds using support vector machines,” in *Computing in Cardiology Conference (CinC), 2016*. IEEE, 2016, pp. 549–552.

[16] S. E. Küçükbay and M. Sert, “Audio-based event detection in office live environments using optimized mfcc-svm approach,” in *Semantic Computing (ICSC), 2015 IEEE International Conference on*. IEEE, 2015, pp. 475–480.

[17] I. Martín-Morató, M. Cobos, and F. J. Ferri, “A case study on feature sensitivity for audio event classification using support vector machines,” in *Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th International Workshop on*. IEEE, 2016, pp. 1–6.

[18] J. C. Wang, J. F. Wang, C. B. Lin, K.-T. Jian, and W. Kuok, “Content-based audio classification using support vector machines and independent component analysis,” in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 4. IEEE, 2006, pp. 157–160.

[19] L. Lu, S. Z. Li, and H. J. Zhang, “Content-based audio segmentation using support vector machines,” in *Proc. ICME*, vol. 1, 2001, pp. 749–752.

[20] Y. Zhu, Z. Ming, and Q. Huang, “Automatic audio genre classification based on support vector machine,” in *Natural Computation, 2007. ICNC 2007. Third International Conference on*, vol. 1. IEEE, 2007, pp. 517–521.

[21] B. Kijsirikul and N. Ussivakul, “Multiclass support vector machines using adaptive directed acyclic graph,” in *Neural Networks, 2002. IJCNN’02. Proceedings of the 2002 International Joint Conference on*, vol. 1. IEEE, 2002, pp. 980–985.

[22] “Sample rates - audacity manual,” https://manual.audacityteam.org/man/sample_rates.html, (Accessed on 12/22/2018).

[23] T. Giannakopoulos, “pyaudioanalysis: An open-source python library for audio signal analysis,” *PloS one*, vol. 10, no. 12, p. e0144610, 2015.

# Appendices

# Appendix A

# Code Listings

Listing A.1: Constants

```
k = 10
sample_rate = 44100
frame_size = 5
```

Listing A.2: Feature Comparison

```
import os
import numpy as np
import librosa as lb
import pyAudioAnalysis.audioFeatureExtraction as fe
import matplotlib.pyplot as plt
import constants as c
import seaborn as sns
sns.set()

path = ”mixClips”
save_path = ”feature comparison”
fs = c.frame_size
sr = c.sample_rate
global time
clips = [”news_ads.wav”, ”news_conv.wav”, ”news_rd.wav”, ”news_rp.wav”,
         ”conv_ads.wav”, ”ads_rd.wav”, ”ads_rp.wav”, ”conv_rd.wav”,
         ”conv_rp.wav”, ”rd_rp.wav”]

def plot_graphs(features, f_names1):
    global time
    extract_vec = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
            16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
            30, 31, 32, 33]
    extracted_features = []
    f_names = []
    for i in extract_vec:
        if isinstance(extracted_features, list):
            extracted_features = np.array([features[i]])
            f_names = np.array([f_names1[i]])
        else:
            extracted_features = np.append(extracted_features,
                [features[i]], axis=0)
            f_names = np.append(f_names, [f_names1[i]], axis=0)

    f1 = extracted_features

    for i in range(len(f1)):
        fig = plt.figure()
        ax = plt.subplot(111)
        time = np.arange(fs, (len(f1[i])+1) * fs, fs)
        ax.plot(time, f1[i])
        plt.xlabel(”time”)
        plt.title(f_names[i])
        img = os.path.join(save_path, f_names[i])
        fig.savefig(img)

# read the wave files, extract features
for i in clips:
    clip1 = os.path.join(path, i)
    x1, sr1 = lb.load(clip1, mono=True, sr=44100)
    F1, f_names1 = fe.stFeatureExtraction(x1, sr1, fs * sr1, fs * sr1, ”news”)
    plot_graphs(F1, f_names1)
```

Listing A.3: Binary SVM Construction for Classify News and Advertisement

```
import numpy as np
import sklearn.svm as svm
import sklearn.model_selection as cv
import constants as c
import seaborn as sns
sns.set()
```

```python

def news_ads(feature_vec):
    features = np.transpose(feature_vec)
    extract_vec = [2, 11, 15, 16, 22, 30, 31, 34]
    extracted_features = []
    for i in extract_vec:
        if isinstance(extracted_features, list):
            extracted_features = np.array([features[i]])
        else:
            extracted_features = np.append(extracted_features,
                [features[i]], axis=0)

    feature_vecT = np.transpose(extracted_features)
    [rows, cols] = feature_vecT.shape

    # create X and Y data
    X = feature_vecT[:, :cols-1]
    Y = feature_vecT[:, cols-1]
    Y = Y.astype(int)


    # 70% training and 30% testing
    x_train, x_test, y_train, y_test = cv.train_test_split(X, Y, test_size
        =0.3)

    # define SVM classifier
    svm_model1 = svm.SVC(kernel='rbf', probability=True)

    kf = cv.KFold(n_splits=c.k, random_state=None)

    for train_indices, test_indices in kf.split(x_train):
        svm_model1.fit(x_train[train_indices], y_train[train_indices])\
            .score(x_train[test_indices], y_train[test_indices])

    y_predict = svm_model1.predict(x_test)
```

Listing A.4: Train One Vs. One Model

```python
import os
import librosa as lb
import numpy as np
import math
import pyAudioAnalysis.audioFeatureExtraction as fe
import constants as c
import seaborn as sns
sns.set()

sample_rate = c.sample_rate
frame_duration = c.frame_size
win = int(math.floor(frame_duration*sample_rate))

def remove_silence(x):
    intervals = lb.effects.split(x, top_db=40, frame_length=
        c.frame_size, hop_length=32)
    for ind in range(len(intervals)):
        start_ind, end_ind = intervals[ind][0], intervals[ind][1]
        if start_ind < len(x):
            abs_seg = abs(x[start_ind:end_ind]).mean()
            abs_audio = abs(x).mean()
            if abs_seg < abs_audio - 0.05:
                x[start_ind:end_ind] = 0
    return x

def feature_extraction(category, feature_category, cls):
    for f in os.listdir(category):
        if os.path.isfile(os.path.join(category, f)):
            wav_path = os.path.join(category, f)
            x, sr = lb.load(wav_path, mono=True, sr=44100)

            if cls == ”news” or cls == ”conversations”:
                first = 0
                interval = lb.effects.split(x)
                xt = np.zeros_like(x)
                for ind, (start, end) in enumerate(interval[first:]):
                    xt[start:end] = remove_silence(x[start:end])
                x = xt

            F, f_names = fe.stFeatureExtraction(x, sample_rate, win, win,
                cls)
            if isinstance(feature_category, list):
                feature_category = np.array(F)
            else:
                feature_category = np.concatenate((feature_category, F), axis=1)
    return feature_category

def shuffle(vec1, vec2, cls1, cls2,  name):
    [row1, col1] = vec1.shape
    [row2, col2] = vec2.shape

    feature_concat = np.concatenate((vec1, vec2), axis=1)
    feature_vec = np.transpose(feature_concat)

    # labeling the files and create feature vector
    if cls1 == 1:
        cls_1 = np.zeros((1, col1), dtype=int)
    elif cls1 == 2:
        cls_1 = np.ones((1, col1), dtype=int)
    elif cls1 == 4:
        cls_1 = np.full((1, col1), 3)
    elif cls1 == 5:
        cls_1 = np.full((1, col1), 4)

    if cls2 == 2:
        cls_2 = np.ones((1, col2), dtype=int)
    elif cls2 == 3:
        cls_2 = np.full((1, col2), 2)
    elif cls2 == 4:
        cls_2 = np.full((1, col2), 3)
    elif cls2 == 5:
        cls_2 = np.full((1, col2), 4)

    cls_all = np.concatenate((cls_1, cls_2), axis=1)
    cls = np.transpose(cls_all)

    feature_vec_trans = np.concatenate((feature_vec, cls), axis=1)
    np.random.shuffle(feature_vec_trans)
    return feature_vec_trans

def train_data():
    news = ”sampleData\\news”
    conv = ”sampleData\\conversations”
    ads = ”sampleData\\advertisements”
    rd = ”sampleData\\radioDramas”
    rp = ”sampleData\\ReligiousProgs”

    # read the wave files, frame blocking
    feature_news = []  # class 1
    feature_conv = []  # class 2
    feature_ads = []  # class 3
    feature_rd = []  # class 4
    feature_rp = []  # class 5

    feature_news = feature_extraction(news,
        feature_news , ”news”)
    feature_conv = feature_extraction(conv,
        feature_conv, ”conversations”)
    feature_ads = feature_extraction(ads,
        feature_ads, ”advertisements”)
    feature_rd = feature_extraction(rd,
        feature_rd, ”radioDrama”)
    feature_rp = feature_extraction(rp,
        feature_rp, ”ReligiousProgs”)

    feature_news_ads = shuffle(feature_news, feature_ads,
        1, 3, ”dataCSV\\news_ads.csv”)
    feature_news_conv = shuffle(feature_news, feature_conv,
        1, 2,  ”dataCSV\\news_conv.csv”)
    feature_news_rd = shuffle(feature_news, feature_rd,
        1, 4, ”dataCSV\\news_rd.csv”)
    feature_news_rp = shuffle(feature_news, feature_rp,
        1, 5, ”dataCSV\\news_rp.csv”)
    feature_conv_ads = shuffle(feature_conv, feature_ads,
        2, 3, ”dataCSV\\conv_ads.csv”)
    feature_conv_rd = shuffle(feature_conv, feature_rd,
        2, 4, ”dataCSV\\conv_rd.csv”)
    feature_conv_rp = shuffle(feature_conv, feature_rp,
        2, 5, ”dataCSV\\conv_rp.csv”)
    feature_rd_ads = shuffle(feature_rd, feature_ads,
        4, 3, ”dataCSV\\rd_ads.csv”)
    feature_rd_rp = shuffle(feature_rd, feature_rp,
        4, 5, ”dataCSV\\rd_rp.csv”)
    feature_rp_ads = shuffle(feature_rp, feature_ads,
        5, 3, ”dataCSV\\rp_ads.csv”)

    svm1 = model1.news_ads(feature_news_ads)
    svm2 = model2.news_conv(feature_news_conv)
    svm3 = model3.news_rd(feature_news_rd)
    svm4 = model4.news_rp(feature_news_rp)
    svm5 = model5.conv_ads(feature_conv_ads)
    svm6 = model6.rd_ads(feature_rd_ads)
    svm7 = model7.rp_ads(feature_rp_ads)
    svm8 = model8.conv_rd(feature_conv_rd)
    svm9 = model9.conv_rp(feature_conv_rp)
    svm10 = model10.rd_rp(feature_rd_rp)
```

Listing A.5: Test One Vs. One Model

```
# selected features for each model
feature_news_ads = [2, 11, 15, 16, 22, 30, 31]
feature_news_conv = [3, 5, 7, 9, 11, 12, 17]
feature_news_rd = [1, 3, 4, 11, 13, 16, 19]
feature_news_rp = [1, 3, 4, 10, 16, 18]
feature_conv_ads = [3, 5, 6, 7, 9, 12, 17]
feature_rd_ads = [2, 3, 4, 5, 11, 14, 15]
feature_rp_ads = [1, 3, 4, 10, 14, 16, 17]
feature_conv_rd = [5, 7, 11, 15, 19, 20]
feature_conv_rp = [1, 2, 4, 8, 10, 11, 16, 18]
feature_rd_rp = [0, 1, 2, 3, 4, 10, 15, 18, 20]

def shuffle_all(vec1, vec2, vec3, vec4, vec5, name):
    [row1, col1] = vec1.shape
    [row2, col2] = vec2.shape
    [row3, col3] = vec3.shape
    [row4, col4] = vec4.shape
    [row5, col5] = vec5.shape

    feature_concat = np.concatenate((vec1, vec2, vec3, vec4, vec5), axis=1)
    feature_vec = np.transpose(feature_concat)

    # labeling the files and create feature vector
    cls_1 = np.zeros((1, col1), dtype=int)
    cls_2 = np.ones((1, col2), dtype=int)
    cls_3 = np.full((1, col3), 2)
    cls_4 = np.full((1, col4), 3)
    cls_5 = np.full((1, col5), 4)

    cls_all = np.concatenate((cls_1, cls_2, cls_3, cls_4, cls_5), axis=1)
    cls = np.transpose(cls_all)

    feature_vec_trans = np.concatenate((feature_vec, cls), axis=1)
    np.random.shuffle(feature_vec_trans)

    # write to a csv file
    with open(name, 'w', newline='') as csvFile:
        writer = csv.writer(csvFile)
        writer.writerows(feature_vec_trans)

    csvFile.close()

    return feature_vec_trans

def compute_mode(numbers):
    mode = []
    counts = Counter(numbers)
    maxcount = max(counts.values())
    for num,count in counts.items():
        if count == maxcount:
            mode.append(num)
    return mode

def max_win(data, cls):
    global identical, more_cls
    prob = []

    data1 = np.array([[data[i] for i in feature_news_ads]])
    y_predict = np.array(md1.predict(data1))
    prob.append(max(md1.predict_proba(data1)[0]))

    data2 = np.array([[data[i] for i in feature_news_conv]])
    y_predict = np.append(y_predict, (md2.predict(data2)), axis=0)
    prob.append(max(md2.predict_proba(data2)[0]))
    data3 = np.array([[data[i] for i in feature_news_rd]])
    y_predict = np.append(y_predict, (md3.predict(data3)), axis=0)
    prob.append(max(md3.predict_proba(data3)[0]))

    data4 = np.array([[data[i] for i in feature_news_rp]])
    y_predict = np.append(y_predict, (md4.predict(data4)), axis=0)
    prob.append(max(md4.predict_proba(data4)[0]))

    data5 = np.array([[data[i] for i in feature_conv_ads]])
    y_predict = np.append(y_predict, (md5.predict(data5)), axis=0)
    prob.append(max(md5.predict_proba(data5)[0]))

    data6 = np.array([[data[i] for i in feature_rd_ads]])
    y_predict = np.append(y_predict, (md6.predict(data6)), axis=0)
    prob.append(max(md6.predict_proba(data6)[0]))

    data7 = np.array([[data[i] for i in feature_rp_ads]])
    y_predict = np.append(y_predict, (md7.predict(data7)), axis=0)
    prob.append(max(md7.predict_proba(data7)[0]))

    data8 = np.array([[data[i] for i in feature_conv_rd]])
    y_predict = np.append(y_predict, (md8.predict(data8)), axis=0)
    prob.append(max(md8.predict_proba(data8)[0]))

    data9 = np.array([[data[i] for i in feature_conv_rp]])
    y_predict = np.append(y_predict, (md9.predict(data9)), axis=0)
    prob.append(max(md9.predict_proba(data9)[0]))

    data10 = np.array([[data[i] for i in feature_rd_rp]])
    y_predict = np.append(y_predict, (md10.predict(data10)), axis=0)
    prob.append(max(md10.predict_proba(data10)[0]))

    max_val = compute_mode(y_predict)
    y = max_val[0]
    max_prob = []

    if len(max_val) > 1:
        identical += 1
        l = len(max_val)
        for j in range(1):
            ind = [i for i, n in enumerate(y_predict) if n == max_val[j]]
            p = sum([prob[i] for i in ind])
            max_prob.append(p)
        max_p = max(max_prob)
        y = max_val[max_prob.index(max_p)]
        if y!=cls:
            more_cls += 1

    return int(y)

# load the models from disk
md1 = pickle.load(open('svm1.sav', 'rb'))
md2 = pickle.load(open('svm2.sav', 'rb'))
md3 = pickle.load(open('svm3.sav', 'rb'))
md4 = pickle.load(open('svm4.sav', 'rb'))
md5 = pickle.load(open('svm5.sav', 'rb'))
md6 = pickle.load(open('svm6.sav', 'rb'))
md7 = pickle.load(open('svm7.sav', 'rb'))
md8 = pickle.load(open('svm8.sav', 'rb'))
md9 = pickle.load(open('svm9.sav', 'rb'))
md10 = pickle.load(open('svm10.sav', 'rb'))

news = ”testingData\\news”
conv = ”testingData\\conversations”
ads = ”testingData\\advertisements”
rd = ”testingData\\radioDramas”
rp = ”testingData\\ReligiousProgs”

# read the wave files, frame blocking
feature_news = [] # class 1
feature_conv = [] # class 2
feature_ads = [] # class 3
feature_rd = [] # class 4
feature_rp = [] # class 5

feature_news = feature_extraction(news, feature_news, ”news”)
feature_conv = feature_extraction(conv, feature_conv, ”conversations”)
feature_ads = feature_extraction(ads, feature_ads, ”advertisements”)
feature_rd = feature_extraction(rd, feature_rd, ”radioDrama”)
```

```
feature_rp = feature_extraction(rp, feature_rp, ”ReligiousProgs”)

feature_vector = shuffle_all(feature_news, feature_conv, feature_ads,
                  feature_rd, feature_rp, ”dataCSV\\testing_data.csv”)

[rows, cols] = feature_vector.shape

# create X and Y data
test_data = feature_vector[:, :cols - 1]
test_class = feature_vector[:, cols - 1]
test_class = test_class.astype(int)

predictions = []
for i in range(len(test_data)):
    predictions.append(max_win(test_data[i], test_class[i]))

# Model Accuracy
accuracy = metric.accuracy_score(test_class, predictions)*100
precision = metric.precision_score(test_class, predictions, average='macro')*100
recall = metric.recall_score(test_class, predictions, average='macro')*100
```

# Appendix B

# Diagrams

Below diagrams depict the feature patterns of selected features for each binary SVM in One Vs. One model.The diagram structure is as follows,

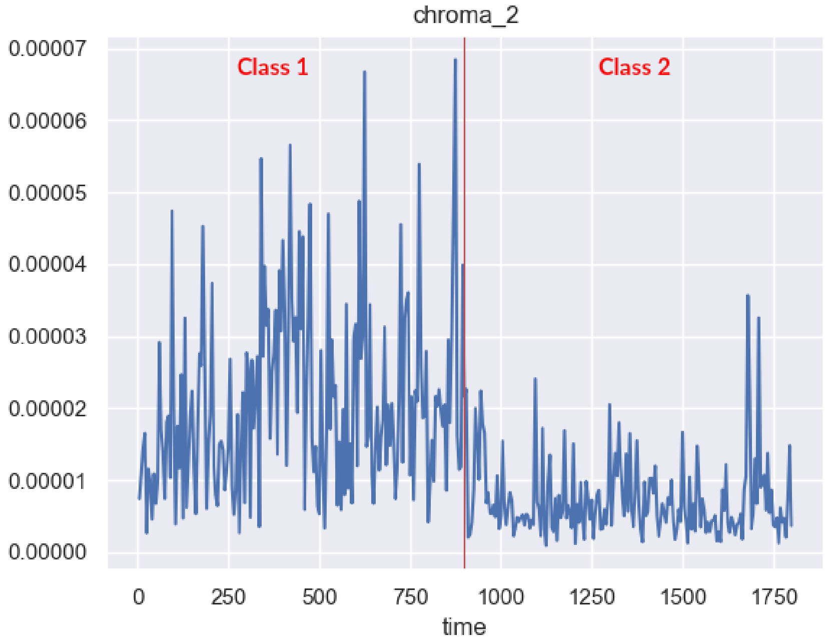
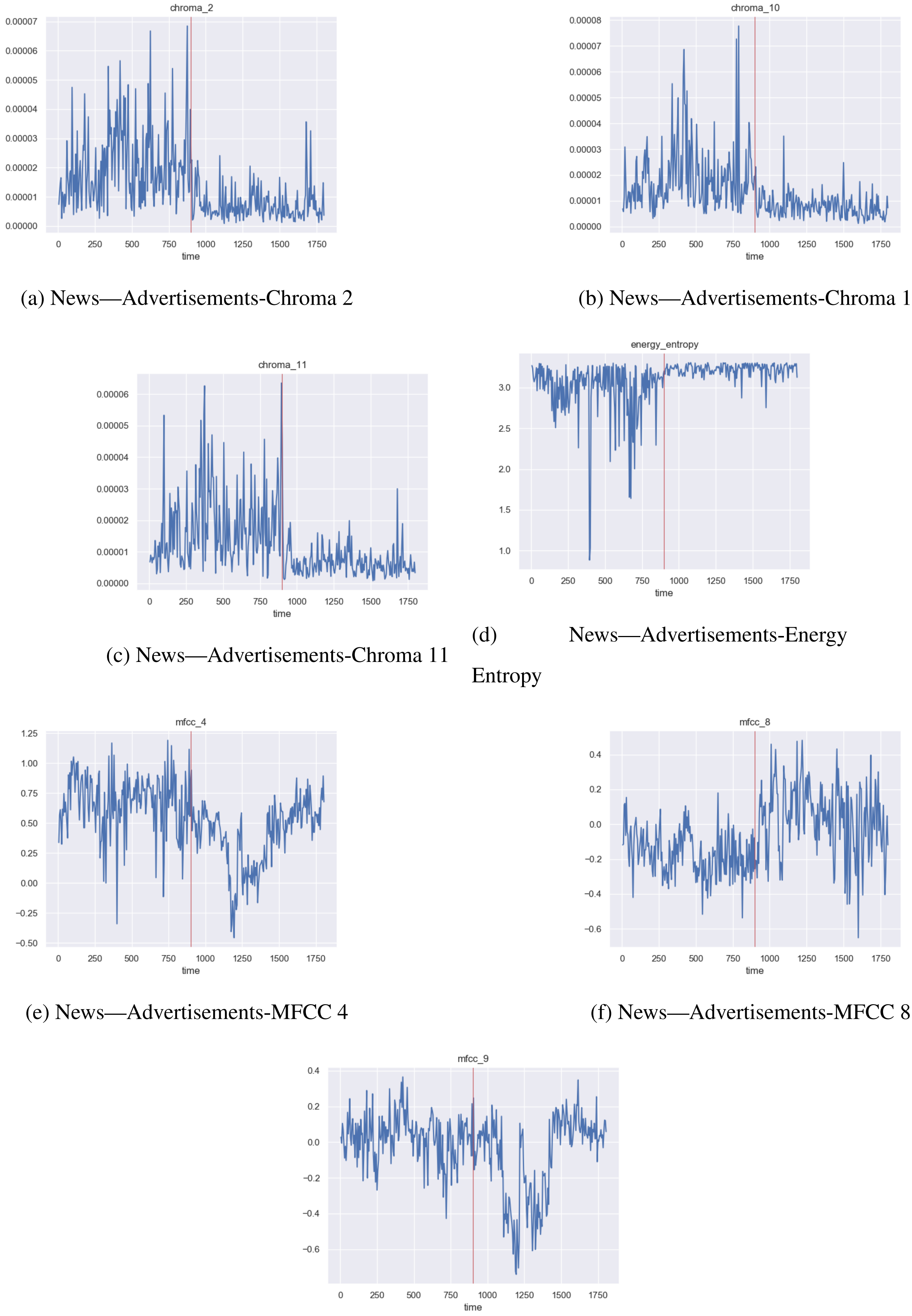


Figure B.1: Class 1—Class 2-Chroma 2



(a) News—Advertisements-Chroma 2

(b) News—Advertisements-Chroma 10

(c) News—Advertisements-Chroma 11

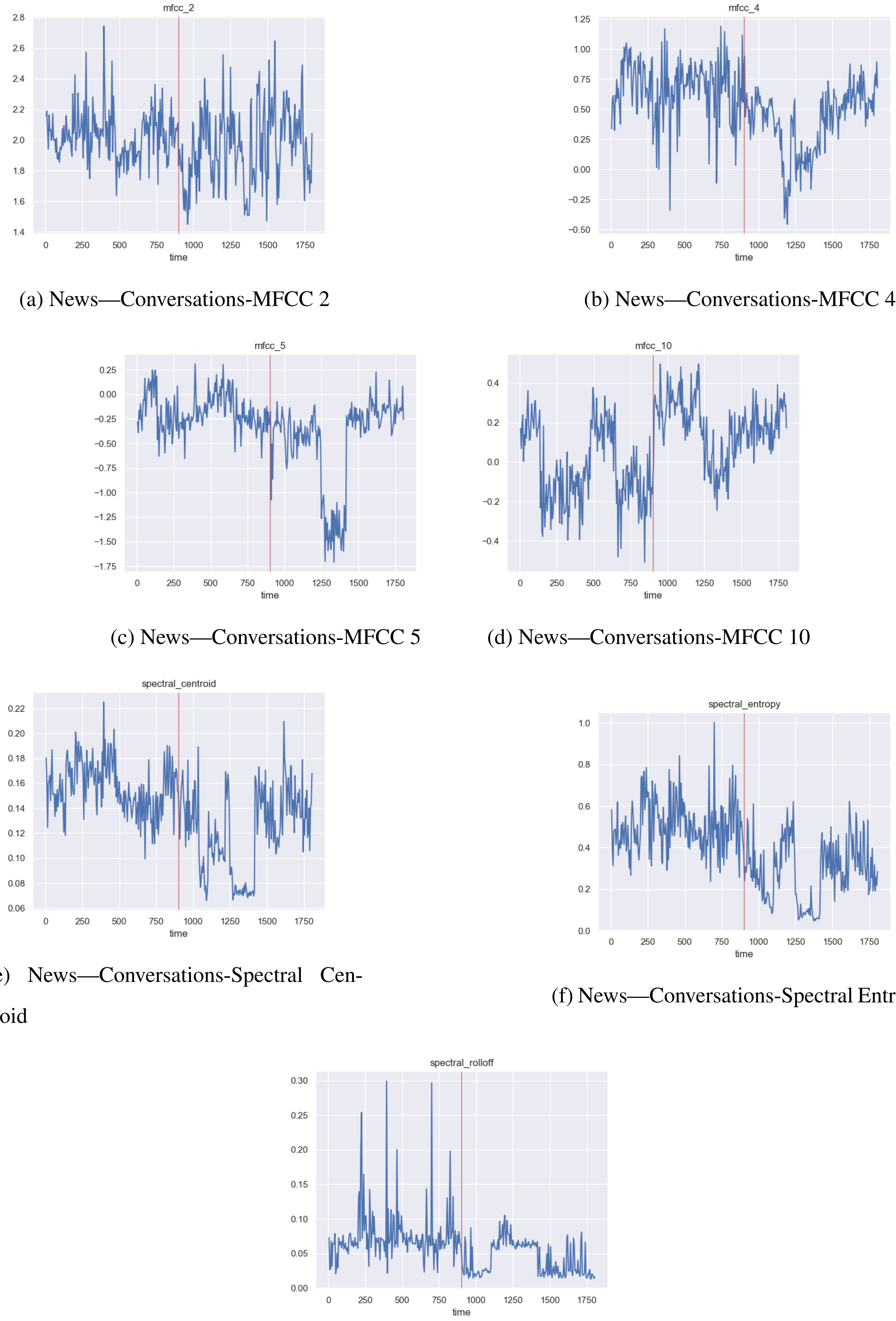
(d) News—Advertisements-Energy Entropy

(e) News—Advertisements-MFCC 4

(f) News—Advertisements-MFCC 8

(g) News—Advertisements-MFCC 9

Figure B.2: Feature patterns of selected features to distinguish news and advertisements



(a) News—Conversations-MFCC 2

(b) News—Conversations-MFCC 4

(c) News—Conversations-MFCC 5

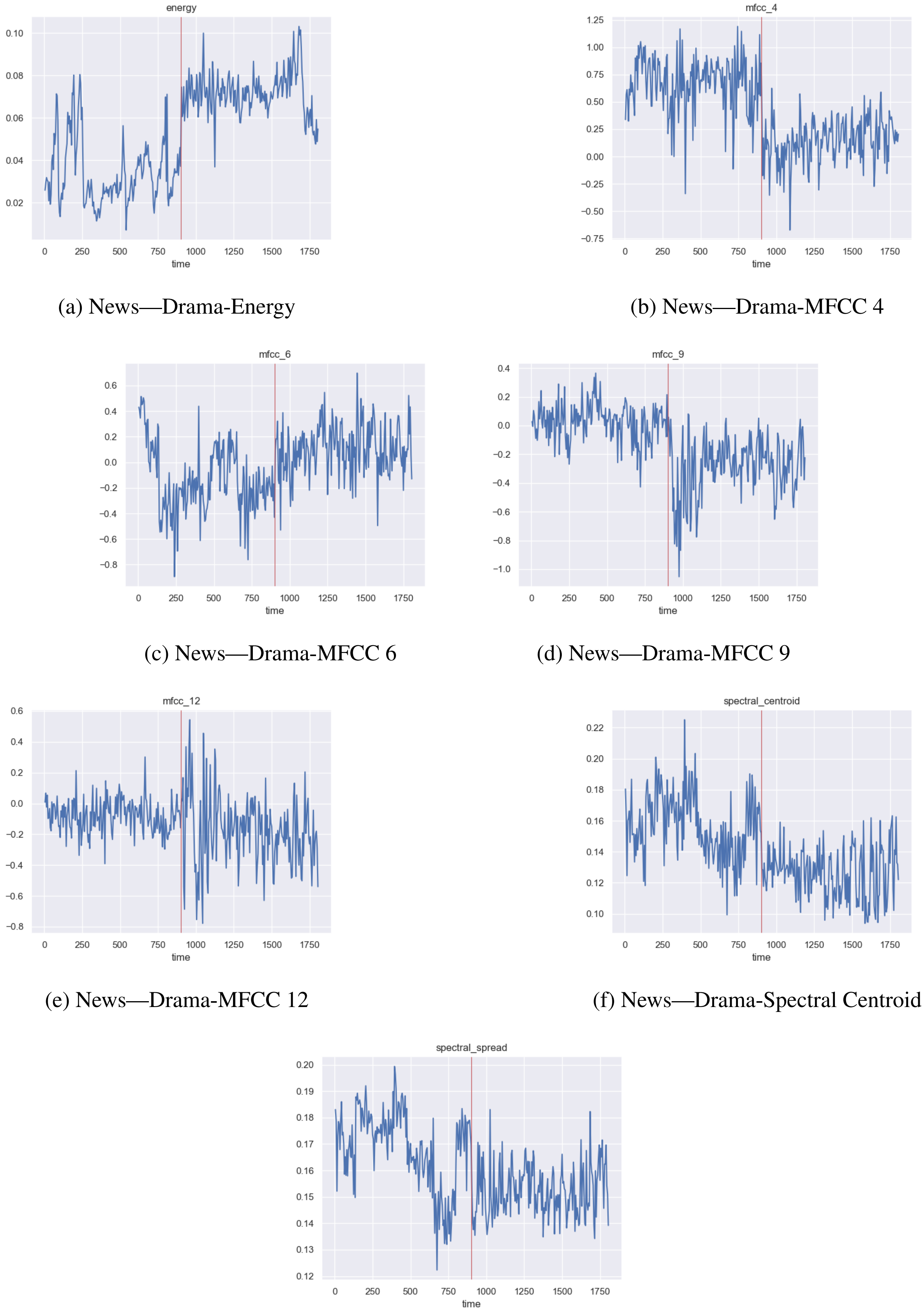
(d) News—Conversations-MFCC 10

(e) News—Conversations-Spectral Centroid

(f) News—Conversations-Spectral Entropy

(g) News—Conversations-Spectral Rolloff

Figure B.3: Feature patterns of selected features to distinguish news and conversations



(a) News—Drama-Energy

(b) News—Drama-MFCC 4

(c) News—Drama-MFCC 6

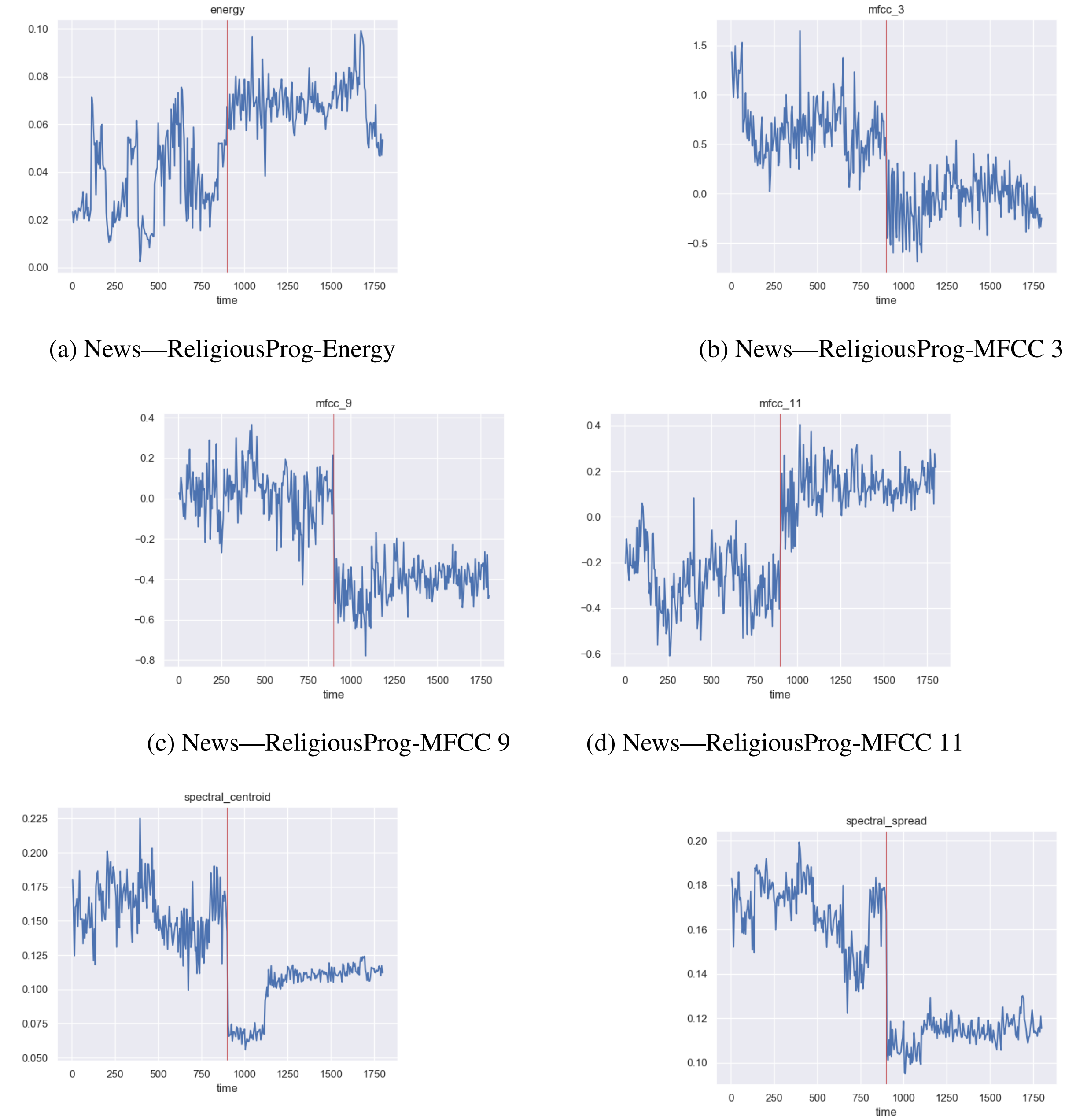
(d) News—Drama-MFCC 9

(e) News—Drama-MFCC 12

(f) News—Drama-Spectral Centroid

(g) News—Drama-Spectral Spread

Figure B.4: Feature patterns of selected features to distinguish news and radio drama



(a) News—ReligiousProg-Energy

(b) News—ReligiousProg-MFCC 3

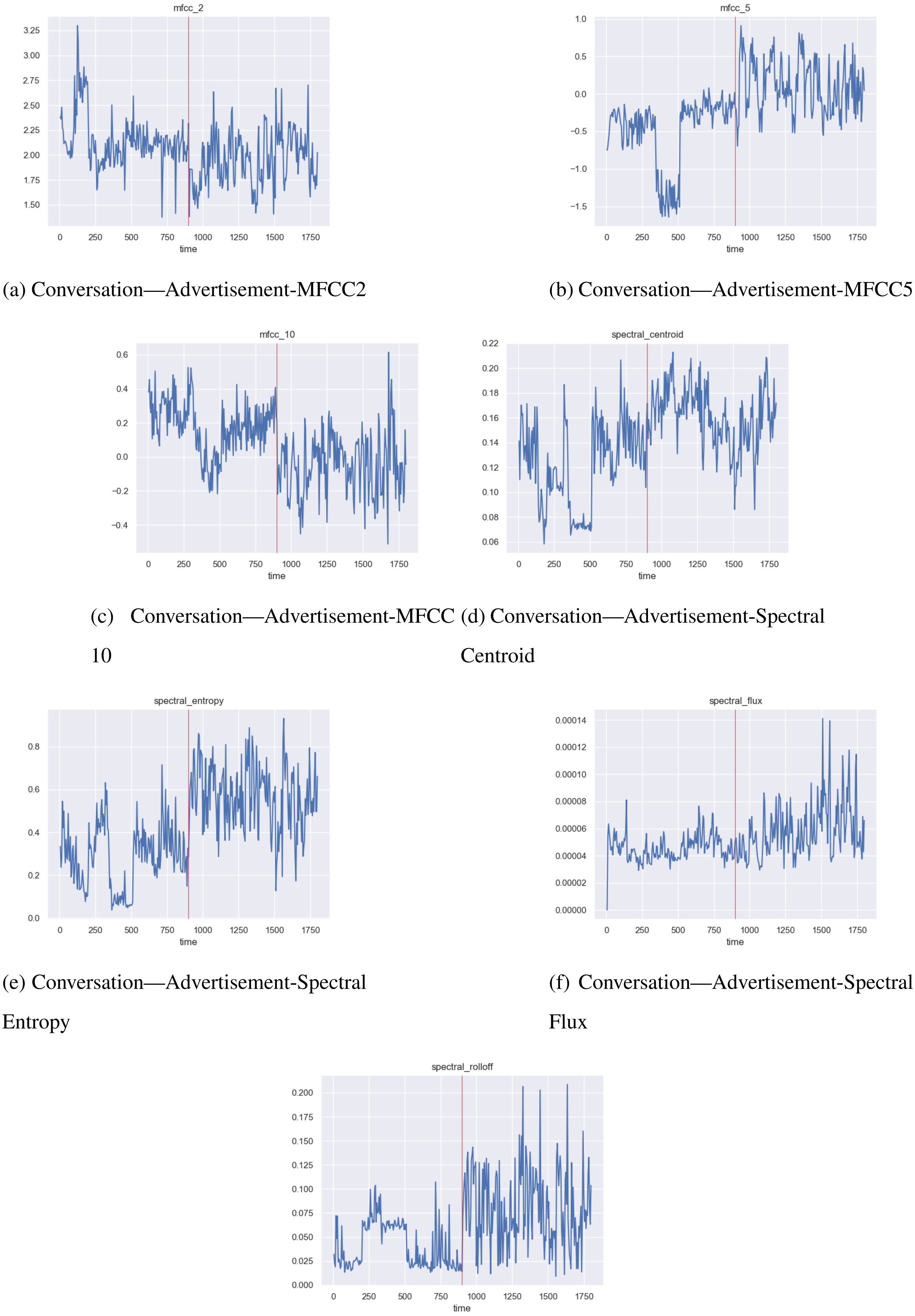
(c) News—ReligiousProg-MFCC 9

(d) News—ReligiousProg-MFCC 11

(e) News—ReligiousProg-Spectral Centroid

(f) News—ReligiousProg-Spectral Spread

Figure B.5: Feature patterns of selected features to distinguish news and religious programs



(a) Conversation—Advertisement-MFCC2

(b) Conversation—Advertisement-MFCC5

(c) Conversation—Advertisement-MFCC 10

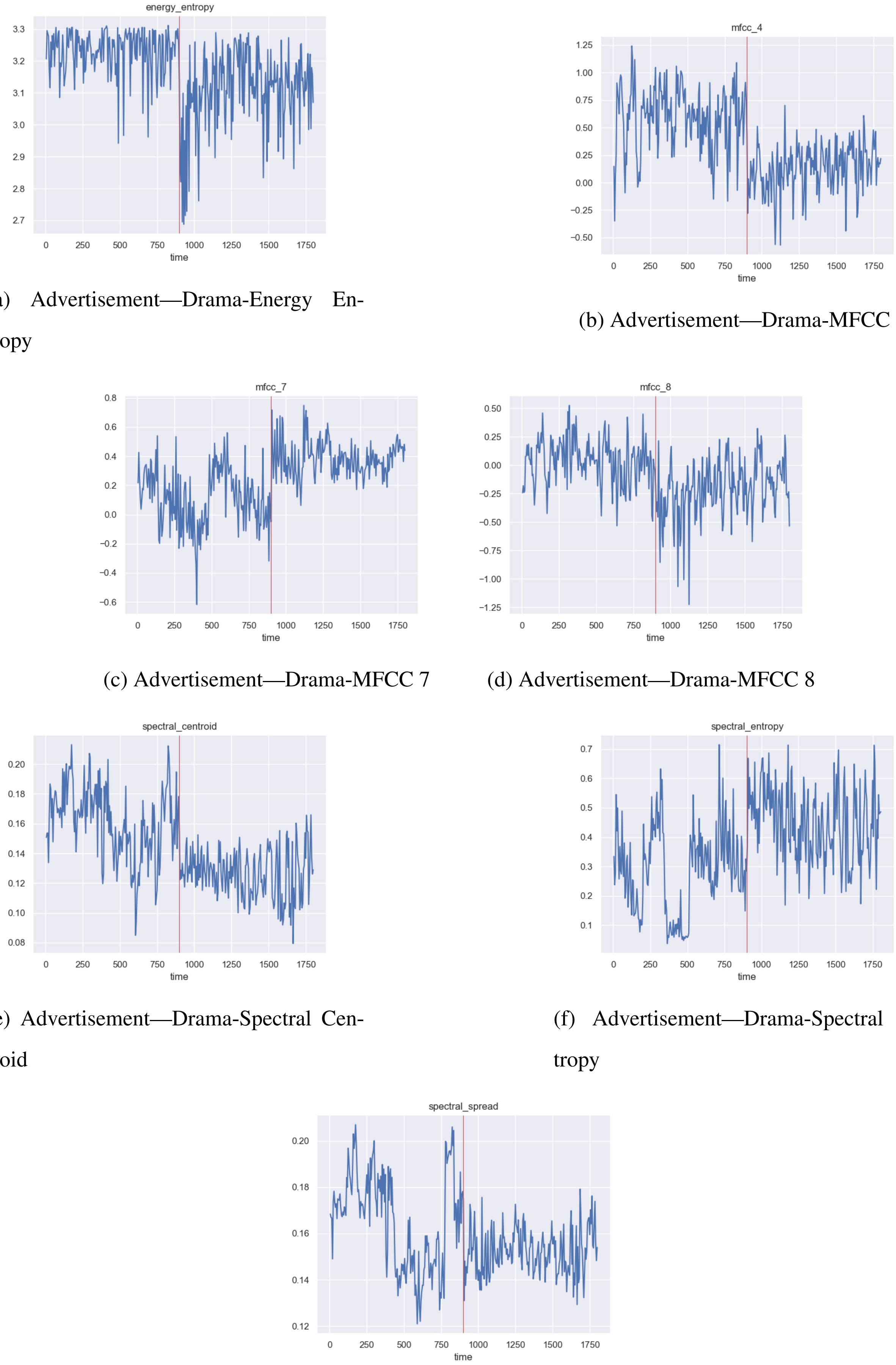
(d) Conversation—Advertisement-Spectral Centroid

(e) Conversation—Advertisement-Spectral Entropy

(f) Conversation—Advertisement-Spectral Flux

(g) Conversation—Advertisement-Spectral Rolloff

Figure B.6: Feature patterns of selected features to distinguish conversations and advertisements



(a) Advertisement—Drama-Energy Entropy

(b) Advertisement—Drama-MFCC 4

(c) Advertisement—Drama-MFCC 7

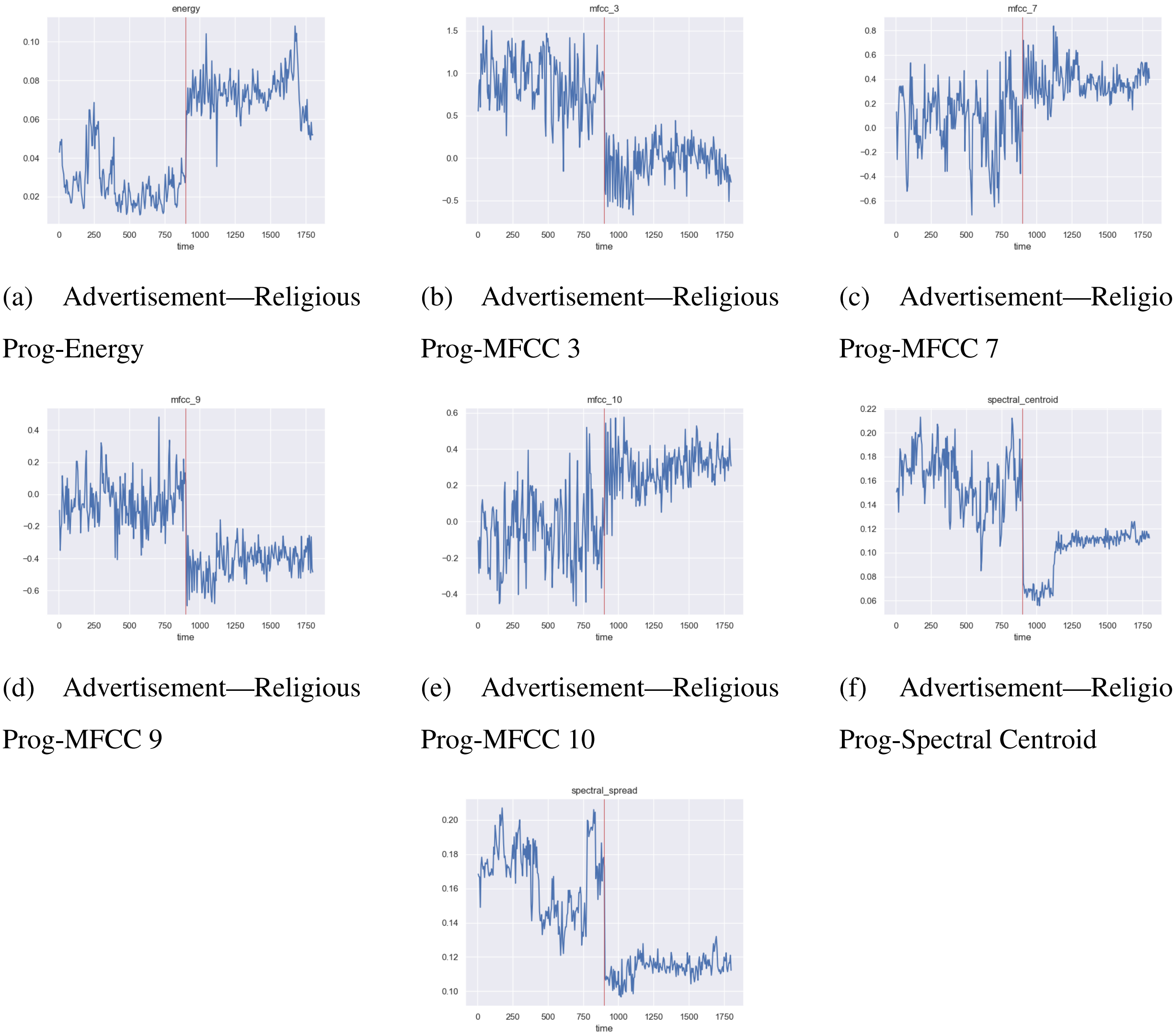
(d) Advertisement—Drama-MFCC 8

(e) Advertisement—Drama-Spectral Centroid

(f) Advertisement—Drama-Spectral Entropy

(g) Advertisement—Drama-Spectral Spread

Figure B.7: Feature patterns of selected features to distinguish advertisements and radio drama



(a) Advertisement—Religious Prog-Energy

(b) Advertisement—Religious Prog-MFCC 3

(c) Advertisement—Religious Prog-MFCC 7

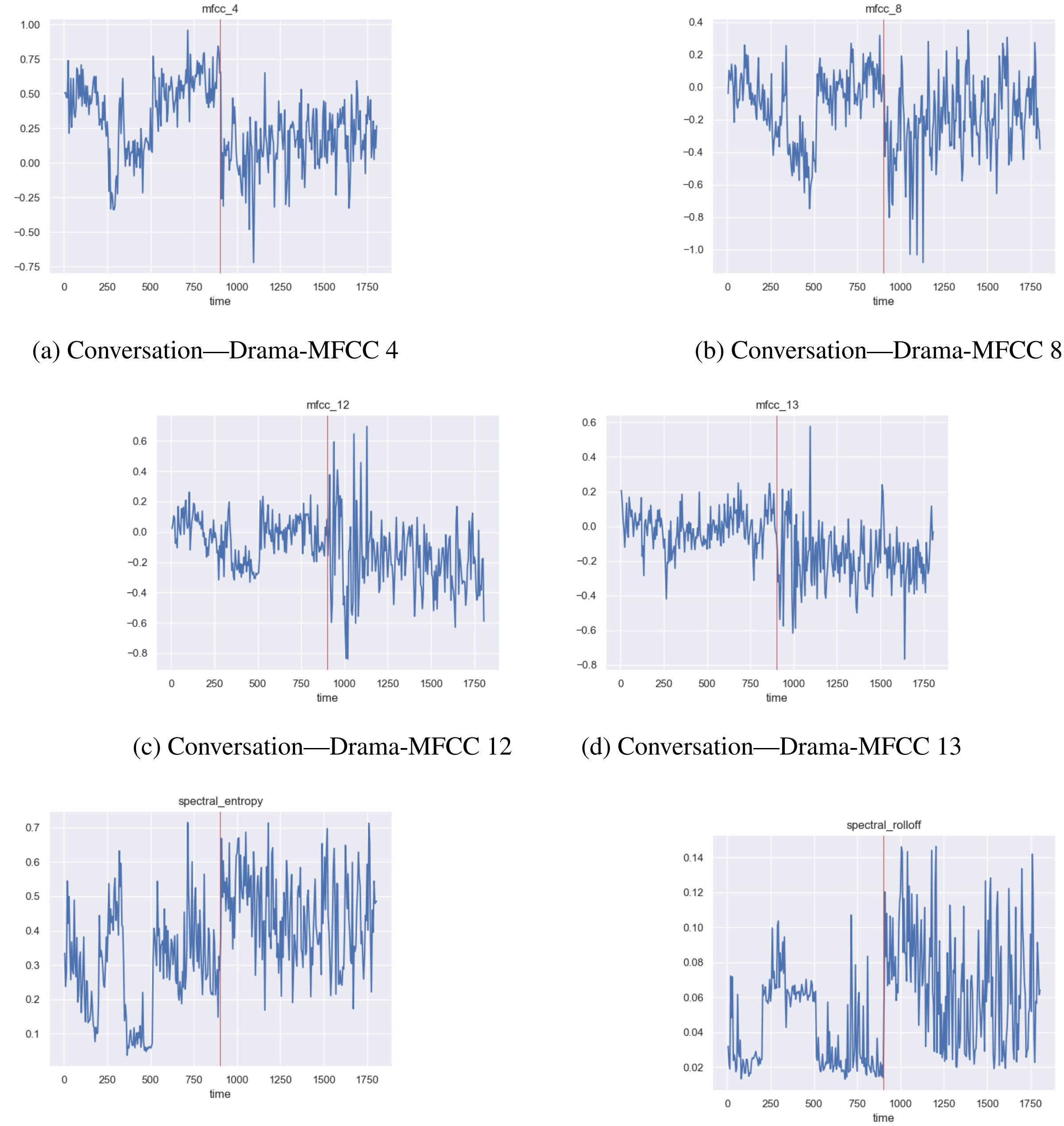
(d) Advertisement—Religious Prog-MFCC 9

(e) Advertisement—Religious Prog-MFCC 10

(f) Advertisement—Religious Prog-Spectral Centroid

(g) Advertisement—Religious Prog-Spectral Spread

Figure B.8: Feature patterns of selected features to distinguish advertisements and religious program



(a) Conversation—Drama-MFCC 4

(b) Conversation—Drama-MFCC 8

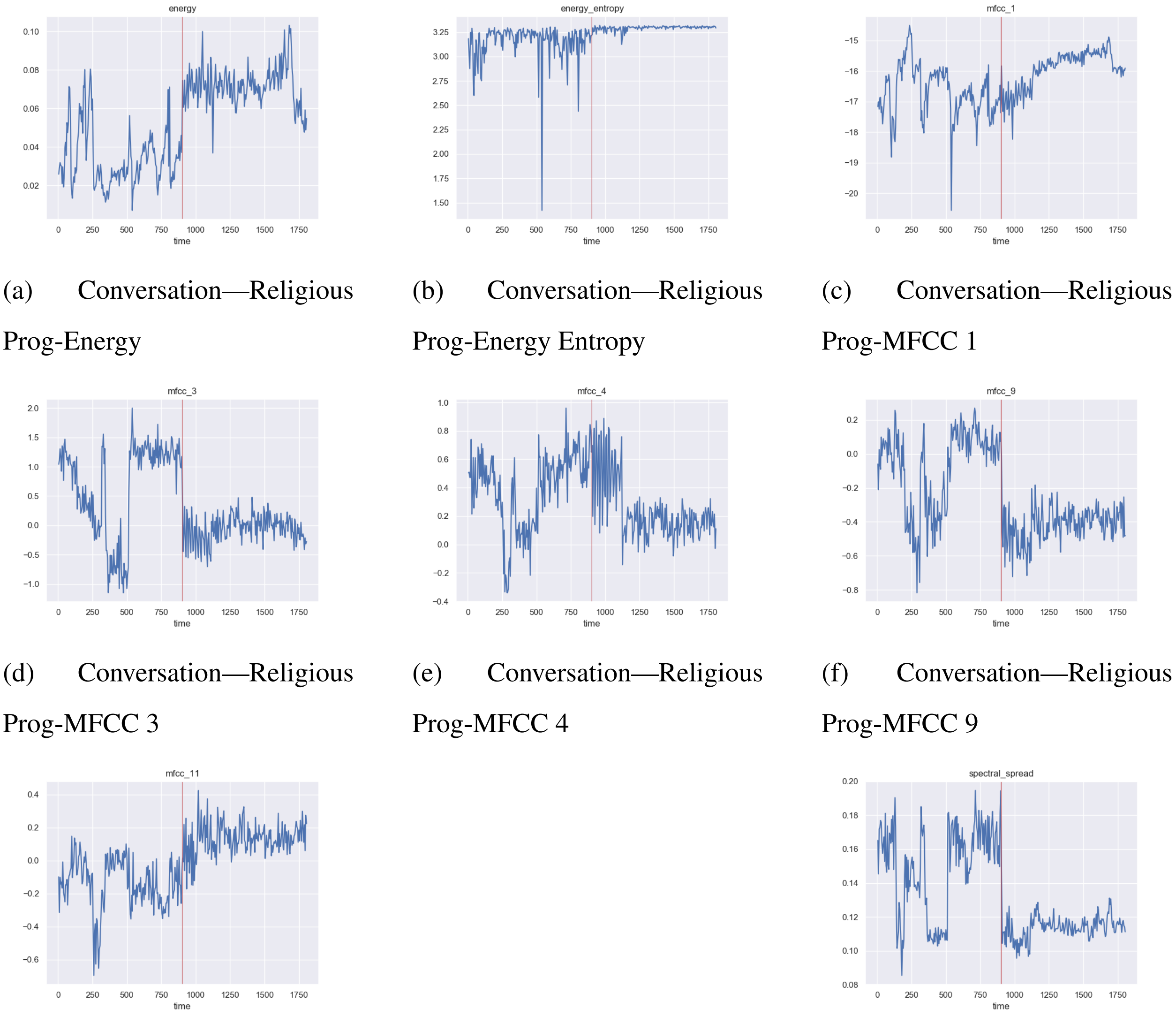
(c) Conversation—Drama-MFCC 12

(d) Conversation—Drama-MFCC 13

(e) Conversation—Drama-Spectral Entropy

(f) Conversation—Drama-Spectral Rolloff

Figure B.9: Feature patterns of selected features to distinguish conversations and drama



(a) Conversation—Religious Prog-Energy

(b) Conversation—Religious Prog-Energy Entropy

(c) Conversation—Religious Prog-MFCC 1

(d) Conversation—Religious Prog-MFCC 3

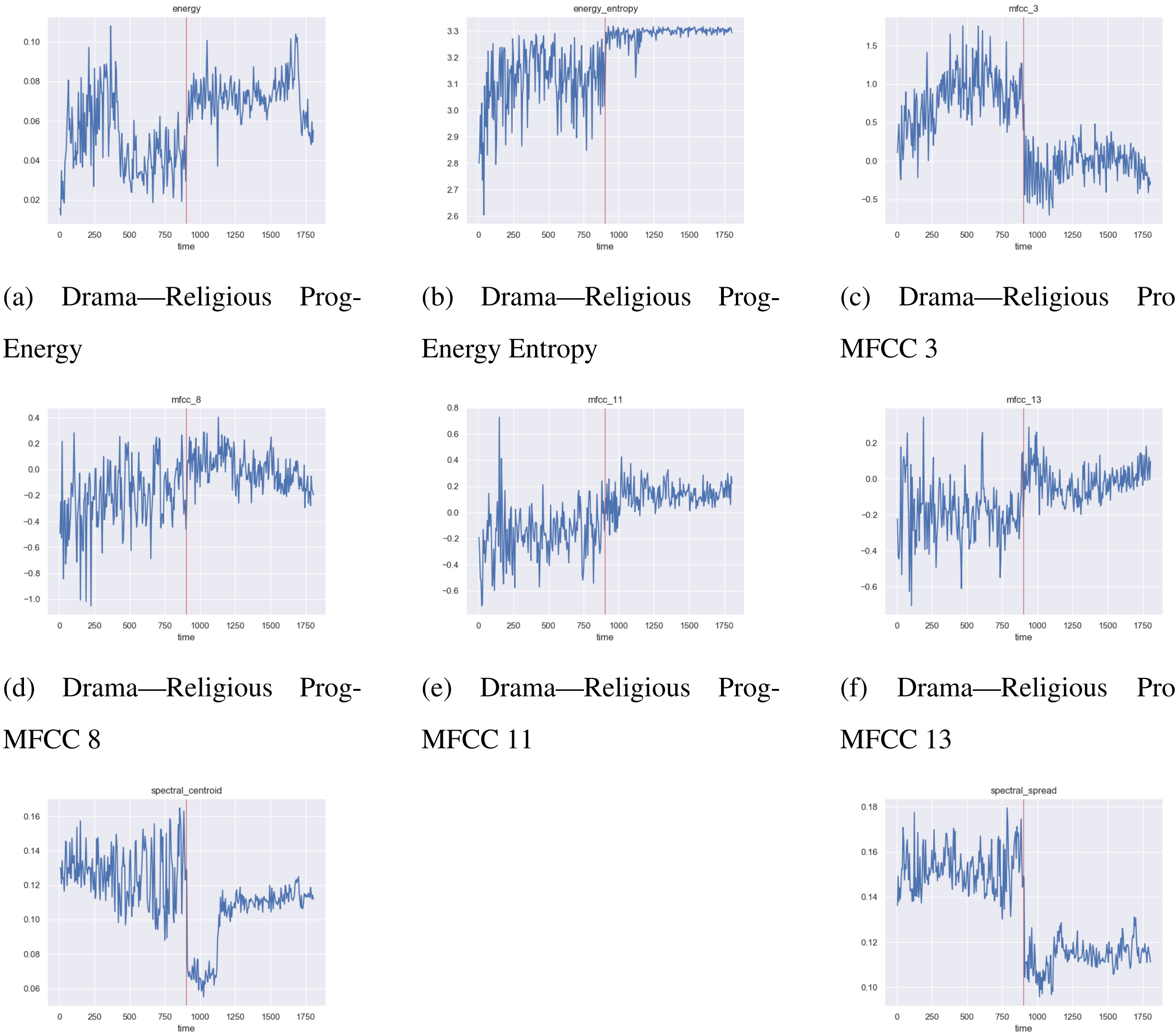
(e) Conversation—Religious Prog-MFCC 4

(f) Conversation—Religious Prog-MFCC 9

(g) Conversation—Religious Prog-MFCC 11

(h) Conversation—Religious Prog-Spectral Spread

Figure B.10: Feature patterns of selected features to distinguish conversations and religious programs



(a) Drama—Religious Prog- Energy

(b) Drama—Religious Prog- Energy Entropy

(c) Drama—Religious Prog- MFCC 3

(d) Drama—Religious Prog- MFCC 8

(e) Drama—Religious Prog- MFCC 11

(f) Drama—Religious Prog- MFCC 13

(g) Drama—Religious Prog- Spectral Centroid

(h) Drama—Religious Prog- Spectral Spread

Figure B.11: Feature patterns of selected features to distinguish radio drama and religious programs