OPTIMISING ACOUSTIC FEATURES FOR SOURCE MOBILE DEVICE IDENTIFICATION USING SPECTRAL ANALYSIS TECHNIQUES

MEHDI JAHANIRAD

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2016

OPTIMISING ACOUSTIC FEATURES FOR SOURCE MOBILE DEVICE IDENTIFICATION USING SPECTRAL ANALYSIS TECHNIQUES

MEHDI JAHANIRAD

THESIS SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2016

UNIVERSITY OF MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: Mehdi Jahanirad

Registration/Matric No: WHA120003

Name of Degree: Doctor of Philosophy

Title of Thesis: OPTIMISING ACOUSTIC FEATURES FOR SOURCE MOBILE DEVICE IDENTIFICATION USING SPECTRAL ANALYSIS TECHNIQUES

Field of Study: Audio Forensics

I do solemnly and sincerely declare that:

I am the sole author/writer of this Work;
This Work is original;
Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date: 23/08/2016

Subscribed and solemnly declared before,

Witness's Signature

Name:

Designation:

Witness's Signature

Name:

Designation:

ABSTRACT

Forensic techniques can be used to identify the source of a digital data. This is also known as forensic characterization, which means identifying the type of device, model, and other characteristics. Despite this, in recent years, the problem of multimedia source identification has extended its focus from identifying image/video sources toward audio sources. To determine the source of the audio several techniques have been developed. Those techniques work by identifying the acquisition device's fingerprint as the detection features. However, the prior works have rarely considered audio evidence in a form of a recorded call. In filling that research gap, this thesis looks at intrinsic artifacts of both transmitting and receiving ends of a recorded call. Meanwhile, the influences such as speakers, environmental disturbances, channel distortions and noise contaminate the discrimination ability of the feature sets for source communication device identification. Hence, addressing robust feature extraction methods for source communication device identification is necessary.

This study utilized spectral analysis techniques to investigate the use of linear and nonlinear systems for modeling the mobile device frequency response on the call recording signal. The context model allows computing the mobile device intrinsic fingerprints for the source mobile device identification. To achieve this aim, this study proposed a novel framework which extracts the mobile device intrinsic fingerprints from near-silent segments by using two spectral analysis approaches: (a) for linearized modeling, the proposed framework uses the cepstrum estimation technique and extracts entropy of Mel-frequency cepstral coefficients (MFCCs), (b) for non-linear modeling, the framework employs higher-order spectral analysis (HOSA) and utilizes the Zernike moments (ZMs) of the bicoherence magnitude and phase spectrum. Both models optimize acoustic features for source mobile device identification based on near-silent segments. The proposed feature sets along with selected feature extraction methods from the

literature are analyzed and compared by using supervised learning techniques (i.e. support vector machines, nearest-neighbor, naïve Bayesian, neural network, logistic regression, and ensemble trees classifier), as well as unsupervised learning techniques (i.e. probabilistic-based and nearest-neighbor-based algorithms). The analysis was performed based on inter- and intra-model mobile device identification among 120 mobile devices in 12 models for speech and non-speech segments under different environmental influences, communication networks, and stationaries. For inter-model mobile device identification, the best performance was achieved with entropy-MFCC features and nearest-neighbor classifier, which resulted in an average accuracy of 99.63%. For intra-model mobile device identification, the best performance was achieved with ZMs of bicoherence magnitude and phase features and nearest-neighbor classifier, which resulted in an average accuracy of 98.45%.

ABSTRAK

Teknik-teknik forensik boleh digunakan untuk mengenal pasti sumber data digital. Ini juga dikenali sebagai pencirian forensik yang bermaksud mengenal pasti jenis peranti, model dan ciri-ciri lain. Bagaimanapun, pada tahun-tahun kebelakangan ini, perhatian terhadap masalah pengenalan sumber multimedia telah beralih daripada mengenal pasti sumber imej / video ke arah sumber audio. Untuk menentukan sumber audio, beberapa teknik telah dibangunkan untuk mengenal pasti cap jari peranti pengambilalihan dengan menggunakan isyarat audio. Walau bagaimanapun, kerja-kerja penyelidikan sebelum ini jarang mengambil kira mengenai bukti audio dalam bentuk panggilan yang telah direkodkan. Untuk menutup jurang penyelidikan ini, tesis ini melihat kepada kandungan artifak intrinsik kedua-dua penghantar dan penerima panggilan yang telah direkodkan. Dan juga, pengaruh-pengaruh seperti pembesar suara, gangguan alam sekitar, gangguan saluran dan bunyi mencemarkan keupayaan diskriminasi set ciri untuk mengenal pasti pengenalan sumber peranti komunikasi. Oleh itu, menangani kaedah ciri pengekstrakan yang teguh untuk pengenalan sumber peranti komunikasi adalah perlu.

Kajian ini menggunakan teknik-teknik *spectral analysis* untuk menyiasat penggunaan sistem *linear* dan *non-linear* untuk pemodelan sambutan frekuensi peranti mudah alih pada isyarat rakaman panggilan. Model konteks ini membolehkan pengiraan untuk mengenal pasti cap jari intrinsik untuk pengenalan sumber peranti mudah alih. Untuk mencapai matlamat ini, kajian ini mencadangkan satu rangka kerja yang novel dimana pengekstrakan cap jari intrinsik peranti mudah alih dibuat daripada segmen-segmen *near-silent* dengan menggunakan dua pendekatan *spectral analysis*: a) untuk pemodelan *linear*, rangka kerja yang dicadangkan menggunakan teknik *cepstrum estimation* dan pengekstrakan *entropy of mel-frequency cepstrum coefficients* (MFCCs), b) untuk pemodelan *non-linear*, rangka kerja ini menggunakan *higher-order spectral analysis* (HOSA) dan *Zernike moments of the bicoherence magnitude and phase spectrum*. Kedua-

dua model ini mengoptimumkan ciri-ciri akustik untuk pengenalan sumber peranti mudah alih berdasarkan segmen *near-silent*. Set ciri yang dicadangkan, bersama-sama dengan kaedah pengekstrakan ciri-ciri ini dipilih daripada penyelidikan sedia akan dianalisis dan dibandingkan dengan menggunakan teknik *supervised learning* (i.e. *support vector machine, nearest-neighbor, naïve Bayesian, neural network, logistic regression,* dan pengelas *ensemble trees*), berserta teknik *unsupervised learning* (i.e. *probabilistic-based* dan *nearest-neighbor-based algorithms*). Analisis telah dilakukan atas dasar *inter-* dan *intra-model* pengenalan peranti mudah alih di kalangan 120 peranti mudah alih dalam 12 model untuk segmen *speech* dan *non-speech* di bawah pengaruh persekitaran, rangkaian komunikasi dan penjagaan yang berbeza. Untuk *inter-model* pengenalan peranti mudah alih, prestasi yang terbaik dicapai dengan ciri-ciri *entropy-MFCC* dan pengelas *nearest-neighbor*, mendapat ketepatan purata 99.63%. Untuk *intra-model* pengenalan peranti mudah alih, prestasi yang terbaik dicapai dengan *Zernike moments of bicoherence magnitude and phase features* dan pengelas *nearest-neighbor*, mendapat ketepatan purata 98.45%.

ACKNOWLEDGEMENTS

Allah is very kind, merciful and sympathetic. His benevolence and blessings enable me to achieve this thesis.

Firstly, I would like to express my profound gratitude to my supervisors, Dr. Nor Badrul Anuar, Dr. Ainuddin Wahid Abdul Wahab for giving me the opportunity to work with them, for encouraging me to be independent and strong about my work, for sharing their knowledge and experience with me, for being caring and supportive, for their guidance and consultancy. They led me through many helpful discussions and have been the constant sources of motivation, guidance, encouragement, and trust. Their invaluable suggestions and ideas have helped me walk through each stage of my research, while their passion and extraordinary dedication have inspired me to work harder and succeed.

I must also acknowledge the financial support given by the Ministry of Education, Malaysia under the University of Malaya High Impact Research (HIR) Grant UM.C/625/1/HIR/MoE/FCSIT/17, which has been the important role to make possible this thesis and allowed me to present results and exchange knowledge and skills in my works.

Also, I dedicate this work to my father who has been my best mentor since childhood, who taught me to work hard and to make my dreams come true and who always kept his belief on me. I owe him for the rest of my life for all the sacrifices that he made and for all the support that he gave to me. To my precious mother, for her endless and undemanding love, for her prayers and for the life that she spent on growing me up. To my two brothers and my only sister for always being supportive, for giving me the best advice when I needed. Last but not least, I want to give my special thanks to my beloved wife for always being on my side and to her nonstop help in crucial moments. I love you all.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AAFE	:	AMSL audio feature extractor
AAST	:	AMSL audio stage analysis toolset
ACC	:	Identification accuracy
ADC	:	Analogue to digital converter
AIC	:	Akaike information criterion
AM–AM	:	Amplitude Modulation–Amplitude Modulation
AM–PM	:	Amplitude Modulation–Phase Modulation
AMSL	:	Advanced multimedia and security lab, Otto-von-Guericke University Magdeburg, Germany
BFCC	:	Bark-frequency cepstral coefficients
CDIM	:	Communication device identification modules
CDMA	:	Code division multiple access
CMN	:	Cepstral mean normalization
CMVN	:	Cepstral mean and variance normalization
CVN	:	Cepstral variance normalization
DAC	:	Digital to analog converter
DBSCAN	:	Density-based spatial clustering of applications with noise
DCT of MFBE	:	Discrete cosine transform of Mel-filterbank energies
DET	:	Detection trade-off
DFT	:	Discrete Fourier transform
DSP	:	Digital signal processor
DWBC	:	Discrete wavelet based coefficient
DWT	:	Discrete wavelet transform
EER	:	Equal error rate
EM	:	Expectation-maximization

ENF	:	Electric network frequency
FaNT	:	Filtering and noise adding tool
FDMA	:	Frequency division multiple access
GLDS	:	Generalized linear discriminant sequence
GMM	:	Gaussian mixture models
GMSK	:	Gaussian minimum shift keying
GSM	:	Global system for mobile
GSV	:	Gaussian supervector
HOSA	:	Higher-order spectral analysis
HTIMIT	:	Handset-Texas instruments and Massachusetts institute of technology
ITU	:	International telecommunication union
LDC	:	Linguistic data consortium
LFCC	:	Linear-frequency cepstral coefficients
LIBSVM	:	Library for support vector machines
LL	:	Log-likelihood
LLHDB	:	Lincoln-labs handset database
LNA	:	Low-noise amplifier
LO	:	Local-Oscillator
LPCC	:	Linear prediction cepstral coefficients
LSF	:	Labeled spectral features
LTI	:	Linear, time-invariant
MAE	:	Mean absolute error
MDCT	:	Modified discrete cosine transform
MDL	:	Minimum discerption length
ME	:	Magnitude error

MFCC	:	Mel frequency cepstral coefficients
FMFCC	:	Fractional mel frequency cepstral coefficients
ML	:	Maximum likelihood
MMI	:	Maximum mutual information
MPEG	:	Moving pictures experts group
MUSIC	:	Multiple signal classification
NGI	:	Non-Gaussianity index
NIST	:	National institute of standards and technology
NIST-SRE	:	National institute of standards and technology - Speaker recognition evaluation
NLI	:	Nonlinearity index
NN	:	Neural network
OCC	:	One class classification
PA	:	Power Amplifier
PCA	:	Principal component analysis
PE	:	Phase error
PFA	:	Probability of false acceptance
PLPC	:	Perceptual linear predictive coefficients
PMF	:	Probability mass function
PSTN	:	Public switched telephone network
QPC	:	Quadratic phase coupling
RAE	:	Relative absolute error
RBF	:	Radial basis function
RF	:	Radio frequency
RICF	:	Representative instance classification framework
RMSE	:	Root mean square error

ROC	:	Receiver operating characteristic
RRSE	:	Root relative squared error
RSF	:	Random spectral features
SDR	:	Software defined radio
SL	:	Simple logistic
SMO	:	Sequential minimal optimization
SNR	:	Signal-to-noise ratio
SRC	:	Sparse representation-based classification
SRE	:	Speaker recognition evaluation
SSF	:	Sketches of spectral features
STFT	:	Short time Fourier transform
SVD	:	Singular value decomposition
SVM	:	Support vector machines
TDMA	:	Time division multiple access
TIMIT	:	Texas instruments and Massachusetts institute of technology
TNLI	:	Total nonlinearity index
TPR	:	True positive rate
UBM	:	Universal background model
VoIP	:	Voice over Internet Protocol
VQ	:	Vector quantization
WEKA	:	Waikato environment for knowledge analysis
WLAN	:	Wireless local area network
WPBC	:	Wavelet packet based coefficient
WPT	:	Wavelet packet transform
ZM	:	Zernike moments
ZMBic	:	Zernike moments of Bicoherence

LIST OF APPENDICES

CHAPTER 1: INTRODUCTION

Audio forensics have recently received considerable attention because it can be applied in different situations that require audio authenticity and integrity (Kraetzer et al., 2012). Such situations include forensic acquisition, analysis, and evaluation of admissible audio recordings as crime evidence in court cases. On the other hand, digital audio technology development has facilitated the manipulation, processing, and editing of audio by using the advanced software without leaving any visible trace. Thus, basic audio authentication techniques, such as listening tests and spectrum analysis (Koenig & Lacey, 2009), are easy to cross over. The authenticity of audio evidence is important as part of a civil and criminal law enforcement investigation or as part of an official inquiry into an accident or other civil incidents. In these processes, authenticity analysis determines whether the recorded information is original, contains alterations, or has discontinuities with respect to the recorder stops and starts. As a result of the critical role of audio authenticity in the audio forensic examination, different approaches have been proposed to define audio authenticity based on artifacts extracted from signals. These techniques consist of: (a) frequency spectra introduced by the recording environment (i.e., environment-based techniques), (b) frequency spectra produced by the recording device (i.e., device-based techniques), and (c) frequency spectra generated by the recording device power source (i.e., electric network frequency (ENF) based techniques) (Maher, 2009).

Although this field is still in the developing stage and faces different challenges, it holds tremendous potential for the further research. For example, the performance of environment-based techniques is highly dependent on the existence of discriminative background noise or strong environmental reverberations (Ikram & Malik, 2010; Muhammad & Alghathbar, 2013). However, recent advanced audio forgery software allows counterfeiting environmental effects without leaving any trace in the original file,

which is a disadvantage of environment-based techniques. Alternatively, in some studies ENF based techniques present high accuracy and novelty (Cooper, 2009, 2011; Garg et al., 2013), but occasionally the audio recording lacks embedded ENF pattern. A special case is when the appliance is battery powered and locates outside the coverage of the electromagnetic field that generates from the electric network. By assuming that the ENF pattern is detectable on the audio evidence, this method requires the ENF database for all power grids in the world. Unfortunately, currently, such database is only available for limited areas. Furthermore, the speech codec has also been considered for validation of the audio acquisition process, whereby the problem is the identification of the codec utilized in the transmission channel (Jenner, Nov. 2011; Sharma et al., 2010). Balasubramaniyan et al. (2010) utilized call recording to determine its origin by identifying the network traversed (i.e. cellular, Voice over Internet Protocol (VoIP), and public switched telephone network (PSTN)) and the call source fingerprint (i.e. speech codec). These approaches took advantage of the fact that each communication network or application utilizes its own standardized codec.

However, the feasibility of speech codec identification approaches is limited to forensic scenarios (Gupta et al., 2012). In overall, due to these weaknesses, the aforementioned techniques sometimes achieve unsatisfactory performance. Hence, source audio device identification has become an important approach for audio forensic examination.

1.1 Forensic Characterization of Physical Devices

Audio source device identification has been established by borrowing ideas from image forensics research on forensic characterization of camera devices and applying them to audio forensics. Most techniques for forensic characterization of devices are blind-passive. The blind approach never uses the original content and the information

from the real device for the analysis. The passive approach hardly uses any watermarking based solution for the analysis. For example, Swaminathan et al. (2007) defined feature-based image source camera identification as a blind method that can identify internal elements of an acquisition device such as digital camera without having access to the real device. Khanna et al. (2006) developed a survey of forensic characterization methods for physical devices in order to verify the trust and authenticity of the data and the device that created it. This study presented three different scenarios, including digital cameras, printers and radio frequency (RF) devices (i.e. cell phones). The study assumed that the device is first stimulated by a specially designed probe signal, whereby the sampled device response contains characteristics unique to each device's brand and model.

1.2 Research Motivation

Audio source device identification has become an important technique for audio forensic examination. Communication devices such as mobile devices are supplied with built-in audio acquisition components such as microphones and software applications that enable the recording, storing, transmission and playback of audio signals. Furthermore, there are a variety of small and portable digital audio recorders (i.e. Olympus, Sony ICD and Zoom H1 digital voice recorder) specifically used for recording, storing and transferring audio evidence. Hence, identifying the characteristics of the device processed the audio signal makes it possible to authenticate the evidence, or to interpret it for further forensic analysis. Both objectives require the tools and techniques of audio engineering and digital signal processing to tackle the automatic content analysis of audio data and discover its inherent hidden patterns. Meanwhile, the intrinsic device fingerprints are computed based on the fact that for different electronic components, every manifestation of an electric circuit cannot have exactly the same transfer function (Hanilçi et al., 2012).

Despite the fact that audio source device identification was first introduced for microphone classification by Kraetzer et al. (2007), the criteria of audio source device identification approaches have been progressed from microphone forensics toward mobile and computer forensics. Nevertheless, realizing the perfect implementation of the audio source device identification for courtroom consideration is still far from accomplished.

1.3 Problem Statement

The majority of works in the field of audio source device identification have focused on identifying the recording device from traces of audio acquisition components on audio recording signal (D. Garcia-Romero & Epsy-Wilson, 2010; Hanilci & Kinnunen, 2014; Kraetzer et al., 2012; Malik & Miller, 2012; Panagakis & Kotropoulos, 2012b). However, these studies almost never considered call recordings collected during communication with mobile devices. The main challenges with call recording signal are because it contains intrinsic artifacts of both transmitting- and receiving ends, whereby the communication device artifacts could be delivered through calls that traverse cellular PSTN or VoIP networks. In filling that research gap, this thesis looks at intrinsic artifacts of both transmitting and receiving ends of a recorded call. Meanwhile, the influences such as speakers' characteristics, speech contents, environmental disturbances, channel distortions and noise contaminate the discrimination ability of the feature sets for source communication device identification. For example, Mel-cepstrum domain features such as Mel-frequency cepstral coefficients (MFCCs) extracted from speech recordings have been proven to be the most effective feature set to capture the frequency spectra produced by a recording device. However, previous works by D. Garcia-Romero and Epsy-Wilson (2010) and Hanilçi et al. (2012) eliminated speech contamination through collecting text- and speaker-independent training dataset. Having this limitation, the majority of the proposed feature sets in the literature such as MFCCs lack robustness for real-time

acquisition device identification. Hence, addressing robust feature extraction methods for source communication device identification is necessary.

In the case of real-time implementation, the source mobile device model identification should be able to identify the source model of the call recordings from mobile devices that are different with the mobile devices in the training dataset. This introduces the open set challenge scenario. Hence, the classification approach should be able to propose a solution based on both closed-set and open set scenarios.

1.4 Research Questions

A hypothesis of this research is that the transmitting mobile device artifacts could be delivered through calls that traverse cellular, PSTN or VoIP networks. These artifacts are the results of the nonlinear distortions generated due to the mobile device frequency response on call recording signal. The thesis utilized two different hypothesis for modeling the mobile device as a linear and nonlinear system. For the first hypothesis, the study eliminated the speech convolution by utilizing the near-silent segments of the call recording signal and utilized cepstral analysis techniques to linearize the convolution generated by the mobile device frequency response on audio spectrum. For the second hypothesis, the study utilized higher order spectral analysis to capture distortions on call recording signal generated by quadratic nonlinear subsystems in mobile devices.

Hence, the study investigates the use of state-of-the-art cepstrum-based and bispectrum-based features for source mobile device identification. For this investigation, the important questions are:

(a) Whether the call recording-based source mobile device identification is actually possible with the introduced approach?

(b) Which existing feature extraction methods and concepts can be used for the intended approach?

(c) Which methodological and conceptual deviations have to be made in this study from the paradigms currently used in the state-of-the-art?

(d) How to optimize acoustic features to capture mobile device specific information?

(e) Which classifiers are suitable for implementing the source mobile device identification?

(f) Is it actually possible to identify the source mobile device of the call recording that processed by the mobile device other than the ones utilized during the training with the introduced open set approach?

1.5 Aim and Objectives

The aim of this study is to propose a novel framework to address the optimization of acoustic features using spectral analysis techniques. The framework applies pattern recognition and machine learning techniques for source mobile device identification. In order to achieve this aim, the research challenges need to be thoroughly understood, analyzed and evaluated based upon the following objectives:

(a) To conduct a comprehensive study in the domain of audio source device identification, along with the most recent developments and emerging trends in the field.

(b) To design and implement a novel framework to facilitate practical evaluation of audio source mobile device identification from recorded call.

(c) To evaluate the performance of a proposed framework in terms of different performance metrics by validating it using evaluation studies in different phases in order to demonstrate the progress of results.

(d) To develop a source mobile device model identification prototype based upon the proposed framework.

The objectives presented above are related to the general sequence of the material presented in this study, the structure of which is discussed in the next section.

1.6 Research Scope and Limitations

For the considerations of the research challenges, the research scopes are as follows:

(a) Limits the contaminating influences during the data collection and preprocessing of the large set of call recording dataset.
(b) Optimizes acoustic features to detect the communication device frequency response despite the existence of convolutional transfer functions such as speech signal, channel noise, echo, and reverberation.
(c) Investigates the performance of state-of-the-art supervised and unsupervised machine learning techniques for source mobile device identification.
(d) Implements source mobile device model identification prototype based on an open set scenario with plausibility and forensic conformity.

Although designing the application-specific classification algorithm is important for performance enhancement, it is outside the direct scope of this study.

1.7 Research Methodology

The important methodical and conceptual components for the investigations performed in this study is summarized in Figure 1.1. The framework consists of three main components: (a) the *input* or *data collection*, (b) forensic analysis method for *source mobile device identification pipeline* and (c) the *output* or *evaluation methodology*. During data collection, the study prepares the test setup for the acquisition of the call recordings from different mobile devices and performs necessary modifications if

required. In the next step, the study extracts optimized features from near-silent segments of the call recordings corresponding to both training and testing dataset.

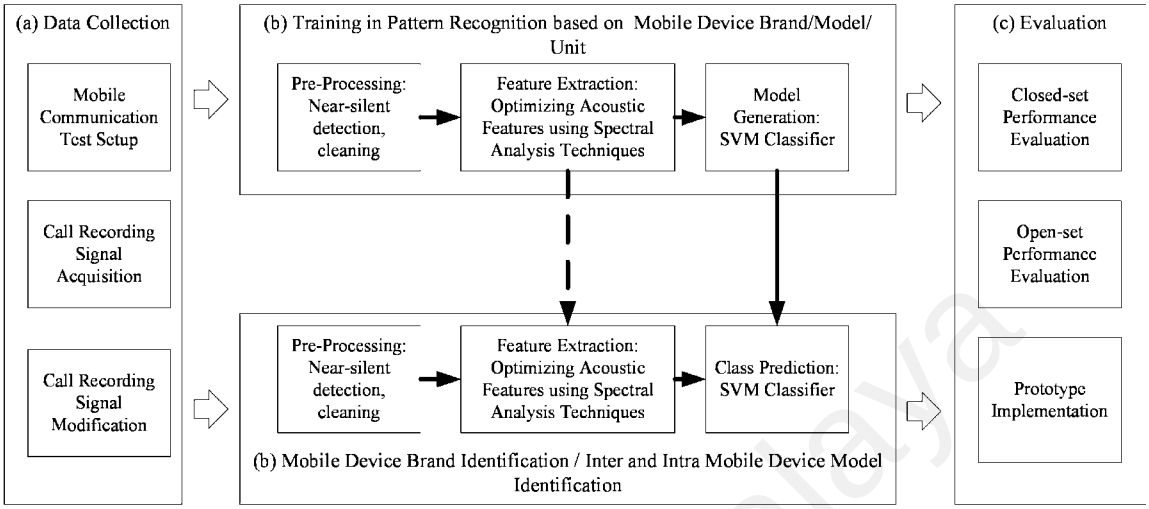


Figure 1.1: Research Methodical and Conceptual Components

Furthermore, this study is the classical pattern recognition problem that requires suitable machine learning technique, which handles large computation, provides accurate multi-class classification and shows robustness against unbiased and noisy data. Therefore, this study selected different classifiers among state-of-the-art supervised and unsupervised machine learning techniques for source mobile device identification.

Finally, this study utilizes the open set scenario for implementing source mobile device model identification prototype with plausibility and forensic conformity.

1.8 Thesis Organization

Chapter 2 studies an evolution of existing research on audio source device identification approaches by: (a) finding a precise and crisp classification of audio source acquisition device identification techniques, (b) giving a short review of previous works, (c) discussing the differences between these approaches and (d) identifying current challenges and open issues. Meanwhile, the general patterns and differences are investigated with regard to the following foundational data mining components: (a) data

preparation, (b) feature extraction, (c) feature analysis and (d) decision making. Overall, the chapter provides a generic and comprehensive view of contemporary audio source device identification approaches, in addition to the most recent developments and emerging trends in the field.

Chapter 3 provides the technical background and theory of the digital signal processing in the call recording context model. The chapter discusses both linearized and nonlinear approaches for modeling the mobile device frequency response on the call recording signal. Moreover, it provides an analysis of the proposed audio signal processing procedures and its significance on computing the intrinsic mobile device fingerprints. Moreover, the chapter describes issues on existing state-of-the-art feature extraction approaches. Finally, it provides justification on proposed features along with special concepts for spectral analysis techniques applied during feature extraction.

Chapter 4 describes design decisions for source mobile device identification framework. The chapter illustrates the source mobile device identification module through flow diagrams, architecture diagrams, and a brief discussion. The schema includes the pre-processing and data preparation steps, the procedure for extracting the features, as well as the detail and number of features. Furthermore, the chapter explains the classification and clustering algorithm, its efficiency and capability in brief, and then discusses the methodology for constructing the model based on the training data. Finally, the chapter describes the benchmark database along with the practical test setup, list of mobile devices and the number of recording data.

Chapter 5 plays a critical section in validating this study through conducting several experiments. The initial phase of the evaluation conducted using benchmark database to assess proposed features through measuring inter- and intra-model mobile device similarity and feature-based mobile device identification. The second phase of evaluation

includes benchmarking feature sets from previous works as well as different supervised and unsupervised learning algorithms. This evaluation also includes testing the robustness of the proposed features under different influences. Finally, the chapter discusses the outcome of the results, analyses the behavior of the features, and also provides the analysis about the reliability of the features from the forensic perspective.

Chapter 6 presents the implementation of a prototype system which embodies a full set of the key elements of the proposed framework, and described the interactions and relationships among them; namely the communication device identification modules (CDIM). Initially, it begins with an overview of the system development process, the system design, and the MATLAB interface modules. In addition, example scenarios are provided to demonstrate how the proposed framework operates, and how the MATLAB interfaces can be used to assist the forensic examiners in making a decision.

Chapter 7 provides the short summary that states, which parts of the research objectives accomplished and how successful are the outcomes. It also describes the limitations of the study and suggests new scopes for the future study.

The thesis also includes a number of appendices, which contain a variety of additional information in support of the main discussion, including several sets of source code and a number of peer-reviewed publications from this study.

CHAPTER 2: AUDIO SOURCE DEVICE IDENTIFICATION

Digital audio forensics have attracted research in developing audio mining methods and tools to analyze and evaluate audio recording from sources such as surveillance, telephone conversations, broadcast news, sports as well as personal and online audio collections. Hence, the trends in audio source device identification have changed in recent decades from microphone toward mobile and computer forensics. To understand the domain of the audio source device identification, this chapter presents an introduction to both spectral analysis and audio mining techniques, which are closely linked to audio forensics. Furthermore, this chapter provides the taxonomy of techniques in the field of the audio forensics and highlights the role of audio source device identification in the audio forensic examination. The chapter also introduces a model for classifying audio source device identification approaches, studies a phylogenetic tree of existing research to discover its conceptualizations, contributions and particular challenges based on the current techniques.

2.1 Digital Audio Forensics

Digital forensics deal with different forms of digital data that leave traceable information for crime investigation. The research in digital forensics suggests different directions related to the type of forensic information. Among many directions, audio forensic investigation refers to the acquisition, analysis, and evaluation of audio recordings that may ultimately be presented as admissible evidence in court or some other official venue. Although digital technology shadows the popularity of analog technology, in practice many law enforcement agencies use analog audio recordings because they can easily prove its admissibility. At the same time, proving the admissibility of digital evidence is problematic due to the widespread availability of digital sound processing software as well as its ease of operation that makes certain types of manipulation of audio

recordings comparatively easy to perform. If done competently, such manipulation may leave no traces and therefore, will be impossible to detect. In 1958 the judge for the specific legal case in the United States, firstly, defined seven admissibility requirements (Maher, 2009):

(a) That the recording device was capable of taking the conversation now offered in evidence;
(b) That the operator of the device was competent to operate the device;
(c) That the recording is authentic and correct;
(d) That changes, additions and deletions have not been made in the recording;
(e) That the recording has been preserved in a manner that is shown to the court;
(f) That the speakers are identified;
(g) That the conversation elicited was made voluntarily and in good faith, without any kind of inducement;

Most state and federal courts in the United States still use these seven requirements as a reference. However, after satisfying all the court admissibility requirements, it also requires the admissible forensic examination technique for analyzing the evidence. The court recognizes the forensic examination technique as admissible if the forensic analyzers can prove that their technique: (a) is unbiased, (b) has known reliability statistics, (c) is non-destructive, and (d) is widely accepted by experts in the field. As a result, researchers from interdisciplinary fields perform outstanding efforts to apply the tools and techniques of signal processing and audio mining in analyzing the audio data as part of the legal proceeding or an official investigation of some kind.

Audio forensic analysis involves three stages: (a) Authenticity, (b) Enhancement and (c) Interpretation. Through authenticity techniques, the forensic analyzer provides strong evidence to prove the admissibility of the audio recording. This includes the evidence that

determines the audio recording is from the same source as it claims, or that it is complete and original. In general, audio authentication techniques are in two types: passive and active. The passive approach focuses on forgery detection through the signal and its characteristics. In contrast, active techniques such as watermarking (Juan Garcia-Hernandez et al., 2013; Steinebach et al., 2012; Xiang et al., 2012), steganography (Kraetzer & Dittmann, 2007) and steganalysis (Geetha et al., 2010; Koçal et al., 2008) involve extra information embedded in the signal. Gupta et al. (2012) classified passive audio authentication techniques as basic or advanced. Basic audio authentication techniques include the listening test, spectrogram analysis, and spectrum analysis. The authors further grouped advanced audio authentication techniques into those that exploit audio recording conditions for forgery detection and those that use compressed audio features. Audio recording condition includes the recording environment (Malik & Mahmood, 2014; Muhammad & Alghathbar, 2013), recording device (Hanilci & Kinnunen, 2014; Panagakis & Kotropoulos, 2012b) and the recording device power source (Cooper, 2009, 2011; Garg et al., 2013). Moreover, double compressed audio files were detected using the modified discrete cosine transform (MDCT) coefficient statistics (Yang et al., 2010; Yang et al., 2009) and frame-off set detection (Koenig et al., 2013; Yang et al., 2008). Similarly, power spectrum analysis was implemented for detecting both copies of digital audio recordings and tampering operation (Cooper, 2008; Korycki, 2013).

Enhancement technique subtracts noise from the audio signal without losing any part of the desired signal (Gerkmann & Hendriks, 2012; Ikram & Malik, 2010). The interpretation technique identifies, verifies or recognizes the audio material such as events (McLoughlin et al., 2015), speech (Hsu & Lee, 2009; Ikbal et al., 2012) and gunshot (Freire & Apolin´ario, 2010; Maher, 2007; Valenzise et al., 2007), along with its source such as environment (Malik & Mahmood, 2014; Muhammad & Alghathbar, 2013),

speaker (Dileep & Sekhar, 2012; Kinnunen et al., 2012; Kuenzel, 2013) and weapon (Khan et al., 2010). The application of these techniques depends on the problem to be examined and its objective. Generally, in real scenarios, forensic examination requires the cooperation of enhancement, authenticity and interpretation techniques. For example, sometimes, the outcome of the enhancement process proves the authenticity of the audio recording through interpretation. In such scenario, the forensic analyzer subtracts the background noise from the audio recording, discovers its source environment and checks if it matches with the environment that the adversary claims. Based on the above discussion, Figure 2.1 illustrates the digital audio forensics' taxonomy based upon its existing research fields.

2.1.1 Forensics in the Context of Audio Source Device Identification

Proving the authenticity of an audio recording involves the verification of claims and statements associated with its content and history. Today, digital audio signal processing devices are made by numerous manufacturers, and come in thousands of different models. The ability to determine the source brand, model or an individual audio source acquisition device used to create a given recording is significantly useful. For example, microphone classification can be a valuable passive mechanism (e.g. perceptual hashing) in solving copyright disputes. Moreover, microphone identification can be used in determining whether the suspicious video has been made by the microphone seen in the video or whether the audio has been tampered with or completely replaced.

Similarly, audio source device identification is useful in determining the source characteristics in terms of another audio forensic approach like gunshot characterization (Maher, 2007). In addition, audio source device identification has applications in collecting metadata regarding the digital audio recordings. Currently, audio metadata contains information about the date that the audio was created and the date that it was

modified (Koenig & Lacey, 2012). However, it is plausible to provide metadata with information about the brand and model of the source device used to create the audio.

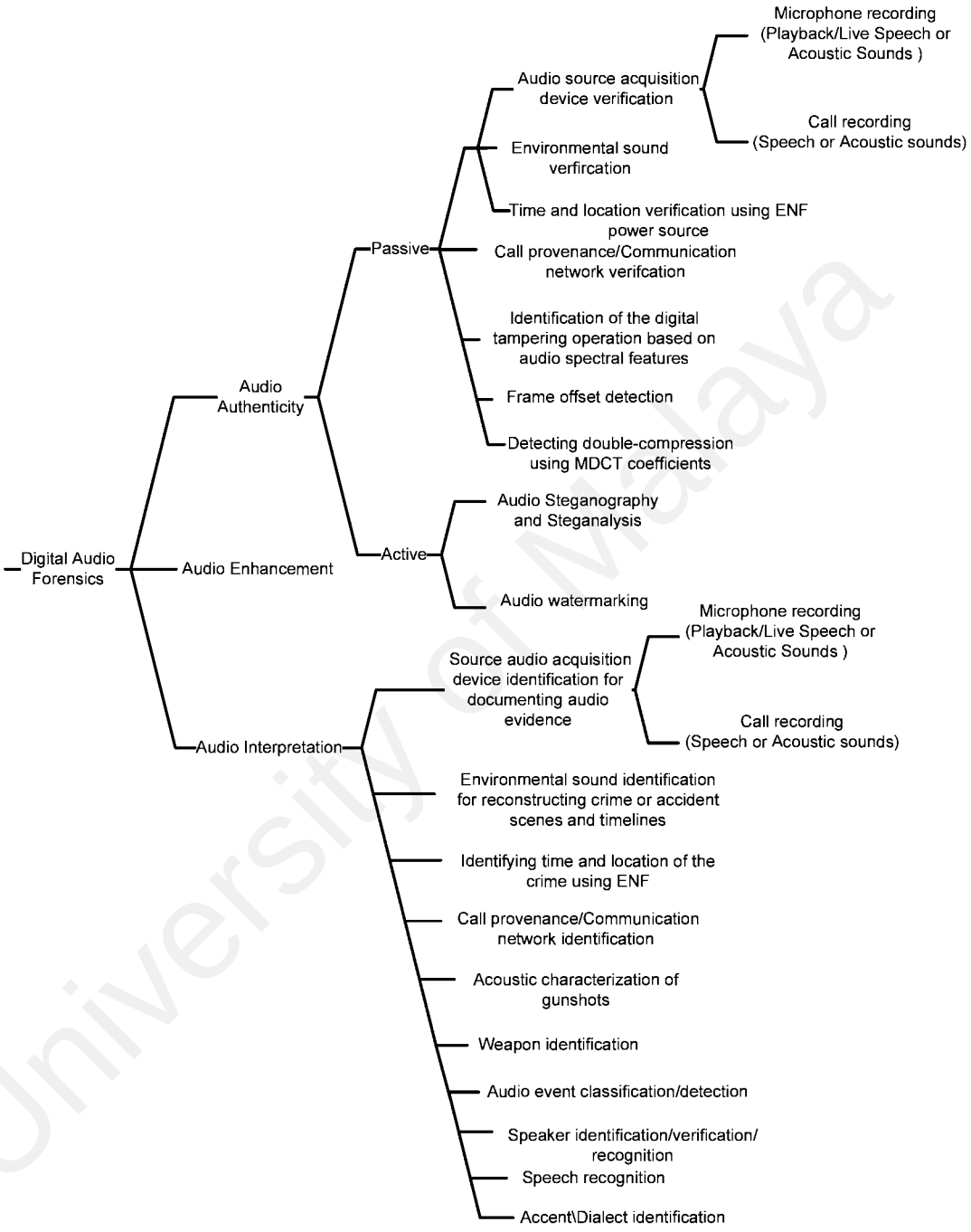


Figure 2.1: Digital Audio Forensics Taxonomy

Today, with the advent of networked computer devices (and especially with the connection of those networked computers to the outside world, and to each other over the Internet), computer and mobile devices (with the hardware necessary for speaking and listening, i.e. headphones and microphones, and voice communication software, i.e.

Skype) are widely used for communication. Hence, it is also possible for the forensic examiner to discover recordings, including calls between two parties, containing critical information. Consequently, identifying or verifying the source brand/model of the recording device through these recordings might be a valuable step in documenting the evidence. Alternatively, for call recording, it is plausible to identify the brand/model of the communication device rather than the recording device. The interpretation of the result also has many applications during law enforcement investigation.

2.1.2 The Use of Mining Techniques in Digital Audio Forensics

The most common pattern among audio forensic approaches is the use of audio mining tools for processing large databases. Audio mining solves audio forensic problems by analyzing available audio databases. This involves machine learning techniques for finding and describing structural patterns in data, as a tool to explain data and make predictions from it. In the essence of audio mining examples, machine learning means the acquisition of knowledge and the ability to use it. In overall all audio forensic approaches that implemented with mining techniques include six hierarchical stages: (a) domain understanding, (b) data selection, (c) pre-processing (data preparation, enhancement and transformation), (d) pattern recognition (feature extraction), (e) interpretation (feature analysis), and (f) decision making. The audio forensic researcher studies the case scenario, audio materials and proposes the forensic examination techniques. The standard data is sometimes available from recognized sources to test and compare different algorithms on the same set of problems. Alternatively, some researchers prefer to collect their data to test algorithms for specific case scenario. The quality and quantity of the database affect the overall audio forensic process performance. The database that integrated from different sources requires preparation and enhancement because it may contain missing values, unrelated data, or large values. The raw data transforms to important features that allow pattern discovery. This process employs machine learning

techniques to discover hidden patterns, relationships, and trends in the data. In the audio forensic investigation, evaluation, analysis, and acquisition of audio recordings occur at this stage. Hence, the final decision can be made based on knowledge gained from data.

2.2 Related Fundamentals on Audio Signals

Understanding signal processing allows to tackle automatic content analysis of audio data and discover hidden patterns. Hence, this section explains selected digital audio signal processing stages relevant to audio forensics.

2.2.1 Audio Signals, Noise, and Information

Hofmann et al. (2012) described a physical sound in a basic form as a mechanical disturbance of a medium that may be air, solid, liquid, gas or a mixture of mediums. This disturbance through the medium starts molecules to move back and forth in a spring-like manner. As one molecule hits the next, the disturbance moves through the medium causing sound to travel. These so-called compressions and rarefactions in the medium can be described as sound waves. The simplest type of waveform that defines 'simple harmonic motion', is a sine wave. Furthermore, Vaseghi (2008) defined a signal as the variation of the quantity such as air pressure sound waves. A signal conveys information regarding one or more attributes of the source such as the state, the characteristics, the composition, the trajectory, the evolution or the intention of the source. As a result, a signal contains information about the past, the current or the future states of a variable. Moreover, a signal is usually accompanied by a mixture of noises, and distortion. In reality, noise and distortions are the fundamental sources of the deficiency of (a) the capacity, or in another word the maximum speed, to send/receive information in a communication system, (b) the accuracy of measurements in signal processing and control systems and (c) the accuracy of decisions in pattern recognition. However, noise is usually represented as the unwanted signal that provides information on the state of its

source. Within this study, the same phenomenon was used to achieve information on type/model of the communication mobile devices based on their transmitted signal. Meanwhile, it is important to take advantage of the digital signal processing potentials, achieve a perfect understanding of the fundamentals of the science of signal processing and acknowledge the assumptions, implications, and limitations inherent in the proposed signal analysis method.

2.2.2 Audio Signal Processing Pipeline

The chain of digital audio signal processing includes an analog to digital converter (ADC), channel coding, transmission/storage, decoding, and playback, as shown in Figure 2.2. The transducer is usually a built-in or an external microphone that converts an acoustical energy in a form of pressure waves into electric waveforms. Digital recording devices such as sound card convert the electrical waveform from the transducer to binary numbers to represent the amplitude of the analogue signal $x(t)$ on an equidistant grid along the horizontal time axis and to perform quantization of the amplitudes to fixed samples that are represented by numbers $x(n)$ along the vertical amplitude axis. The time distance between two consecutive samples is known as sampling period T, and its reciprocal is defined as sampling frequency ($f_s=1/T$). This parameter determines the number of samples per second in hertz (Hz). The sampling frequency should be selected appropriately to avoid overlapping the spectrum of two adjacent sampling frequencies. According to sampling theory, to represent a signal digitally, its sampling frequency should be twice as the maximum frequency of the analog signal ($f_s > 2 \, . \, f_{max}$). The input signal with frequencies above ½ of the sampling frequency is passed to the low pass filter to pass all frequencies up to $f_{max.}$ The ADC uses more than 8bits/sample to convert the analog samples to digital format with a good quality. Digital recording device store or transmit these numbers in a coded form $x_c(n)$. Upon replay or reception, a playback device decodes the signal $\tilde{x}_c(n)$ and looks for unintended signal variations introduced and

makes the corrections as much as possible $y(n)$. However minor distortions and imperfections during transmission and storage has minimal effect on the design quality of the signal and remain within the design limits of the system. After decoding and signal modification the signal is passed through digital to analog converter (DAC) to convert the numerical data back to a time continuous analog audio signal and sends the electric waveforms $y(t)$ to the loudspeaker. However, the chain of digital audio signal processing slightly varies for microphone recording and call recording scenarios.

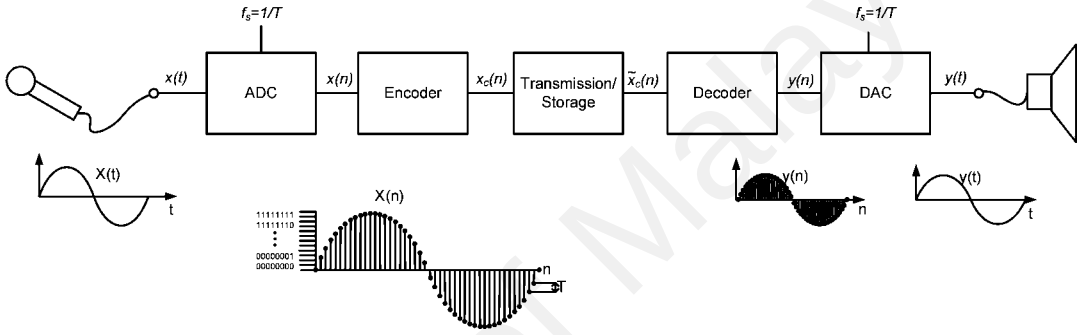


Figure 2.2: Digital Audio Signal Processing Pipeline

2.2.2.1 Microphone recording scenario model

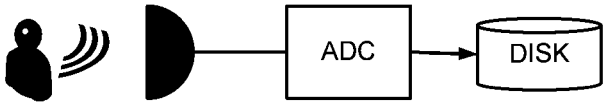
The microphone recording scenario model records playback of the reference files in front of the microphone (i.e. telephone handset, cell phone), whereby the reference files may include music, speech, noise or silence. Alternatively, the microphone may perform live records from different types of audio sources such as speakers or musical instruments. Figure 2.3 demonstrates the differences between signal flow prior to storing the audio file.

2.2.2.2 Call recording scenario model

Traditionally, communication systems are limited to PSTN (i.e. telephone handsets) and cellular technology (i.e. cell phones). However, the development of VoIP technology enabled the employment of the IP telephony network in addition to a variety of computer-based devices containing VoIP software, which are connected to the World Wide Web

Internet. At this point, anonymity, ease of access and free offerings of VoIP, irrespective of geographic distance and location, provide fertile ground for criminal activities. Moreover, during the last decade, most smartphones and mobile devices can facilitate both cellular and VoIP calls. Hence, a call may traverse any of the aforementioned networks before reaching its destination. For example, some organizations record their employees' conversations with customers to improve quality and service; furthermore, in some circumstances, the criminal, non-criminal or law enforcement organization, or an individual, creates evidence against other individuals or organizations by recording their conversations. Thus, the mobile or computer devices identified at the crime scene or during law enforcement investigation may contain several call recording files, whereby authenticating and analyzing those files enables the discovery of information key to resolving judiciary cases. Figure 2.4 illustrates the general call recording setup irrespective of communication or appliance type, whereas the signal processing for PSTN, cellular and VoIP communication differs in terms of specific components.

(a) Signal flow in microphone recording playback audio.



(b) Signal flow in microphone recording live audio.

Figure 2.3: General Recording Set-Up for Microphone Recording.

2.3 Related Fundamentals on Audio Mining Techniques

The key aspect of audio source device identification approaches is the analysis of the relationship between digital evidence and the digital device through the use of audio mining techniques. Audio mining detects important patterns from a large audio dataset to discover the knowledge hidden within. The required information hidden in the audio

signal is determined in parameterize form known as the audio feature. These approaches inputted the extracted audio features into machine learning algorithms to discriminate the audio samples relating to source device (i.e. microphone model) classes. As discussed in Section 2.1.2, all audio source device identification approaches include six hierarchical audio mining stages.

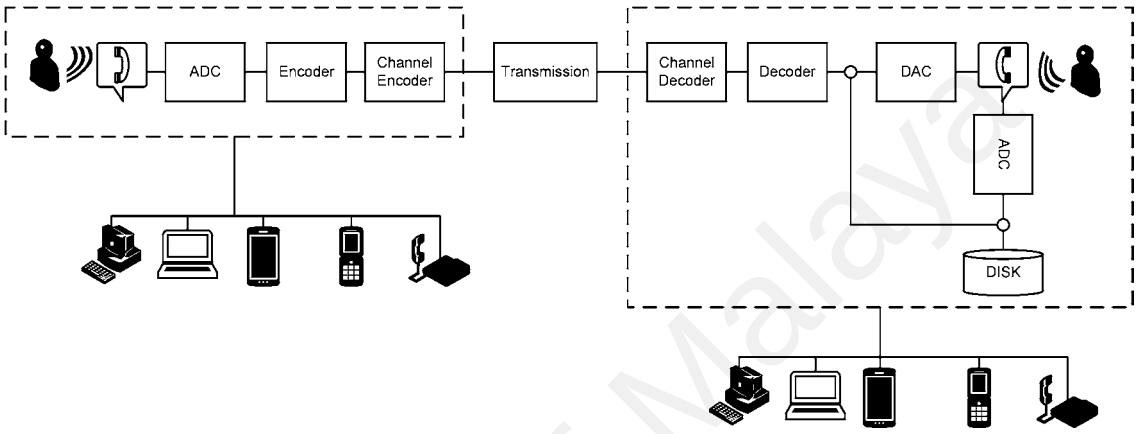


Figure 2.4: General Recording Set-Up and Signal Flow for Call Recording.

2.3.1 Domain Understanding

Domain understanding defines the case scenario, problem domain, and forensic examination objectives. This process is important because audio forensic approaches with signal-based classification can only determine strong feature sets with the representation of knowledge from various stages of signal processing and audio content (i.e. speech, music). For example, Gupta et al. (2012) specified the sources that left unintended signal variations during each stage in the digital audio processing pipeline as detailed in Figure 2.5, whereas these modifications lead to class specific features based on different audio forensic examination techniques. These features may describe any of the following effects: (a) environmental effects, (b) components effects, (c) ENF properties, (d) speaker characteristics, (e) quantization effects, (f) compression effects and (g) double compression effects.

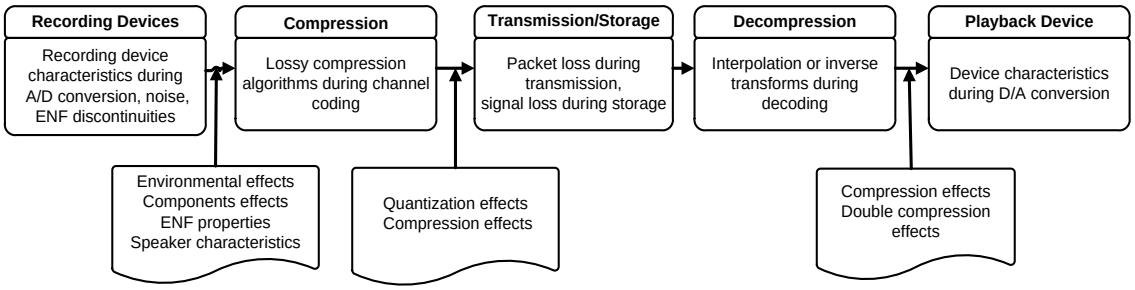


Figure 2.5: Audio Processing System Architecture

2.3.2 Data Selection

In order to successfully estimate the pattern among the samples and extract the required information, a database containing samples from a large number of objects that constitute a representative sample of the relevant population is required. The degree of validity and reliability of the data mining approach depends on the number of samples from each object in the database. Audio mining approaches use available databases such as speech corpora from linguistic data consortium (LDC) or collect audio samples for specific evaluation purposes. The main part of the sound samples applied in the recognition experiments are collected from various sound libraries. Table 2.1 summarizes the list of most prevalent speech corpora and audio databases in the state-of-the-art audio source device identification approaches.

2.3.3 Pre-processing

Pre-processing includes data preparation, enhancement, and transformation to remove possible missing values, unrelated data, or large values. The determination of the audio features requires completion of the pre-processing stages prior to measuring the parameters. The pre-processing stage at first performs digitization and segmentation of the audio data as units of analysis, blocks or instances. This process includes sampling process, framing the signal, windowing, discrete Fourier transform (DFT) and spectral estimation. It also includes audio activity detection and enhancement techniques to extract

the desired content of the signal. This is because audio data consist of a mixture of speech, music, different types of environmental sounds and silence, as classified in Figure 2.6.

Table 2.1: Available Speech Corpora for Signal Processing and Analysis

Name of dataset	Data Source	Description of application domains
NIST (National Institute of Standards and Technology)-SRE (Speaker Recognition Evaluation) (2006)	Telephone speech, Microphone speech	The speech data supports the development of robust speaker recognition technology by providing carefully collected and audited speech from a large pool of speakers recorded simultaneously across numerous microphones and in different communicative situations and/or in multiple languages. The data is mostly English speech but includes some speech in Arabic, Bengali, Chinese, Hindi, Korean, Russian, Thai, and Urdu. The telephone speech segments are multi-channel data collected simultaneously from a number of auxiliary microphones.
NIST-SRE (2008)	Telephone speech, Microphone speech	For this dataset, the participants are native English and bilingual English speakers. The telephone speech in this corpus is predominately English but also includes the variety of other languages. All interview segments are in English. Telephone speech represents approximately 565 hours of the data, whereas microphone speech represents the other 75 hours.
NIST-SRE (2010)	Telephone speech, Microphone speech	This dataset is developed for evaluation of speaker detection in the context of conversational speech over multiple types of channels. Hence, in addition to the telephone speech recording over ordinary telephone channels, the speech recordings are collected over a room microphone channel, and conversational speech from an interview scenario recorded over a room microphone channel.
TIMIT (Texas Instruments and Massachusetts Institute of Technology) (Garofolo et al., 1993)	Microphone speech	This database designed to provide speech data for the acquisition of acoustic-phonetic knowledge and for the development and evaluation of automatic speech recognition systems; it consists of 630 speakers from 8 major dialects of American English, each reading 10 phonetically rich sentences.
HTIMIT (Handset-Texas Instruments and Massachusetts Institute of Technology) (Reynolds, 1997)	Microphone speech	This database was collected by playing a subset of the clean TIMIT corpus that consists of 384 TIMIT speakers (192 males and 192 females). Nine telephone handsets were used including four carbon-buttons, four electrets, and one cordless telephone handset.
LLHDB (Lincoln-Labs Handset Database) (Reynolds, 1997)	Microphone speech	This database consists of 53 speakers (24 males and 29 females) along with nine telephone handsets from HTIMIT corpus and recorded three types of speech for each handset.

For more than two decades, researchers have been developing different content-based audio retrieval such as segmentation (Haque & Kim, 2013), sound classification

(Dhanalakshmi et al., 2011), VAD (Saeedi et al., 2015), and sound quality enhancement (Martinez et al., 2015) techniques for improving the classification accuracy, computational efficiency and robustness of audio mining techniques.

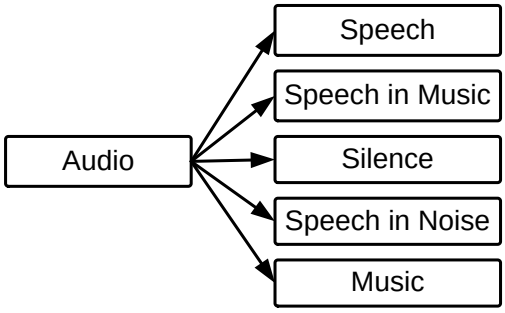


Figure 2.6: Hierarchy of Audio Segments

2.3.4 Feature Extraction

The required information hidden in the audio signal is determined in parameterize form known as the audio feature. Due to diverse nature of the audio features, it is hardly possible to find the ambiguous, applicable and general taxonomy for audio features. Tzanetakis (2002) proposes the audio feature taxonomy for music information retrieval based on two main principles: (a) computational aspect of features and (b) feature attributes. Alternatively, Peeters (2004) introduced specific factors for classification of the audio features, as demonstrated in Table 2.2. This project developed and used a large set of audio features for sound description, similarity, and classification. Although this approach is moderately adaptable to general audio retrieval approaches, it lacks more systematic principles for classification of audio features. To overcome these limitations, Mitrović et al. (2010) proposed one of the most systematic categorizations of audio features. They introduced multidimensional principles in regard to feature properties, the carried information, and the extraction process to avoid ambiguities as much as possible in audio feature's classification. Table 2.3 summarizes the principles used for categorization of audio features in this work.

Table 2.2: Audio Feature Classification Factors (Peeters, 2004)

The steadiness or dynamicity of the feature	i.e. Mean, Standard Deviation, Derivative or Markov model of the parameter
The time extent of the description provided by the features	i.e. Attack of the sound (Instantaneous), Loudness of a note (Global)
The "abstractness" of the feature	i.e. Cepstrum and linear prediction
The extraction process of the feature – features computed on the waveform data – features extracted after performing the transform of the signal – features that relate to a signal model – features that try to mimic the output of the ear system	 i.e. Zero-cross rate i.e. Spectral centroid i.e. Sinusoidal model or the source/filter model i.e. Bark or erb bank filter output

Table 2.3: Multidimensional Principles of Audio Features (Mitrović et al., 2010)

Principles	Values
Signal representation	linear coded, lossily compressed
Domain	temporal, frequency, correlation, cepstral, modulation frequency, reconstructed phase space, eigendomain
Temporal scale	intraframe, interframe, global
Semantic meaning	perceptual, physical
Underlying model	psychoacoustic, non-psychoacoustic

The most basic principle of the audio feature is its signal representation. This principle allows to group the audio features into linear coded signals, and features that perform on lossily compressed audio signals. Most feature extraction approaches function on linear coded signals. However, because moving pictures experts group (MPEG) audio compressed signals are widely used, there has been some research on lossily compressed audio signals. Lossily audio compression transforms the signal into frequency representations by employing psychoacoustic models. This technique eliminates the information from signals that is not perceptible to the human listener. In addition, the generated features may reduce the computational time. The second principle is the audio feature domain, which is the spatial representation of the feature after feature extraction. This principle analyzes the feature data gives details of the feature extraction process and computational complexity. The audio features are represented in any of the following domains:

(a) *Temporal domain*: This domain also known as the time domain represents the signal changes over time. Temporal domain features are computationally light; however, they contain irrelevant information.

(b) *Frequency domain*: This domain demonstrates the spectral distribution of a signal when for each frequency provides the corresponding magnitude and phase. Frequency-domain features capture transfer signal variations from the membrane such as vocal tract; however, they hardly handle convolutions. The convolution exists due to the multiplication of different transfer functions resulted from sources such as speech, environment, and devices.

(c) *Correlation domain*: This domain represents temporal relationships among signals. The autocorrelation domain reveals the correlation of a signal with a time-shifted version of the same signal for different time lags.

(d) *Cepstral domain*: The features in this domain are determined by computing the Fourier transform of the logarithm of the magnitude of the spectrum. Cepstral domain features possess the same ability as frequency domain features in capturing the signal variations, however, the advantage of this representation is the use of summation amongst the convolved signals.

(e) *Modulation frequency domain*: This domain contains information about the temporal modulations contained in a signal.

(f) *Reconstructed phase space*: The convolutions in signals are reconstructed by embedding the signal into the high dimensional phase space, where each point is correlated with the specific state of the system.

(g) *Eigendomain*: The eigendomain features are spanned by eigen- or singular vectors. Eigendomains are generated with different transformations and decompositions, for example, principal component analysis (PCA), and singular value decomposition to

statistically independent components. The significance of this domain is that it reduces the size of data to most optimal features.

The third principle is the feature temporal scale (inter-frame, intra-frame, and global). Intra-frame features operate on independent short-time frames of audio. Inter-frame features operate on the larger temporal scale to describe the temporal change of the signal. A global feature represents the complete length of the audio signal. The fourth principle is semantic meaning that determines if the feature represents aspects of human perception. The last principle is based on the feature extraction that benefits from psychoacoustic models to improve the information content of the features and to approximate human similarity matching. Mitrović et al. (2010) also discussed the general components of audio feature extraction based on three mathematical groups of functions and argued that these three operations are the building blocks in any feature extraction:

(a) *Transformation*: This operation maps data from one domain to another. For instance, DFT maps data from the temporal domain to the frequency domain.
(b) *Filters*: This operator maps a set of numeric values to other sets of numeric values that exist in the same domain, whereby it maintains the number of features.
(c) *Aggregations*: This operation maps a series of values to a singular scalar. The aggregation aims to reduce the dimensionality of the data.

2.3.5 Feature Selection

Because the audio features contain minimized and limited part of the information, it is plausible to determine the most class specific features from the combination of different feature types. In spite of this, the addition of irrelevant or confusing features often confuses pattern recognition systems. Hence, the redundant features should be removed prior to classification to optimize the classification accuracy achieved. Meanwhile, reducing the dimensionality of the feature space minimizes the computation time and

complexity. The process of selecting the most class specific features for the problem is known as feature selection. It is more secure to manually select features based on the proper understanding of the learning problem and the actual meaning of the features. However, when the necessary information is unavailable, automatic feature selection is possible. Witten et al. (2011d) proposed two fundamentally different approaches for selecting the optimal features. The first approach is known as the filter method because it performs an independent evaluation based on general characteristics of the data and develops the most promising subset by eliminating the most redundant features prior to initiation of the learning stage. The second approach is represented by the wrapper method because it evaluates the subset through the machine learning algorithm that wrapped into the selection procedure and will eventually be utilized for learning.

Meanwhile, more advanced methods for feature evaluation are symmetric uncertainty, PCA or random projection, which eliminate redundancies in the set in addition to removing insignificant or vague features. Symmetric uncertainty method applies entropy and joint entropy among features and then conducts a correlation-based feature selection (i.e. (Hanilci & Kinnunen, 2014)). The PCA method transforms the data linearly into lower dimensional space at the expense of large computations (i.e. (Buchholz et al., 2009)). The random projection projects the data into a subspace with a fixed number of dimensions with much less computational cost compare to PCA (i.e.(Panagakis & Kotropoulos, 2012a)). An alternative approach to the introduced scheme is the application of normalization methods such as statistical measures of mean, variance, skewness and kurtosis to determine the suitability of features and reduce the dimensionality of the feature space.

2.3.6 Feature Analysis

Audio mining allocates different styles of learning that best fits into the structure of datasets. Learning techniques include: (a) classification learning, (b) association learning,

(c) clustering, and (d) numeric prediction (Witten et al., 2011a). Classification learning is also known as supervised learning because of the supervision provided by using a set of classified examples to learn the steps of classifying unknown examples. Association learning observes any association between features, not just the one predicts the specific class value. They differ from classification rules based on the fact that they can predict any attribute in addition to the class, and also more than one attribute's value at a time. Thus, they are more association rules than classification rules. Clustering is also known as unsupervised learning because there is no specified class, and it groups examples that belong together as clusters. Numeric prediction is a variant of classification learning that predicts the outcome as a numeric quantity. All techniques incorporate a great deal of standard statistical techniques that used to validate learning models and to evaluate learning algorithms. Table 2.4 and Table 2.5 reveal the list and description of different basic and advanced machine learning techniques, correspondingly. Furthermore, Table 2.6 presents lists and description of advanced clustering algorithms.

An alternative to aforementioned techniques is semi-supervised learning. This technique utilizes the clustering algorithm followed by a second step of classification learning, in which rules are learned based on an intelligible description of how new instances should be placed into the clusters. Using a more advanced approach, different models learned from the dataset are combined to produce an ensemble of learned models. Ensemble learning can be very powerful by transforming a relatively weak learning scheme into an extremely strong one. The most common ensemble learning methods are bagging, boosting and stacking. Although these methods increase predictive performance over a single model, they are difficult to interpret. Bagging combines the decision of different models by taking a vote for classification and calculating the average for numeric prediction, whereby the models receive equal weight. Boosting is similar to bagging when weighting is used to give more influence to the more successful ones.

Table 2.4: Basic Machine Learning Algorithms

Algorithms	Specifications	Selection criteria
1R (1-rule)	A one-level decision tree that expressed in the form of a set of rules, which all test one particular attribute	The simplest rule-based classification learning algorithm for discrete attributes
Naïve Bayesian method	It is based on Bayes' rule of conditional probability that assumes independence	Super simple, suitable for less training data, converge quicker than discriminative models
Decision Trees	The divide-and-conquer approach	Applicable to both continuous and discrete attributes, easy to interpret and explain, non-parametric
Covering Algorithms	At each stage identifies an accurate rule that covers some of the instances	Improves accuracy compared with 1R
Association Rules	Generates [a]item sets with the specified minimum [b]coverage, and from each item, set determines the rules that have the specified minimum [c]accuracy	Applicable for finding multi-correlated items in transactions
Linear Regression	Express the class as a linear combination of attributes	It provides good probabilistic interpretation, unlike decision trees or support vector machines, but shows limited complexity on the constructed boundary, allows to easily update models to take new data, allows to easily adjust classification thresholds or to get confidence intervals
Logistic Regression	Builds a linear model based on a transformed target variable	
Perceptron Learning Rule	Finds a separating [d]hyperplane	Guaranteed to converge in a finite number of steps if the problem is separable, but maybe unstable otherwise
Nearest-Neighbour Instance-Based Learning	Determines the Euclidean distance, and sets the training set as KD-tree or ball tree to find the nearest neighbor	No training is needed; confidence level is achievable, but classification accuracy is low and requires large storage
Unsupervised Learning Distance-Based k-means Clustering	Choose *k* points at random as cluster centers. All instances are arranged relating to their distance to the cluster centers. Then the cluster centroid is updated based on the *mean* of the instances in each cluster. The process continues until the cluster centers stabilize.	One of the simplest unsupervised learning algorithms that solve the well-known clustering problem, order independent (partitions the data irrespective of the order in which the patterns are presented to the algorithm)

Note:

[a]combinations of attribute-value pairs with minimum coverage. [b]number of instances that the algorithm predicts correctly. [c]number of instances to which the rule applies. [d]decision boundary plane.

Table 2.5: Advanced Machine Learning Classification Algorithms

Algorithms	Specifications
C4.5 or J48 algorithm	Restricts the numeric values to a two-way binary split; provides the solution for missing values; simplifies the fully expanded decision trees by post-pruning; converts decision trees to the classification rules
Classification Rules	Attempts to maximize the correctness of the rule on the basis that the higher the proportion of positive examples it covers, the more correct a rule is, treats numeric values and missing values the same way as trees, generates rules using incremental reduced-error pruning, or tree-building algorithm, uses rules with exceptions to ignore rules with deeper structure and focus only at first level or two
Association Rules	Treats numeric and missing values as in trees build a frequent pattern tree to store a compressed version of the dataset in main memory, finds large item sets that meet the minimum support threshold
Support vector machines (SVM)	Uses linear models to implement nonlinear class boundaries, finds the maximum-margin hyperplane when the instances closest to it are called support vectors, applies to non-linear problems by using a [a]kernel function that performs non-linear mapping i.e. *radial basis function (RBF) kernel* and the *sigmoid kernel*, sets a user-specified parameter ε to control how closely the function will fit the training data, and *C* restricts the influence of the support vectors on the shape of the regression function. The larger *C* is, the more closely the function can fit the data.
Multilayer Perceptron (Neural Network)	Assumes the network output's layer has just one perceptron or units, for more than two classes. A separate network could be learned for each class that distinguishes it from the remaining classes. The perceptron layer with no direct connection to the environment is called hidden unit, given a fixed network structure *backpropagation* algorithm determines the appropriate weights for the connection in the network. However, for the hidden units correct outputs are unknown, thus *gradient descent* algorithm modifies the weights of the connections leading to the hidden units based on the strength of each unit's contribution to the final prediction. At last, it provides the solution for overfitting through *early stopping algorithm* or *weight decay.*
RBF Network	Contains two layers without the input layer, and differs from a multilayer perceptron in the way that the hidden units or RBFs perform computations, but the output layers are the same as multilayer perceptron. Each hidden unit represents a particular point in input space, and its output for a given instance depends on the distance between its point and the instance, which is just another point. It learns the centers and width of the RBFs and the weights used to form the linear combination of the outputs obtained from the hidden layer.
Nearest-Neighbour Instance-Based Learning	Enhances the algorithm by pruning noisy exemplars, because they repeatedly misclassify new instances, learns the relevance of each attribute incrementally by dynamically updating feature weights. After classifying new instances, it computes its distance to a [b]hyperrectangle by using the generalized distance function. Then the method merges the function with the nearest exemplar of the same class, which is a single instance or a hyperrectangle. If the new hyperrectangle performed the wrong prediction, boundaries are altered so that it shrinks away from the new instance.
Linear Model Tree	Applies numeric prediction with local linear models divides the instance space into regions similar to normal decision trees and finds a linear regression model for each of them.
Locally Weighted Linear Regression	Introduces an alternative approach to numeric prediction; Generates local models at prediction time by giving higher weight to instances in the neighborhood of the particular test instance.
Bayesian Networks	Constructs a function for evaluating a given network based on the data and a method for searching through the space of possible networks. The quality of a given network is measured by its [c]log-likelihood (LL). It provides a solution to overfit problem by using local scoring metrics such as [d]*Akaike Information Criterion* (AIC) or [e]minimum description length (MDL); uses learning algorithms such as *K2*, *tree-augmented Naïve Bayes*; uses a structure known as *all dimensions tree* for fast learning.

Note:

[a]$K(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x}) \bullet \Phi(\mathbf{y})$, where Φ is a function that maps an instance into a high-dimensional feature space. [b]rectangular regions of instance space. [c]the sum of the logarithms of the probabilities that the network accords to each instant. [d]AIC Score= $-LL + K$, where K is the number of parameters. [e]MDL Score= $- LL + K/2 \log N$, where N is the number of instances in the data.

Table 2.6: Advanced Machine Learning Clustering Algorithms

Algorithm	Specifications
Hierarchical Clustering	The *dendrogram* hierarchy produces a binary tree that forms an initial pair of clusters and then recursively splits each one further if required, *agglomerative* clustering starts with individual instances and successively joins them up into clusters relating to the measure of distance among them. The distance measure determines by algorithms such as *single-linkage* (the distance between their two closest members), *complete-linkage* (all instances in their union are relatively similar), *centroid-linkage* (the distance between centroids), *average-linkage* (average distance between each pair of members of the two clusters), *group-average* (the average distance between all members of the merged cluster)
Incremental Clustering	Forms a tree with instances at the leaves and a root node that represents the entire dataset, starts the tree with a root node and adds the instances one by one, updates the tree by finding a right place to put a leaf representing new instance or radical restructuring the tree. The radical restructuring includes *merging* (merges to nodes into a single cluster before adding new instances), *splitting* (takes a node and replaces it with its children). The algorithm performs updating based on a quantity known as *category utility* that measures the overall quality of a partition of instances into clusters. The category utility is measured based on an estimate of the mean and standard deviation of the value of that attribute. However, it requires supplying an artificial minimum value for the standard deviation of clusters. It uses a cutoff to suppress the growth when the increase in category utility from adding a new node is sufficiently small.
Probability-Based Clustering	Uses a statistical model called *finite mixtures*. A *mixture* is a set of *k* probability distributions, representing *k* clusters, which govern the attribute values for members of that cluster. The *expectation-maximization* (EM)-algorithm calculates the cluster probability distributions, which are the "expected" class values and then calculates the distribution parameters, which are "maximization" of the likelihood of the distributions given the data available. This clustering approach models the dataset that contains correlated attributes by bivariate normal distribution to model such attributes as covarying, it also allows different distributions such as *normal, log-normal, log-odds, Poisson* distribution.
Bayesian Clustering	Uses *finite mixture models*, when every parameter has a prior probability distribution, whenever a new parameter is introduced, its prior probability must be incorporated into the overall likelihood figure. To improve the overall likelihood, the algorithm introduces Laplace estimator to use a particular prior distribution for the introduction of new parameters. AutoClass is a comprehensive Bayesian clustering scheme.

Alternatively, stacking is applied to models built by different learning algorithms. This algorithm aims to learn the most reliable classifiers by using the meta-learning algorithm, which differs with voting. The algorithm discovers how best to combine the output of the base learners. In overall, ensemble learning is an extensive topic that introduces sophisticated learning algorithms among data mining approaches, yet more detail in this topic hardly fits into the context of this study. The specifications on different algorithms

in Table 2.4, Table 2.5 and Table 2.6 result from comprehensive details on learning techniques exist in (Witten et al., 2011b, 2011c).

2.3.7 Decision Making

Decision-making process determines the final decision based on the knowledge gained through performance evaluation by the data. The samples of the underlying problem are arranged as training and test data. The most common evaluation approach is to measure the classification performance in terms of error rate on a dataset that played no part in the formation of the classifier. This independent dataset is called the test set. It is important that the test data has no contribution creating the classifier. Availability of the large enough training and test data allows the better classifier and more accurate error estimate, respectively. However, in case the amount of data for training and testing is limited cross-validation is the most practical solution. In n-fold cross-validation, the fixed number of folds (*n*) or partitions of the data are decided. Thus the data is partitioned into *n* approximately equal folds. By completing *n* rounds, each fold should be used once for testing, when the remaining folds are used for training. The standard method for predicting the error rate of a learning technique, given a single and fixed sample of data, is to use stratified ten-fold cross-validation. Moreover, in order to justify choices made by the proposed solution, different learning schemes should be compared against each other on the same problem to find out which one performs better. The evaluation factors are: (a) estimating the error rate or classification accuracy, (b) counting the cost of error, and (c) determining the performance metrics. The evaluation offered by just estimating the error rate is hardly enough, especially for numeric predictions. In addition, in terms of optimizing the classification rate, it is important to determine the cost of making wrong classifications. For example, multiclass classification problems reveal the result on a test set in terms of a two-dimensional confusion matrix with a row and column for each class. Each matrix element shows the number of test examples for which the actual class is the

row, and the predicted class is the column. The ideal confusion matrix contains large numbers down the main, diagonal and small, ideally zero, off-diagonal elements (Witten et al., 2011d).

The evaluation based on the cost also includes graphical techniques such as receiver operating characteristic (ROC) curves. The significance of these techniques is its ability to construct the performance of the different learning scheme against each other for comparison. ROC curves demonstrate the performance of a classifier without consideration of the class distribution or error costs. This curve demonstrates the true positive rate on the vertical axis versus the true negative rate on the horizontal axis. In overall, the large area under the ROC curves determines the higher rate of correct classification. Moreover, the evaluation based on the performance metrics uses predicted values on the test instances and the actual values to calculate metrics such as mean error and correlation coefficients (Witten et al., 2011d).

2.4 State-of-the-art in audio source device identification

Audio source device identification approaches differ with respect to the device type (i.e. microphone, telephone handset, mobile device), the nature of the audio material (i.e. microphone recording or call recording of acoustic sound, music or speech), and the application scenario (i.e. interpretation through identification, authentication through verification). Thus, Figure 2.7 demonstrates the categorization of the existing methods according to these differences.

2.4.1 The Evolutionary Body of the Research

On the basis of the existing audio source device identification approaches, the evolutionary body of research is illustrated in Figure 2.8 by using the phylogenetic tree. Phylogenetic trees demonstrate the evolutionary relations through affiliations among species (Coelho et al., 2007). The tree was constructed by assuming the audio source

device identification approaches to be a multi-objective optimization problem. The multi-objective optimization problem achieves more than one optimization solution at the same time, whereas the concept of a single optimal solution is inefficient. The audio source device identification approach requires three distinct optimization criteria: (1) classification accuracy, (2) computational efficiency and (3) robustness. Therefore, it is essential to optimize the objectives in an iterative manner and to identify an optimal solution.

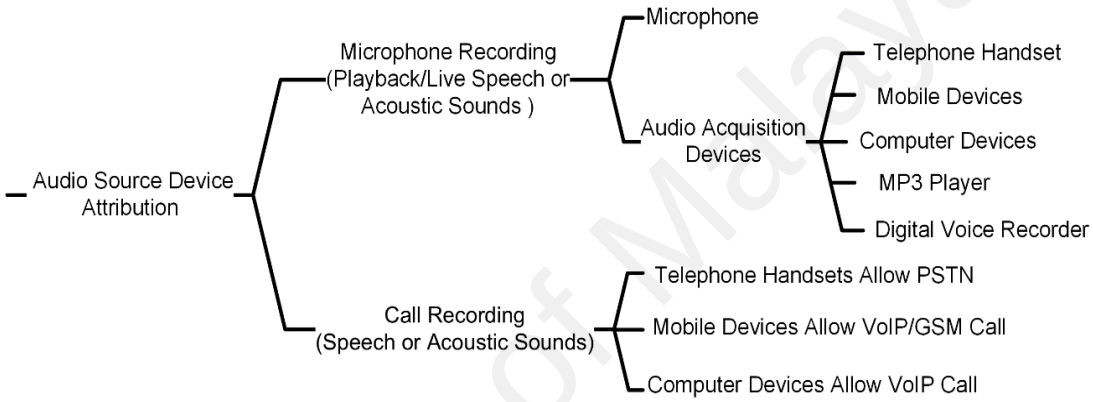


Figure 2.7: Classification of Audio Source Device Identification Approaches

The tree nodes illustrated in Figure 2.8 represent the selected works reviewed for this study with respect to their history (vertical axes) and progression (horizontal axes). The tree consists of one main root and three sub-roots. The first practical microphone classification approach in (Kraetzer et al., 2007) represents the main root for audio source device identification. The authors introduced a set of 63 statistical features, including 7-time domain and 56 Mel-cepstral domain based features for microphone and environmental classification. Furthermore, three roots descended from the proposed microphone classification solution:

(a) *Methods proposed*: microphone identification from recorded sound files (i.e. music, noise, silence, and speech), as the example in Kraetzer et al. (2007) and Buchholz et al. (2009);

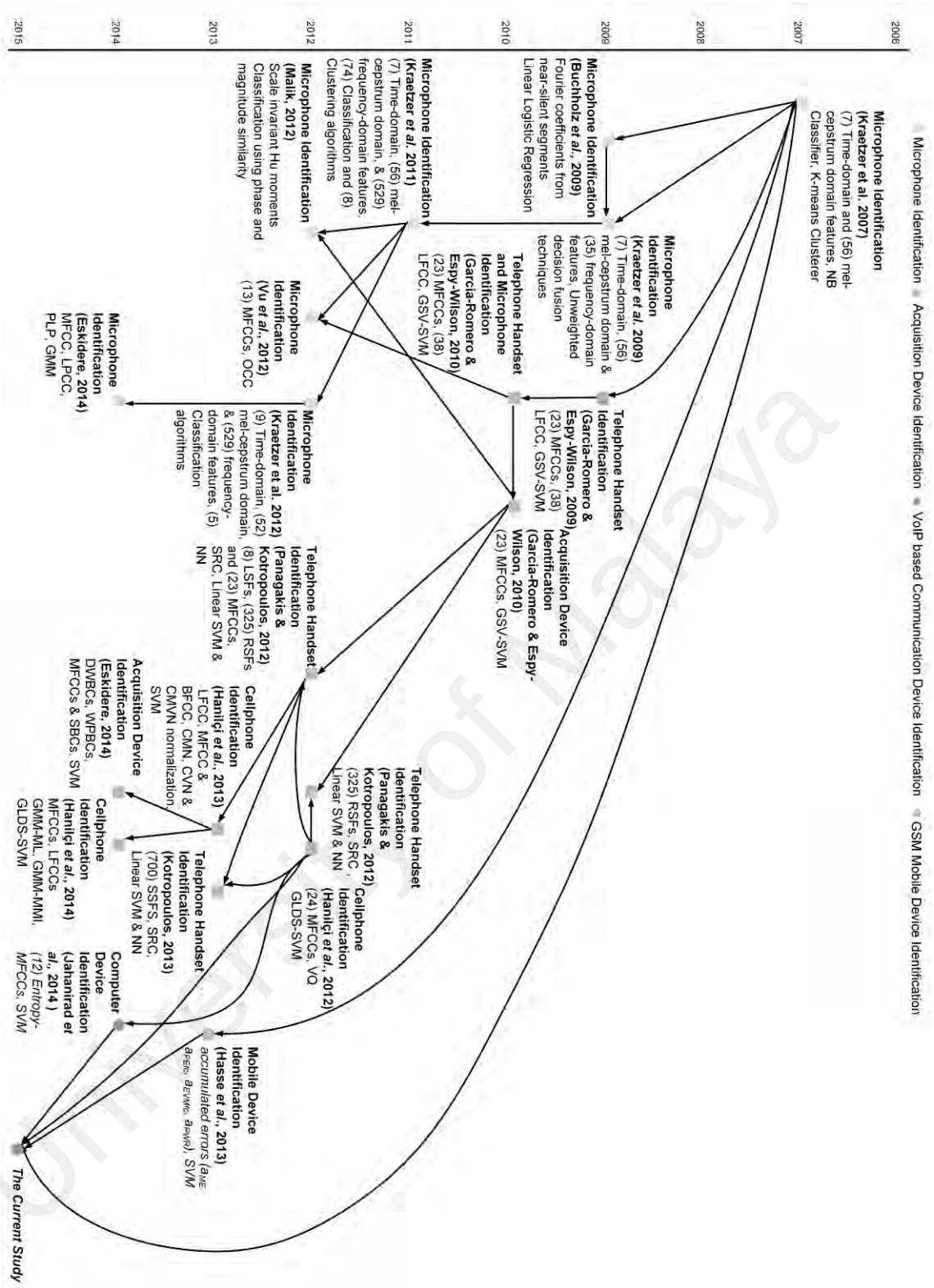


Figure 2.8: Phylogenetic Tree of the Audio Source Device Identification Approaches

(b) *Methods proposed*: audio acquisition device identification from speech recording. For example cell phone recognition approach proposed by Hanilçi et al. (2012);

(c) *Methods proposed*: communication device identification from recorded calls such as (Hasse et al., 2013).

2.4.2 Recording device identification based on microphone recording

Audio source device identification approaches based on microphone recording are classified under two groups. The first group only concentrates on identifying the source microphone, whereas the second group identifies the recording acquisition device.

2.4.2.1 Challenges of source recording device identification

There are a number of issues that make audio source device identification a challenging task. The majority of these issues arise from preparing the sample data to simulate possible real-time forensic scenarios, collecting the evaluation test set from all available devices, and the nature of the application environment:

(a) *Environmental acoustic disturbances*: the recording signal generated by the sound source is often contaminated by various environmental factors before it reaches the microphone. These distortions are categorized into three groups: reflections, reverberation, and the addition of environmental noise. The long-term reflections, with delay times longer than 80ms, create echoes. If multiple long-term reflections exist in the sound field, reaching an energy intensity equal to that of direct sound, the reverberation is triggered (Pawera, 2003). Hence, the classifier performance might be reduced when the training and testing data instances are generated from audio files recorded in different environments.

(b) *Microphone orientation, mounting and aging influences*: in (Kraetzer et al., 2011), influences caused by the orientation of the microphone to sound sources, microphone mounting and possible aging phenomena of the microphone are modeled as multiplicative influences to the microphone recording signal. The classifier performance might be reduced if the microphone orientation, mounting or age of the audio file recording that generated the training data differs from the testing data.

(c) *Audio content dependency*: the speech recording signal contextualized by different speech utterances and speakers' characteristics, and the music recording signal contextualized by different musical instruments' characteristics. Hence, the classifier performance might be reduced when the training and testing data instances are generated from audio files that recorded different sounds.

(d) *Audio playback recording*: the audio playback influenced by the effects of the original microphone recorded the reference signal, in addition to the influence of the loudspeaker playing it. In (Kraetzer et al., 2011), the authors simulated the process of a loudspeaker with multiple drivers playing the audio signal.

(e) *Audio settings dependencies*: the software or hardware recording interface enables the setting of the sampling frequency, quantization mode, sound quality and the file format of the microphone recording signal. The variation of these parameters influence the characteristics of the microphone recording signal. Hence, the classifier performance might be reduced when the training and testing data instances are recorded with different parameter settings and audio formats.

(f) *Dependencies on the dataset*: the classifier performance depends on factors such as the number of audio files and acquisition devices for training, the length of the audio and the number of selected data instances for training each class.

(g) *Sensitivity to audio manipulation*: the classifier performance might be reduced when the training data instances differ from the test data instances as a result of post manipulations such as compression, normalization, noise reduction or phase manipulation on audio files.

(h) *Device-specific features*: the majority of proposed features for audio source device identification are simply adopted from state-of-the-art speech and speaker recognition literature. However, feature extraction techniques best suited to the audio source

device identification systems require analytical solutions for computing device-specific features.

(i) *Computation cost*: the application of kernel-based classifiers such as the support vector machine (SVM) on audio signal processing produces large dimensional kernels that require too much time and space for computation. This is because audio features such as MFCCs are extracted from short audio frames (i.e. 30 ms) so that the training/test data in the set of audio samples comprise a sequence of vectors rather than a single vector.

(j) *Intra- and inter-model similarity*: the proposed features within the literature demonstrated a relatively small intra-model similarity ideal for individual audio source device identification. Hence, no prior work has focused on audio source device brand/model identification. For example, the successful inter-model audio device identification requires audio features with a large intra-model and small inter-model similarity.

(k) *Open set*: for pattern recognition, previous works mainly focused on multi-class supervised learning problems. However, collecting a large audio database corresponding to each device or microphone model is a time-consuming and expensive operation. On the other hand, it is impractical to collect a database of all available microphones or acquisition device models such as the training dataset. Moreover, as illustrated in Figure 2.9, by introducing the new microphone to the market the entire supervised classification model requires rebuilding. Hence, the classification approach should be able to propose a solution based on both closed-set and open set scenarios.

(l) *Classifier benchmarking*: the proposed supervised or unsupervised learning classifier should be able to find a trade-off between the classification accuracy, computation time and robustness for the proposed audio source device identification approach.

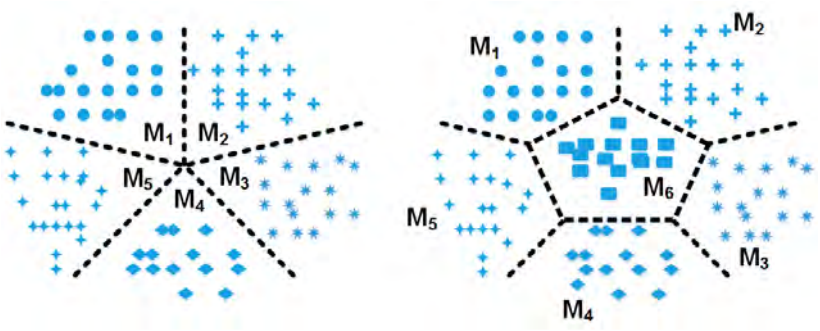


Figure 2.9: Supervised Machine Learning Illustration

2.4.2.2 Microphone identification

The microphone identification approaches focus on the artifacts left by the frequency response function of the microphone on the signal. Table 2.7 presents a comparative overview of the microphone identification frameworks with respect to challenges and resolution strategies, which will be discussed in the following components.

(a) ***Data selection and pre-processing***

Kraetzer et al. (2007) recorded playback of 10 reference files in 10 acoustically different rooms (i.e. small rooms, lecture halls) by using four microphones of different brands. The reference files included different classes of audio material such as music, noise, digital silence, a pure sine wave and recorded speech. Buchholz et al. (2009) extended the database to seven microphones of different models, including microphones of different models but of the same brand. Eight reference files were selected from syntactic samples and short audio clips of popular music to eliminate the effects of the original microphone recording. The samples were collected by playing back the reference files in 12 different rooms with different environmental characteristics.

Table 2.7: Challenges and Strategies for Microphone Identification.

Method	Challenges	Resolution strategies
Kraetzer et al. (2007)	The discriminating power of audio steganalysis features for microphone and environmental classification	Proved that the filtering effect of a microphone is probably stronger and more unique than the filtering effect of the environment
	Controlling the influences of audio	Recorded the training and test sets under
Buchholz et al. (2009)	Fourier coefficients are usually characteristic of the sounds recorded, not of the device recording them; thus, audio contents reduce the device's discriminative power of the Fourier coefficients	Detected near-silent segments of the audio recording and applied feature extractor to those segments
	Large feature space	Applied PCA to the feature space to speed up classification without minimal reduction in accuracy
	Loudspeaker influences	Minimized loudspeaker influences by employing high-quality loudspeaker with higher dynamic range compare to all tested microphones
	Original microphone influences	Used synthetic sound files and short audio clips of popular music comprising mixed sounds from different sources
Kraetzer et al. (2009)	Environment independent classifier	Randomly selected eight of the source files as the reference for all training processes, while the remaining two source files were used as the references for testing
	Classifier benchmarking	Selected fusion strategy on two multiclass classifiers (a decision tree and linear logistic regression models) and increased the accuracy of practical testing up to 100%, showing less confidence in comparison to the single classifier
Kraetzer et al. (2011)	Discriminating microphone effects from other influences	Proposed the context model to evaluate all influences during audio recording process
	Microphone orientation influences Microphone mounting influences	Found minimal impact by microphone orientation to the classification behavior Found strong influence for microphone mounting if it affects the reverberation behavior
	Microphone aging influences	Utilized a recently recorded test set on a training set recorded one year, beforehand, which showed no significant change over time
	Classifier benchmarking	Determined the best performance with the Meta-classifiers, whereas the clustering algorithms showed poor performance for microphone classification

Table 2.7, continued: Challenges and Resolution Strategies for Source Microphone Identification.

Method	Challenges	Resolution strategies
Kraetzer et al. (2011)	Microphone specific features	Achieved very good performance for the 2nd order derivative MFCC based features
	Computation cost	Applied PCA and selected the 20 most significant features to reduce classifier run-time
Kraetzer et al. (2012)	Audio playback detection Sensitivity to audio manipulation	Detected traces of both microphones involved in a recording process Normalization of the audio playback degrades the performance of the playback detection
Malik and Miller (2012)	Microphone non-linearity modeling	Used higher-order spectral analysis (HOSA) for non-linear identification to extract microphone specific features
	Audio content dependency	Recorded noise-like sound
Vu et al. (2012)	Expensiveness in collecting data and updating system for real-world application	Applied open set one class classification (OCC) approach to microphone forensics
	Environmental noise influences	Improved performance of OCC algorithm with representative instance classification framework (RICF) to improve its performance for occasions when the recording signal contains noise
Eskidere (2014b)	Microphone specific features	Investigated the feasibility of linear prediction cepstral coefficients (LPCCs) and perceptual linear predictive coefficients (PLPCs) against MFCCs for microphone identification
	Dataset dependencies	Tested the impact of the different component Gaussian densities, number of training data instances and test utterance lengths
	Audio content dependency	Collected recordings from both speaker-dependent dataset (with similar and different content) and speaker-independent dataset

Subsequently, Kraetzer et al. (2009) employed the reference files and recording rooms in (Kraetzer et al., 2007) to generate two new recording sets based on four and seven different microphone sets in (Kraetzer et al., 2007) and (Buchholz et al., 2009), respectively. A suitable context model for microphone forensics in (Kraetzer et al., 2011) provides a fundamental evaluation of the previous outcomes by formalizing the signal processing pipeline and its influence factors. The proposed context model includes five components, as shown in Figure 2.10. S(f) is the representation of the input audio signal in the frequency-domain, whereas S1 (f) , S2(f), S3(f) and S4(f) denote the audio

signal after each processing step, and S' (f) denotes the final output audio signal. The context model in Figure 2.10 was simulated based upon the process of different components such as the loudspeaker driver amplifying function (F_{driver}) and its thermal noise (N_{ls}), distortion caused by echoes or reverberations (F_{echo}), environmental noise (N_{envi}), the microphone frequency response function (F_{mic}) and its thermal noise (N_{mic}), ENF influence (N_{ENF}), transmission distortion function (F_{tran}), transmission environmental thermal noise (N_{tran}), sampling function (F_{samp}), quantization noise (N_{quan}), thermal noise of the A/D device ($N_{thermal}$). In this work, F_{mic} was estimated based on the microphone's membrane characteristics $F_{membrane}$(*MembCharacteristics*) (its individual vibration behavior and interaction with the other microphone parts), times the overall microphone orientation to sound sources, the microphone's mounting and aging that represented as F_{inf}(*O*, *M*, *A*).

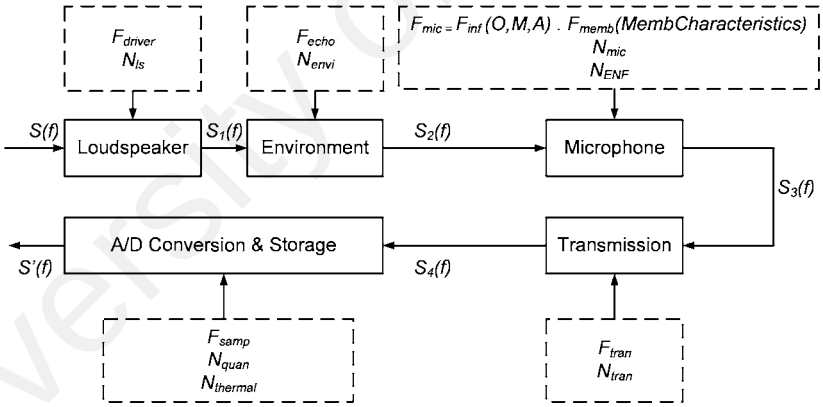


Figure 2.10: A Context Model for Microphone Forensics (Kraetzer et al., 2011).

In order to capture the microphone frequency response and minimize the effects of other components, the authors chose four different microphone recording sets. The first set contains seven different microphones from (Buchholz et al., 2009) for re-evaluation. The second set includes two sets of four identical microphones that record the playback often reference files in (Kraetzer et al., 2007) to allow intra-model microphone identification. The third recording set was collected by using three different microphone types and two different reference sounds in the same room with eight different

orientations. The fourth recording set was collected by recording the same reference files from one microphone through eight different microphone mounting positions. The third and fourth recording set aimed to evaluate the effects of microphone orientation and mounting on recordings by eliminating other influences.

Kraetzer et al. (2012) generalized the context model in (Kraetzer et al., 2011) to construct new application scenarios for microphone forensic investigations on the detection of playback recordings. This approach utilized six microphones, where two were from the same brand and model. The method used four different test set-ups. The first set-up was created in order to evaluate the microphone classification performance, where a male speaker read a specific text in front of the array of six microphones. The second set-up was collected in the same way except for the fact that the sample recordings were normalized at the end. The third set-up was generated for evaluating the playback detection, where the initial recordings were played back and recorded by the same microphone arrays. Lastly, the fourth set-up comprised the normalized version of the playback recordings. The sample recordings were normalized based on a different normalization factor n calculated as the ratio between the maximum possible amplitude value and the peak amplitude of the recording. The set-ups for the playback recording were arranged according to Figure 2.11, whereby S4 represents the normalized audio input to the loudspeaker, S5 (f) , S6 (f) , S7 (f) and S8 (f) represent the audio signal after each processing step and S" (f) , denotes the final output playback signal. The influence factors are the same with the initial recordings. The experiment collected all recordings in a soundproof, anechoic chamber with constant environmental conditions, in short, temporal sequence and with unchanged microphone orientations and mountings. Taking a different approach, Vu et al. (2012) used a set of digital recording devices (three audio recorders and two cameras) with a built-in microphone to collect audio samples. The built-in microphones were from different models, and the audio recordings were

collected in three different recording locations with different noise conditions such as indoors, quiet park and busy street.

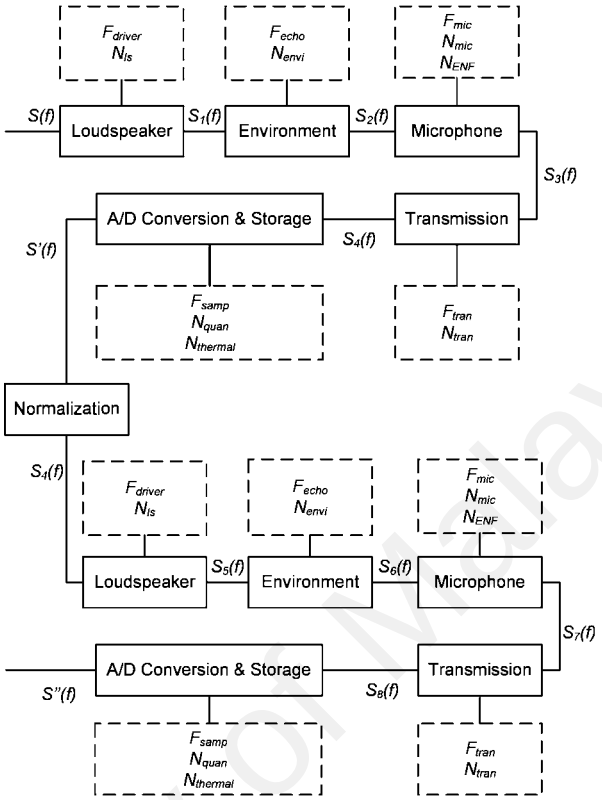


Figure 2.11: A Context Model for the Playback Recordings (Kraetzer et al., 2012).

The microphone identification scheme in (Eskidere, 2014b) utilized three different speech recording datasets, including DS1, DS2, and DS3. DS1 and DS2 were collected by recording two utterances (one with the same content, and the other with different contents) for each microphone in a silent room from a single speaker. DS3 was collected using a TIMIT database (Garofolo et al., 1993) to evaluate speaker-dependent microphone identification performance. This dataset consists of 40 speakers from the test portion of TIMIT, where 120 phonetically-diverse sentences of these 40 speakers were played and recorded by each microphone in the same room. Moreover, this method used 16 different types of microphone, including headsets (behind-the-neck headset, over-the-head, earbud headset), lavalier microphone, and desktop microphone. Malik and Miller (2012) collected a total of 24 audio recordings using 8 microphones (4 microphone pairs) and an 8-channel USB multitrack recorder during three different sessions. The method

utilized a fan operating at the maximum speed as the sound source. Meanwhile, the microphones were precisely located toward the back of the fan in order to minimize any effects from wind noise.

(b) ***Extracting microphone specific features***

Kraetzer et al. (2007) investigated whether known features from the detection of hidden communications in (Kraetzer & Dittmann, 2007) could help to classify the origin of audio streams. The idea was to adopt known statistical features and to evaluate their discriminative power for microphone and environmental classification. The method employed Verifier-Tuple from (Oermann et al., 2005), to structure and analyze information in detail and extract specific features for defined information layers. The feature extraction algorithm computed a set of 63 statistical features by using AAST (advanced multimedia and security lab, Otto-von-Guericke University Magdeburg, Germany (AMSL) audio steganalysis toolset; version 1.03 (Kraetzer & Dittmann, 2007)), as shown in Table 2.8. On the basis of the Verifier Tuple, the proposed features were limited to the particular information layer known as executive semantics. Moreover, there is a little evaluation of the effects on the audio contents when extracting the microphone-specific features.

Consequently, Buchholz et al. (2009) captured the microphone properties by extracting the Fourier coefficients from near silent frames. The method split the audio signal into non-overlapping windows. Meanwhile, the near silent detection algorithm only selected windows on the basis of the fact that the maximum amplitude in the window did not exceed a variable near-silence threshold *t*. The feature extractor transformed the selected windows to the frequency domain by using FFT and utilized the amplitude portion of the complex-valued Fourier coefficients. The algorithm summed up the corresponding Fourier coefficient for all selected windows to determine the Fourier coefficient histogram as being the feature vector with length *n*. Unfortunately, this approach utilized

a large feature set limited to the frequency domain. Thus, Kraetzer et al. (2009) computed a total of 98 features including 7-time domain, 56 Mel-cepstral domain and 35 frequency domain features with A9SL (version 1.04) (Kraetzer & Dittmann, 2008). Moreover, the microphone forensics context models in (Kraetzer et al., 2012; Kraetzer et al., 2011) employed the feature extractor tool AAFE (AMSL audio feature extractor, version v.0.2.5 (Kraetzer & Dittmann, 2010)) and computed a total of 590 intra-frame features. Table 2.9 presents the feature sets in detail.

Table 2.8: List of Features Computed By AAST (Kraetzer & Dittmann, 2007)

Fracture index	Feature label	Feature details
1	sf_{ev}	empirical variance
2	sf_{cv}	covariance
3	$sf_{entropy}$	entropy
4	$sfLSB_{rat}$	least significant bit ratio
5	$sfLSB_{flip}$	LSB flipping rate
6	sf_{mean}	mean of samples in the time domain
7	sf_{median}	the median of samples in the time domain
8	$sfmel_1$	Mel-cepstral domain based features
⋮	⋮	
63	$sfmel_{56}$	

As a result of the good performance of MFCC-based features in (D. Garcia-Romero & Epsy-Wilson, 2010; Kraetzer et al., 2011), Vu et al. (2012) computed all 13 MFCCs as audio features for identifying microphones from noisy recordings. Eskidere (2014b) studied further the microphone properties and their effect on microphone identification, in an attempt to investigate more discriminant microphone-specific features over MFCCs. Hence, the LPCC (Rabiner & Juang, 1993a) and perceptually-based linear predictive coefficients (PLP) (Hermansky, 1990) were utilized because of their widespread use and easy modeling. Taking an alternative approach, Malik and Miller (2012) performed polyspectral analysis to capture microphone-induced properties. Hence, this approach modeled microphone artifacts using a non-linear function. The non-linear function is modeled through HOSA (Farid, 1999) based on third-order cumulants (bispectrum). The

HOSA reveals the amplitude information, in addition to the phase information regarding the underlying process. Thus, for feature extraction, the method computed the bicoherence, which is a normalized bispectrum (the Fourier transform of the third-order cumulant or bicorrelation), and then computed the scale invariant Hu moments (Hu, 1962) of the bicoherence magnitude spectrum to capture non-linear correlations.

Table 2.9: List of Features Computed By AAFE (Kraetzer & Dittmann, 2010)

Categorization	Size	Description
Time domain	9	(1) zero crossing rate, (1) energy, (1) pitch, (1) root mean square-amplitude, (1) entropy, (1) LSB-ratio, (1) LSB-fliprate, (1) mean, (1) median
Frequency domain	529	(1) spectral centroid, (1) spectral roll-off, (2) differently computed spectral bandwidths, (1) spectral irregularity, (1) spectral entropy, (11) formants and base frequency, (512) frequency coefficients histogram
Mel-cepstrum domain	52	(13) MFCCs, (13) fractional mel frequency cepstral coefficients (FMFCCs), and 2nd order derivative of MFCCs (13) and FMFCCs (13)

(c) ***Machine learning approaches for feature analysis***

Kraetzer et al. (2007) determined the classification accuracy based on both supervised (classification made using Naïve Bayes classifiers) and unsupervised (K-means clustering algorithm) learning techniques using the data mining tool Weka (Hall et al., 2009). The Naïve Bayes classifier used two approaches for classification; the first approach utilized 66 percent for training and the remaining 34 per cent for testing, whereas the second approach adopted the 10-fold cross validation. Similarly, Buchholz et al. (2009) employed six different classification algorithms including Naïve Bayes, SMO (a multi-class SVM construct), Simple Logistic (regression models), J48 (decision tree), IB1 (1st-nearest neighbor), IBk (2nd-nearest neighbor) by using the Weka data mining tool (Hall et al., 2009). The classifiers used default parameters and 10-fold cross validation for evaluation. In addition, the PCA reduced the dimensionality of the feature space and computation time without degrading the classification accuracy.

Alternatively, in an optimization approach adopted from state-of-the-art works in the field of biometrics (Ross et al., 2006), Kraetzer et al. (2009) used match-, rank- and decision-level fusions on the output of two different classifiers, simple logistic (SL) and the J48 decision tree, to increase the classification accuracy and confidence in the fused decision for microphone identification. Information fusion locates the best set of experts in a given problem domain and devises an appropriate function that can optimally combine the decisions of the individual experts. The algorithm estimated the specific confidence function for each fusion approach in addition to supervised classification accuracy to improve compatibility of the results. The confidence is computed as a worst-case distance from the decisions to the decision boundary. Kraetzer et al. (2011) implemented a total of 74 classification and 8 clustering techniques using the data mining tool Weka (Hall et al., 2009). This approach evaluated the performance of different learning techniques for microphone classification. Kraetzer et al. (2012) used the five best-performing classifiers in (Kraetzer et al., 2011) for identifying the microphone from the playback recording. The selected classifiers are represented as RotationForest, Multiclass, RandomSubSpace, EnsembleSelection and Logistic classifiers. Meanwhile, the initial recordings prior to and after normalization were utilized as the training dataset and the playback recordings were used as the testing dataset.

To overcome the shortcomings of the closed-set problem and difficulties of implementing the open set problem, Vu et al. (2012) introduced a novel machine learning approach to microphone forensics known as OCC. This approach developed new tools and techniques for more efficient analysis. The OCC classifier trains a model based on the target class that discriminates it from outlier classes built corresponding to all counter-examples. This classifier built a model using a training dataset from each individual microphone at a time, thus generating n OCC models for identifying n microphones. Based on this consideration, there is no need to retrain the classifier when new

microphones are introduced to the database. As a result of its efficiency, the OCC classifier has been implemented in many audio forensic approaches such as sound classification (Rabaoui et al., 2008), scene classification (FengJuan et al., 2010), and speaker verification (Brew et al., 2008). The OCC algorithm classifies a sample as a member of the target class if its assigned score by the OCC model is above a predetermined threshold; otherwise, it is recognized as an outlier class. Figure 2.12 demonstrates the OCC model construction for the microphone classification. This approach adopted a set of OCC algorithms implemented in the Data Description toolbox from (Brew et al., 2008) including: (1) One-class Gaussian model, (2) One-class Gaussian mixture model, (3) One-class K-means, (4) One-class K-nearest neighbors, (5) One-class principal component analysis and (6) One-class incremental support vector machine. Moreover, Vu et al. (2012) proposed an RICF to improve OCC performance on microphone forensic tasks for noisy recording signals. The RICF reduces the effect of noise on the training dataset through sampling microphone-specific data instances from the audio records. This approach used kernel density estimation from (Terrell & Scott, 1992) with the Gaussian kernel, to find the densest region among the training data instances drawn from each microphone record (cluster center).

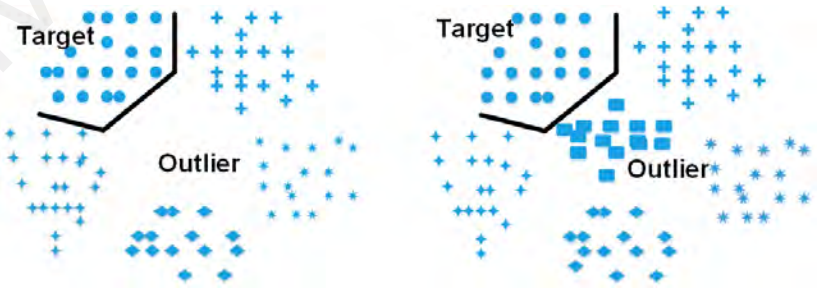


Figure 2.12: OCC Approach Illustration (Vu et al., 2012).

Furthermore, Eskidere (2014b) adapted Gaussian mixture models (GMM) from text-independent speaker identification in (Reynolds, 1997) for pattern classification and data modeling. The author employed the training feature vectors to generate a GMM model

regarding each source microphone. Subsequently, it created the GMM model through a vector quantization (VQ) algorithm for the initial parameter estimate and then estimated the variances by using a diagonal covariance matrix. Afterward, it calculated a simple set of likelihood functions using the test feature vectors and the GMM for each microphone. Finally, it predicted the microphone model where the model with the largest likelihood function indicates the most likely microphone. In an unsupervised learning approach, Malik and Miller (2012) employed distance- and correlation-based similarity measures for automatic microphone identification. More specifically, the distance between scale-invariant Hu moments of the bicoherence magnitude spectrum and cross-correlation between bicoherence phase spectra are used. Eventually, threshold-based multiple hypothesis testing is used for microphone identification.

(d) ***Performance metrics for decision-making***

Kraetzer et al. (2007) computed classification accuracy as a performance metric to evaluate and compare the performance of the microphone and environmental identification under different test set-ups. The classification accuracy determines the percentage of correctly classified instances over all instances. Buchholz et al. (2009) also adopted the same metric to compare the performance of the proposed feature extraction for different threshold values, the percentage of the near-silent segments, frequency coefficients and classifiers. In (Kraetzer et al., 2009), the authors used the confidence estimation function in addition to classification accuracy in order to compare the applicability of the fused classifiers in real world investigations. The authors determined confidence as a measure of how far the fusion decision is away from the complex decision boundary. Moreover, the approach also evaluated the cost of the fusions by assuming that the complexity of a classifier is constant for a fixed number of feature vectors to be evaluated. The match- and decision-level fusions produced linear complexity based on

the number of feature vectors, whereas rank level fusion was highly dependent on its complexity by the number of microphones in the evaluation.

The performance metrics defined in (Kraetzer et al., 2011) include classification accuracy, classification gain (accuracy based on the number of classes), run-time (time duration of the experiment on the test machine expressed, relative to its corresponding strict time out boundaries), and classifier quality (the Euclidean distance from the optimum point in a run-time classification gain diagram). In addition, the method evaluated the performance of a single feature vector by a two-stage ranking fusion. This study also introduced two new metrics when implementing the proposed microphone identification approach for the composition detection of different splicing scenarios. The first metric identifies the change rate in a predefined sequence of frames of the classifiers' decision on the class of the audio material under observation. The second metric identifies the average sequence length for which the classifier returns an answer of the same class. These metrics are suggested to discover whether the wrong model is being used and whether a composition is likely. Kraetzer et al. (2012) evaluated the performance of microphone detection based on live recordings, and its normalized version using the classification accuracy and error rates. However, during the playback detection when the first and second microphones were different, the method developed a new metric known as result classes in an attempt to perform an evaluation. The eight different possible result classes are summarized in Table 2.10.

Taking an alternative approach, Vu et al. (2012) used overall recall and precision rates throughout the various microphone classes to evaluate the detection capability of the OCC algorithms; the fraction of the target class data instances classified correctly as target class is known as recall, whereas the precision, is the fraction of the outlier data instances classified incorrectly as target class. Experiment results showed that the tested OCC algorithms were able to detect the microphone model with high recall and low precision

rates in the indoor or quiet outdoor environment. In (Eskidere, 2014b; Malik & Miller, 2012) the accuracy was computed using performance metrics to evaluate and compare the performance of the proposed source microphone identification approach for different test set-ups.

Table 2.10: List of result classes (Kraetzer et al., 2012).

Result	Classes Description
D1	The best result was achieved for detecting the initial recording microphone and the second best result (<16:67%) was achieved for detecting the playback recording microphone
D2	The best result was achieved for detecting the initial recording microphone and the second best result (>16:67%) was achieved for detecting the playback recording microphone
D3	The best result was achieved by detecting the playback recording microphone, yet the results for both microphones are close
D4	Either the best or the second best result was achieved for detecting the initial recording microphone, but the result for detecting the playback recording microphone was never among the best or the second best
D5	The best result was achieved by detecting the playback recording microphone and the second best result (>16:67%) was achieved for detecting the initial recording microphone
D6	The best result was achieved by detecting the playback recording microphone, and the second best result (< 16:67) was achieved for detecting the initial recording microphone
D7	Either the best or the second best result was achieved by detecting the playback recording microphone, but the result for detecting the initial recording microphone was never among the best or the second best
D8	None of the best two results was achieved for detecting the initial and playback recording microphones

2.4.2.3 Acquisition device identification

Motivated by the first practical microphone identification approach in (Kraetzer et al., 2007), Garcia-Romero and Espy-Wilson (D. Garcia-Romero & Espy-Wilson, 2009) expanded the idea from microphone identification to automatic acquisition device identification based on landlines and telephone handsets. The method assumed that the physical devices, along with their associated signal processing chain, leave behind intrinsic fingerprint traces in the speech signal. This method characterized the model for the handset as a linear system and estimated its frequency response by measuring the output of the system when stimulated by white noise. Table 2.11 presents a comparative overview of the acquisition device identification frameworks regarding challenges. Resolution strategies will be discussed in the following sections.

Table 2.11: Challenges and Strategies for Acquisition Device Identification

Method	Challenges	Resolution strategies
D. Garcia-Romero and Espy-Wilson (2009), D. Garcia-Romero and Espy-Wilson (2010), D. Garcia-Romero and Epsy-Wilson (2010)	Device specific features	MFCCs showed the best trade-off between performance and dimensionality
	Computation cost and speech content dependency	Used GSV as a statistical characterization of frequency domain; information from a device contextualized by speech content
Hanilçi et al. (2012)	Extracting the device-specific portion of the the whole information contained in the recorded speech	Provided analytical justification regarding the feasibility of MFCCs in characterizing the device; showed that MFCCs convert the convolutionally embedded device-specific information in the recorded speech to additive form
	Classifier benchmarking	Compared the performance of VQ against generalized linear discriminant sequence (GLDS) kernel SVM classifier
	MFCCs are a sequence of vectors rather than a single vector	Employed GLDS kernel with SVM classifier to replace a sequence of feature vectors from a speech sample with a single characteristic vector
	Inter-model similarity	Determined significant performance for cell phone recognition among five cell phones of the same brand
	Intra-model similarity	Determined significant distance of the same brand and model of cell Phone pairs by using average squared Euclidean distances of each MFCC feature
Panagakis and Kotropoulos (2012a), Panagakis and Kotropoulos (2012b)	Modeling the impact of the acquisition device on the recorded speech	Computed the mean spectrogram of the speech recording
	Computation cost	Applied unsupervised random spectral features (RSFs) and supervised labeled spectral features (LSFs) feature selection on mean spectrogram of the speech recording to determine the intrinsic traces of recording device and reduce dimensionality
	Classifier benchmarking	Proved feasibility of proposed sparse representation based classification (SRC) against linear SVM, and neural network (NN) classifiers.
(Kotropoulos, 2013)	Computation cost	Proposed sketches of spectral features (SSFs), mapped the mean spectrogram of the speech recording to lower dimensional space, in order to preserve the distance properties of the high-dimensional mean spectrograms

Table 2.11, continued: Challenges and Strategies for Acquisition Device Identification

Method	Challenges	Resolution strategies
Hanilci and Ertas (2013)	Comparison and optimization of acoustic features	Compared zero order, 1st and 2nd order MFCCs, linear-frequency cepstral coefficients (LFCCs), bark frequency cepstral coefficients (BFCCs) and LPCCs with different normalization techniques including cepstral mean normalization (CMN), cepstral variance normalization (CVN), and cepstral mean and variance normalization (CMVN)
Eskidere (2014b)	Fourier analysis is difficult as a result of its sharp localization in time and frequency at the same time	Suggested discrete wavelet analysis for multi-resolution time-frequency analysis and presented two new sets of features: discrete wavelet based coefficient (DWBC) and WPBC
	Feature optimization	Investigated the feasibility of different SMs: Shannon entropy, sure entropy, log-energy entropy, standard deviation, and mean of subband signals
	Justification of feature extraction algorithm	Investigated the effects of different decomposition levels of discrete wavelet transform (DWT) and wavelet packet transform (WPT), and the different order of LP analysis on DWBC and WPBC performance
Hanilci and Kinnunen (2014)	Speech signal influences	Analyzed the amount of device-specific information in speech and non-speech parts using maximum mutual information (MMI) criterion
	Irrelevant information reduces the discriminatory power of MFCC and LFCCs	Proved that information about the source device might be more pronounced in the non-speech parts of the signal
	Classifier benchmarking: GMM-ML, GMM-MMI, and GLDS-SVM	Proved the performance of GMM-MMI over other approaches because the MMI training of GMM maximizes the probability of a correct decision by taking all the training samples of each class into account. The maximum likelihood (ML) training of GMM maximizes the overall likelihood of training data for a cell phone, however, it lacks sufficient data
	Feature optimization	Tested the effects of preprocessing (spectral subtraction) and feature normalization (CMVN)
	text- and speaker independent classifier	Utilized speech recordings from TIMIT database consisting of different speakers with different dialects reading different sentences, and live recording database consisting of the same speakers reading different texts
	Influence of environmental variability on recognition rate	Showed dramatic performance reduction for when each cell phone is trained using its original training samples, and noise is added to the test speech samples

(a) ***Data selection and pre-processing***

In (D. Garcia-Romero & Espy-Wilson, 2009), the authors employed the HTIMIT and LLHDB databases (Reynolds, 1997), comprising short sentences from more than 300 speakers over 10 different acquisition devices. D. Garcia-Romero and Epsy-Wilson (2010) collected speech recordings using four carbon-button and four electret landline handsets from the HTIMIT and LLHDB databases. The microphone database was collected using the NIST-SRE (2006) from international conferences on scientific information, including 8 different microphones and 61 speakers. D. Garcia-Romero and Espy-Wilson (2010) utilized the NIST-SRE (2008) for collecting speech recordings using multiple devices of the same brand and model in different rooms. Similarly, Hanilçi et al. (2012) collected 2 different speech recording datasets using 14 cell phones always situated in the same silent room. The first dataset was collected by playing back speech recordings from the TIMIT database (Garofolo et al., 1993), whereas the second dataset was collected via live recording of the spoken utterances from the same speaker. Hanilci and Ertas (2013) re-evaluated the TIMIT-based speech recordings for testing the feasibility of different acoustic features for source cell phone identification. Hanilci and Kinnunen (2014) adopted both datasets in (Hanilçi et al., 2012) to consider recognizing source Cell Phones using non-speech segments of recorded speech.

Although prior works achieved high recognition accuracy by extracting the features from the whole signal, information about the source device might be more specific in the non-speech segments of the signal. This approach implemented adaptive energy-based speech activity detectors to locate the speech and non-speech parts. This study also investigated the source device identification performance under additive noise conditions. Hence, for additive noise contamination, the authors used a filtering and noise adding tool (FaNT (2015), an open-source tool that follows international telecommunication union (ITU) recommendations for noise adding and filtering. The method selected white and

babble noises from the Noisex-92 (2015) database with three different signal-to-noise ratio (SNR) levels; 0, 5 and 10dB. White noise has constant power spectral density and, specifically, strongly masks the lower-amplitude higher formants of human speech. For speech and speaker recognition studies, white noise is considered to be a difficult-to-handle case (Zilovic et al., 1998). Secondly, babble noise (Krishnamurthy & Hansen, 2009) simulates an unintelligible mix of multiple speakers. For testing the effects of noise contamination, the noise was added only to the testing speech samples in order to apply a mismatch between the training and test conditions. Overall, the cell phone identification methods developed in (Hanilci & Ertas, 2013; Hanilçi et al., 2012; Hanilci & Kinnunen, 2014) focused only on microphone recording rather than call recording to eliminate the complications produced during transmitting and receiving a signal. The approaches in (Kotropoulos, 2013; Panagakis & Kotropoulos, 2012a, 2012b) adopted eight telephone handsets from the LLHDB database (Reynolds, 1997) for automatic telephone handset identification. The methods prepared the training and testing data instances according to experimental set-up in (D. Garcia-Romero & Epsy-Wilson, 2010). Taking a different approach, Eskidere (2014a) utilized 14 different models of portable acquisition devices for the collection of speech recordings. To record the speech recordings through portable computers such as notebooks, netbooks, and ultrabook computers in addition to tablet PCs, the audio editing software Audacity was used. To record the speech recordings through smartphones, the audio recording software Hertz was employed. The speech recording dataset was collected using the TIMIT database followed by the test set-up in (Hanilci & Ertas, 2013).

(b) ***Extracting device specific features***

The studies in (D. Garcia-Romero & Epsy-Wilson, 2010; D. Garcia-Romero & Espy-Wilson, 2009) minimized the effects of speech convolution by using the statistical characterization of the frequency response of the device itself. The method uses 23

MFCCs, 38 LFCCs and their combination with the first order derivative (delta) of a set of both MFCC and LFCC features. D. Garcia-Romero and Espy-Wilson (2010) tested the effects of room acoustics and intra-device variability for the feasibility of MFCCs for acquisition device identification. Similarly, Hanilçi et al. (2012) attempted to extract device-specific information from speech recordings using the speech processing techniques learned from state-of-the-art speaker recognition approaches (Campbell, 2002; Campbell & Assaleh, 2002; Campbell et al., 2007). Their study showed that device identification is identical to speaker identification, and the speech feature extraction techniques can be adopted for cell phone identification. This approach constructed the model by considering the recording components of the cell phone as a linear time-invariant filter with impulse response $h(n)$, and therefore the recorded speech signal $y(n)$ as the output of this filter in response to the original speech signal $x(n)$ given by $y(n) = x(n) * h(n)$. This model defined the impact of cell phones on the recorded speech as a convolutional distortion that requires logarithmic transformation to additive form and hence allows cell phone identification. For this purpose, the proposed method extracted the device-specific information by using the MFCCs for recognition of the brands and models of cell phones. The feature extraction algorithm used 12 MFCCs and their first-order derivatives as the feature set; Hanilci and Ertas (2013) tested different acoustic feature extraction methods such as MFCC, LFCC, BFCC, and LPCC for cell phone identification. This study also evaluated the effect of appending dynamic features, first and second order coefficients, and the effect of applying feature normalizations such as CMN (Liu et al., 1993), CVN (Zheng et al., 2006) and CMVN (Zheng et al., 2006) to the performance of Cell Phone identification. Hanilci and Kinnunen (2014) studied the relative importance of non-speech segments in cell phone identification based on MFCC and LFCC features. Using MMI criterion (Hubeika et al., 2008), the study showed that

features extracted from non-speech parts of the signal contain higher mutual information compared with those extracted from the speech segments.

In an alternative study, Panagakis and Kotropoulos (2012a) proposed that the RSFs were intrinsic to tracing the recording device. Initially, the feature extraction algorithm computed the mean spectrogram of each speech recording, and then the random projection approach used the orthogonal random Gaussian matrix for reducing its dimensionality as detailed in (Bingham & Mannila, 2001). This study evaluated different dimensions of RSFs to determine the best trade-off between feature dimensionality and classification accuracy. Moreover, it used 23 mean MFCCs as baseline features for comparison; thereafter Panagakis and Kotropoulos (2012b) proposed LSFs in addition to RSFs as intrinsic fingerprints suitable for landline telephone handset identification. The RSFs are recognized as unsupervised feature selection algorithms because they reduce the dimensionality of the mean spectrogram by random projections; however, LSFs are defined as supervised feature selections on the mean spectrogram of the recordings. The LSFs contain the label vector that reveals information regarding the device class of the training speech recordings. This algorithm creates a mapping between the feature space and the label space. The mapping is performed by solving a regression problem. This algorithm also maps the unlabeled test's mean spectrogram to the label information adapted from the training dataset. Kotropoulos (2013) enhanced the RSFs by proposing the SSFs as intrinsic fingerprints suitable for device identification. The SSFs' algorithm computes the mean spectrogram of the speech signal and then maps the mean spectrogram into a low-dimension space while preserves the distance properties of the high-dimensional mean spectrogram. The mapping is performed by taking the inner product of the mean spectrogram with a vector of independent identically distributed (i.i.d.) random variables determined using a p-stable distribution (Indyk, 2006; Zolotarev, 1986). Among the wide class of stable distributions, the method selected two classes for computing the

SSFs including (1) Gaussian distribution, (2) Cauchy distribution (Arce & Hoboken, 2005). The Gaussian distribution is the most well-known stable distribution with zero mean and unit standard deviation N (0, 1), and is 2-stable. This distribution was also employed for computing the RSFs in (Panagakis & Kotropoulos, 2012a, 2012b). Cauchy distribution is among the heavy-tailed distributions and is 1-stable.

Eskidere (2014a) discussed the shortcomings of frequency analysis such as short-time Fourier transform (STFT), the difficulty of synchronous sharp localization in time and frequency. Thus, the authors proposed the use of novel wavelet based (Sarikaya et al., 1998) features known as DWBC and wavelet packet based coefficient (WPBC). The method divides the speech signal into overlapping frames, then computes the wavelet packet decomposition and discrete wavelet transform for computing the DWBC and WPBC correspondingly. Subsequently, the method estimates the linear prediction coefficients from the subband signals and converts the linear prediction coefficients to cepstral coefficients. The resulting LPCCs are appended to the statistical measures of the subband signals to provide complementary information. The SMs include various entropies (Shannon entropy, log-energy entropy, and sure entropy), the standard deviation, and mean. For comparison, this study also implemented MFCCs and subband based coefficient (Sarikaya & Hansen, 2000; Sarikaya et al., 1998) features.

(c) ***Machine learning approaches for feature analysis***

For statistical modeling, the works in (D. Garcia-Romero & Epsy-Wilson, 2010; D. Garcia-Romero & Espy-Wilson, 2010; D. Garcia-Romero & Espy-Wilson, 2009) adopted the MIT Lincoln Laboratory's, GMM-based speaker verification system (Reynolds, 1997). This offered a means of generating an only means adapted GMM-universal background model (UBM) architecture with 2048 mixtures and diagonal covariance matrices. The proposed method computed the intrinsic artifacts of the acquisition devices

using the means of the GMM since the Bayesian adaptation process only updates the means of the Speaker's GMM for the UBM. By stacking the means of the GMM, the model constructed a Gaussian supervector (GSV) that represents the speech recording in a high-dimensional vector space (Campbell et al., 2006). The methods used the feature vectors from the training set to build the GMM model, then computed the GSV by using the GMM model and later trained the linear SVM classifier. Finally, the remaining dataset performed classification using log-likelihood ratio scores. Overall, the employment of the GMM model for this work enables the reduction of the dimensionality and redundancy of the feature space in the GSV-SVM classifier. Subsequently, Hanilçi et al. (2012) employed VQ and SVM-based classification for cell phone identification. The authors adopted the application of the GLDS kernel with SVM from a speaker and language recognition approach in (Campbell, 2002). The GLDS method reduced the computation cost by computing a single characteristic vector using the sequence of feature vectors extracted from a speech sample. This method maps the feature vectors into a kernel feature space by a polynomial expansion. In (Hanilci & Ertas, 2013), the GLDS-SVM classifier enabled the evaluation and comparison of the effects of different feature optimization approaches for cell phone identification. Finally, Hanilci and Kinnunen (2014) compared the performance of three different classification approaches for cell phone identification including classical GMM trained in both ML and MMI criteria, in addition to GLDS-SVM.

The studies in (Kotropoulos, 2013; Panagakis & Kotropoulos, 2012a, 2012b) implemented SRC (Wright et al., 2009) for telephone handset identification. The SRC algorithm generated the dictionary atoms using the proposed features extracted from the training speech recordings. The algorithm presented the features extracted from the test speech recordings as a compact linear combination of the dictionary atoms for the device actually used for its recording. This classification approach is known as a sparse

representation as a result of the fact that it utilizes a small fraction of the dictionary atoms and its efficiency can be computed via 1-norm optimization. The SRC algorithm assigns each vector of test instances to the device identity (ID) based on the dictionary atoms weighted by non-zero coefficients associated with it. The studies also employed linear SVM, GSV-SVM and nearest neighbor (NN) classifier (with the cosine similarity measure) and compared their performances against SRC. Eskidere (2014a) employed and tested the SVM classifier to validate the applicability and efficiency of the audio device identification with wavelet based features.

(d) ***Performance metrics for decision-making***

The majority of telephone handset and cell phone identification studies simply adopted classification accuracy as a performance metric to evaluate and compare performance of proposed techniques under a variety of test scenarios (Eskidere, 2014a; D. Garcia-Romero & Epsy-Wilson, 2010; D. Garcia-Romero & Espy-Wilson, 2010; D. Garcia-Romero & Espy-Wilson, 2009; Hanilçi et al., 2012; Kotropoulos, 2013; Panagakis & Kotropoulos, 2012a, 2012b). However, Hanilci and Kinnunen (2014) additionally considered equal error rate (EER) (the rate at which the false alarm rate and miss rates are equal) and detection trade-off (DET) curves (Martin et al., 1997) (graphically presents the error rate by using the trade-off between false acceptance rate and false rejection rate) for performance evaluation.

2.4.3 Communication device identification based on call recording

The majority of works in the field of audio source device identification have focused on identifying the recording device. However, this study establishes a new direction known as communication device identification based on recorded VoIP calls received from mobile devices. This approach assumes that call recording signals contain intrinsic artifacts of both transmitting- and receiving-end devices. The study suggests that the

transmitting device artifacts could be delivered through calls that traverse cellular, PSTN or VoIP networks. The proposed method was implemented for source computer device identification from recorded VoIP calls (Jahanirad et al., 2014). Alternatively, inspired by previous RF device identification approaches using the transmitted wireless signal by Suski II et al. (2008), Hasse et al. (2013) implemented the first practical identification system of the global system for mobile (GSM) communication devices based on the physical characteristics of RF hardware. These approaches differ from the current study because the radio transmissions are captured passively by third party receivers located within the communication range of the sending device and are independent of the cooperation of the sender. Therefore, Hasse et al. (2013) developed a two-way receiving software defined radio (SDR) for full extraction of bursts allocated to individual mobile devices. The authors placed SDR in the receiving range of both communication entities. As such, the complete communication stream can be observed in both directions without interfering with or disrupting the ongoing GSM communication.

2.4.3.1 Challenges of source communication device identification

Section 2.4.2.1 discussed a number of issues for audio source device identification based on microphone recording. However, for call recording signals, additional issues arise from controlling the influences of both transmitting and receiving ends as follows:

(a) *Transmitting/receiving side disturbances*: environmental disturbances are induced during the call from both the receiving and transmitting sides, and the echo is caused by the coupling between microphones and speakers. The classifier performance might be reduced when the training and testing data instances are generated with different disturbances.

(b) *Channel distortions and channel noise*: signals are shaped, delayed and distorted by the frequency response and the attenuating (fading) characteristics of the channel

(Vaseghi, 2008); hence, a signal is distorted in transmission through a channel by a convolutional process. Furthermore, most channels are noisy, whereby the influence of the channel noise to signal is in the additive form.

(c) *Communication network*: the communication network (i.e. VoIP, PSTN, cellular) influences the call recording signal. The codec transformations applied to multiple intermediary PSTNs, VoIP and cellular networks, in combination with packet loss and noise characteristics, left their artifacts on call recording signals corresponding to different communication networks. The classifier performance might be reduced when the training and testing data instances are created from call recordings for different communication networks.

(d) *Multi-source audio signals*: call recording comprises a two-sided conversation, for example, the speech recording conversation is contextualized by two different speech signals and two speakers' characteristics.

(e) *Effects of the second microphone from recording side*: the call recording model scenario records the mixture of two signals, including the audio signal from the microphone on the transmitting side and the audio signal from the microphone on the recording side. The influences of the second microphone may introduce confusion for source communication device identification.

2.4.3.2 Communication device identification

Table 2.12 presents a comparative overview of the communication device identification frameworks based on resolution strategies, which will be discussed in this section.

(a) ***Domain understanding and data preparation***

For computer device identification, Jahanirad et al. (2014) engaged the VoIP communication between computer devices and the single stationary device. The setup

recorded conversations between two different speakers (one male and one female) on either side using Pamela (2013). The setup consists five iMacs of the identical model located in the Multimedia Research Lab and five desktop PCs of the identical model located in the Micro Lab.

Table 2.12: Challenges and Strategies for Communication Device Identification

Method	Challenges	Resolution strategies
(Hasse et al., 2013)	Mobile phone (RF hardware) specific features	Introduced time-based patterns of the modulation errors as a unique device-dependent feature
	Channel distortions and wireless communications	Removed random effects of the wireless communication channels due to different locations or variable radio parameters
	Recording the radio signals	Utilized a two-way receiving SDR to capture GSM signals from the mobile phone (Uplink) and from the base station (Downlink) at the same time without interfering with the ongoing GSM communication
	Environmental influences	Placed the mobile phones next to the receiver during the training stage. For the test stage, placed the phones four meters apart from the receiver at different locations
	Intra-model similarities	Identified correctly a total of thirteen mobile phones including four identical and nine almost identical mobile phones
(Jahanirad et al., 2014)	Speech signal influences	Recorded speech conversation during Skype VoIP call and applied near-silent detection to select the non-speech segments for extracting the entropy-MFCC features Mixing the speech conversations between two different speakers have shown little contamination effect on classification accuracy among all computer devices
	Environmental influences	The PCs and iMacs were located in two different rooms; mixing the speech samples recorded in different environments has shown little contamination effect on classification accuracy among all computer devices
	Intra-model similarity	Perfectly identified among five identical iMacs and five identical PCs

Alternatively, for signal acquisition, Hasse et al. (2013) used two universal software radio peripheral N210 (USRP N210) devices operating in synchronized multi-input and multi-output mode. The acquisition site was the interior of an office building, without any arrangements such as shielding. The site was exposed to other common radio signals

present in office buildings, such as wireless networking, at the time of acquisition. The method selected a base station operated by T-Mobile, because of the strong signal and a recordable combination of radio channels used for frequency hopping. The method recorded two analog radio sources on four channels in total (the main and the hopping channels per GSM signal) with a sample rate of 5MHz each. For the test setup, the method used a total of 13 mobile phones of 4 different manufacturers and 9 models. During the training stage, the mobile phones were positioned beside the receiver, whereas during the test stage they were placed in a different location, four meters away from the receiver. This test setup assures that the performance evaluated for the identification algorithm is location independent. Moreover, prior to feature extraction, the raw captured signal was preprocessed to demodulate the GSM signal. The method split the recorded signal into individual GSM radio channels using a polyphase filterbank channelizer.

(b) ***Extracting communication device specific features***

The computer device identification approach developed by Jahanirad et al. (2014) computed the entropy-MFCC features from both the original speech signal and near-silent segments. This is to justify the robustness of the proposed method against the characteristics of speech signals. For GSM device identification, Hasse et al. (2013) investigated RF hardware characteristics that remain stable over different dimensions and especially over time. This study identified and removed all aspects that introduce random behavior or, which can contaminate the features' discriminative ability. Thus, the approach interprets the signal according to the Gaussian minimum shift keying modulation. For every extracted burst, the receiver demodulated the signal to produce the binary representation employed to create a mathematical ideal simulation of the modulated burst. The differences between every observed and ideal sample are used to estimate the error metrics. The common error metrics describing the precision of a modulated signal include magnitude error (ME) (the difference in amplitude), phase error

(PE) (the phase difference), error vector and error vector magnitude (the vector between the observed and ideal sample and its length). Based on these metrics the proposed feature extraction algorithm computed the accumulated ME trajectory a_{ME}, accumulated PE trajectory a_{PE} and accumulated EVM trajectory a_{EVM}. However, the a_{PE} and a_{EVM} trajectory show a dependency on a linear model. The slope of this model is a remaining frequency error, attributed to imperfect synchronization mechanisms in mobile phones and receivers. Hence, the method determined the linear frequency model with a least-squares approximation and deducted the frequency-related part from PE and EVM in respect to time, resulting in the frequency-corrected PE, $a_{PEfc,}$ and frequency-corrected EVM a_{EVMfc}. The a_{PEfc}, $a_{EVMfc,}$ and a_{ME} are the error patterns used as input for a classification algorithm. In addition, the method determined the accumulated power trajectory a_{PWR} for each sample position t of the normal burst using normalized in-phase signals and quadrature signals. The power trajectory a_{PWR} acts similarly to the a_{ME} but represents the general sending power of an RF signal instead of magnitude errors in the modulation domain.

(c) ***Machine learning approaches for feature analysis***

Jahanirad et al. (2014) evaluated and compared the supervised learning techniques, including Naïve Bayesian, linear logistic regression, NN, SVM, and SMO classifier, using the data mining tool, Weka. Using a different approach, Hasse et al. (2013), assigned the GSM signatures into a linear SVM for classification.

(d) ***Performance metrics for decision-making***

Jahanirad et al. (2014) employed performance metrics, including truly classified blocks, false classified blocks, root mean squared error (RMSE), classification accuracy and elapsed time (the complete run-time of the experiment). For GSM device identification, Hasse et al. (2013) used the true acceptance rate (probability of detecting

the given device correctly) the performance metrics. The average TAR among every device under test indicates the overall success rate or classification accuracy of an experiment.

2.4.4 Discussion and emerging trends

The state-of-the-art reviewed papers play a significant role in audio authenticity and interpretation. As a result of the lack of systematic research in this area, many problems were addressed for the first time, and some have been left unresolved. At the beginning, Section 2.4 drew upon the evolutionary body of existing research for audio source device identification approaches using the phylogenetic tree. The tree showed that until recently this research field focused on source recording device identification, including: (a) microphone identification and (b) acquisition device identification. However, source communication device identification has been introduced during this study as a new direction for audio authenticity and interpretation device-based techniques. The communication devices could be classified as mobile devices, computer devices or any other type of device used for GSM, PSTN or VoIP communication. Moreover, this study discusses the audio mining stages (data preparation, feature extraction, feature analysis and decision making) for each specific study in an attempt to provide the possible ground truth for comparison. However, the quantitative comparison of the performance of different techniques is technically impossible because these techniques utilized different test sets and lack the established benchmark specifically collected for audio source device identification. Meanwhile, the study evaluates the challenges and resolution strategies for the existing research in each field, as listed in Table 2.7, Table 2.11 and Table 2.12. The progression of each study is considered in terms of the optimality of resolution strategies applied to address challenges in each category. Consequently, in terms of presenting the evolutionary body of the research in Section 2.4.1, each approach's progression and history are mapped using its reference author with the phylogenetic tree in Figure 2.8.

The discussion in this section is an attempt to shed light on some of the emerging trends and frameworks in the field of audio source device identification.

Table 2.13: Comparison Based on Data Preparation and Feature Extraction

Ref	Audio acquisition devices		Recording Signal	Feature set
	Type	No.		
Kraetzer et al. (2007)	Microphone	4	speech/non-speech	time domain (7) and Mel-cepstrum domain (56)
Buchholz et al. (2009)	Microphone	7	Non-speech	Fourier Coefficients
Kraetzer et al. (2009)	Microphone	7	speech/non-speech	time domain (7) Mel-cepstrum domain (56), and frequency domain (35)
Kraetzer et al. (2011)	Microphone	4	speech/non-speech	time domain (7) Mel-cepstrum domain (56), and frequency domain (529)
Kraetzer et al. (2012)	Microphone	6	speech	time domain (9) Mel-cepstrum domain (52), and frequency domain (529)
Vu et al. (2012)	Microphone	5	speech/non-speech	MFCCs (13)
Malik and Miller (2012)	Microphone	8	non-speech	Bicoherence magnitude and phase spectrum
Eskidere (2014b)	Microphone	16	speech	LPCCs (13)
D. Garcia-Romero and Epsy-Wilson (2010)	Microphone	8	speech	MFCCs (23)
	Telephone Handset	8	speech	
Panagakis and Kotropoulos (2012a)	Telephone Handset	8	speech	RSF (325)
Panagakis and Kotropoulos (2012b)	Telephone Handset	8	speech	LSF (8)
Kotropoulos (2013)	Telephone Handset	8	speech	SSF (700)
Hanilçi et al. (2012)	Cell Phone	14	speech	MFCC (24)
Hanilci and Ertas (2013)	Cell Phone	14	speech	MFCC, BFCC and LFCC
Hanilci and Kinnunen (2014)	Cell Phone	14	speech/non-speech	MFCC and LFCC (24)
Eskidere (2014a)	Acquisition device	14	speech	DWBCs and WPBCs
Hasse et al. (2013)	Mobile Phone	13	Normal burst	accumulated errors: a_{PEfc}, a_{EVMfc}, a_{PWR} and a_{ME}
Jahanirad et al. (2014)	Computer Device	10	speech/non-speech	entropy-MFCCs (12)

Table 2.14: Comparison Based on Feature Analysis and Decision Makings

Ref	Classifier	Optimization Solution	Test Option	Identification Approach	Accuracy
Kraetzer et al. (2007)	Naïve Bayesian	Verifier-Tuple	10-fold cross validation	Inter-device	75.99%
Buchholz et al. (2009)	SL	Near-silent Detection PCA	10-fold cross validation	Inter-device	93.5%
Kraetzer et al. (2009)	SL and J48	Rank level fusion	Impartially supplied training and test sets	Inter-device	100%
Kraetzer et al. (2011)	Benchmarking	Feature Selection	10-fold cross validation	Inter/Intra-device	80<=ACC < 82.5%
Kraetzer et al. (2012)	Rotation Forests	Feature Selection	10-fold cross validation	Inter/Intra-device	99.85%
Vu et al. (2012)	1-ISVM	RICF	Impartially supplied training and test sets	Inter/Intra-device	Recall =0.85 Precision =0.00
Malik and Miller (2012)	Threshold-based multiple hypotheses testing	Scale invariant Hu moments	Magnitude similarity and inter-class phase variability	Inter/Intra-device	100%
Eskidere (2014b)	GMM	GMM	Impartially supplied training and test sets	Inter-device	99.58%
D. Garcia-Romero and Epsy-Wilson (2010)	GSV-SVM	GMM-UBM	2-fold cross validation	Inter-device	99%
			2-fold cross validation	Inter-device	93.2%
Panagakis and Kotropoulos (2012a)	SRC	Feature Selection	2-fold cross validation	Inter-device	95.55%
Panagakis and Kotropoulos (2012b)	SVM	Feature Selection	2-fold cross validation	Inter-device	97.58%
Kotropoulos (2013)	SRC	Feature Selection	2-fold cross validation	Inter-device	94.99%
Hanilçi et al. (2012)	SVM	GLDS	Impartially supplied training and test sets	Inter/Intra-device	96.42%
Hanilci and Ertas (2013)	GLDS-SVM	CMN, CVN and CMVN normalization	Impartially supplied training and test sets	Inter/Intra-device	98.15%
Hanilci and Kinnunen (2014)	GLDS-SVM GMM-ML GMM-MMI	CMVN normalization, speech activity detector, and noise enhancement	Impartially supplied training and test sets	Inter/Intra-device	98.39%
Eskidere (2014a)	SVM	Normalization with SMs	Impartially supplied training and test sets	Inter/Intra-device	93.81%
Hasse et al. (2013)	SVM	Synchronization and Frequency correction	Impartially supplied training and test sets	Inter/Intra-device	97.62%
Jahanirad et al. (2014)	Naïve Bayesian	Near-silent Detection	10-fold cross validation	Inter/Intra-device	100%

Table 2.15: Summary of the Contribution and Limitations of Respective Studies

Ref	Contribution	Limitations
Kraetzer et al. (2007)	Introduces the first practical evaluation on microphone classification	Insignificant result for generalization because the training and test set have originated from the same very small set of audio signals
Buchholz et al. (2009)	Extracts the microphone specific features from non-speech segments	Large set of features from the particular domain
Kraetzer et al. (2009)	Uses different information fusion techniques for optimization	Lacks systematic evaluation on microphone specific features
Kraetzer et al. (2011)	(1) Provides a context model for microphone forensics, (2) uses microphone classification for composition detection	Small microphone set
Kraetzer et al. (2012)	Identifies the initial and second microphones based on playback recordings	Lacks standard evaluation for the playback detection
Vu et al. (2012)	(1) Uses OCC approach instead of supervised learning, (2) develops representative instance of OCC approach for noisy recordings	Lacks evaluation of the effects of different noise levels (SNR) on classification performance
Malik and Miller (2012)	Develops noise-robust microphone specific feature because HOSA reveals not only the amplitude information but also the phase information about the underlying process	Not applicable to speech data
Eskidere (2014b)	(1) Proves higher performance of LPCC features over MFCCs for microphone identification, (2) implements the GMM-based modeling	(1) Limited search space, (2) lacks systematic feature selection approach, (3) lacks proper justification regarding the selected features
D. Garcia-Romero and Espy-Wilson (2009), D. Garcia-Romero and Espy-Wilson (2010), D. Garcia-Romero and Epsy-Wilson (2010)	Captures all discriminant information from the acquisition device using the GMM-UBM architecture	Lacks clarification regarding environmental factors and control conditions
D. Garcia-Romero and Epsy-Wilson (2010)	Analyzes the effects that room acoustics and intra-device variability have in respect to classification accuracy	Lack of evaluation based on noise levels of each environment
Panagakis and Kotropoulos (2012a)	Uses sparse representation of RSF as an intrinsic fingerprint for device identification	Lacks evaluation of the effects of environmental disturbances
Panagakis and Kotropoulos (2012b)	Uses LFS as an intrinsic fingerprint for device identification	LSF reduces the performance of SRC
Kotropoulos (2013)	Uses SSF as an intrinsic fingerprint for device identification	The experiments are limited to the LLHDB database
Hanilçi et al. (2012)	(1) provides both fundamental and experimental evidence for the feasibility of the MFCC features, (2) uses GLDS for normalizing MFCCs	Limited forensic application as a result of the use of the Cell Phone as an ordinary tape recorder

Table 2.15, continued: Summary of the Contribution and Limitations of Respective Studies

Ref	Contribution	Limitations
Hanilci and Ertas (2013)	Compares and optimizes the acoustic features for source cell phone identification	Lack of proper justification of their findings with larger device datasets, under mismatched additive noise conditions
Hanilci and Kinnunen (2014)	Analyzes the amount of device-specific information in speech and non-speech parts	Considerable reduction of identification accuracy under additive noise
Eskidere (2014a)	Investigated the applicability of DWT over STFT for extracting acquisition device-specific features	Computational complexity of the DWBCs and WPBCs over MFCCs
Hasse et al. (2013)	The first practical identification system of GSM devices based on RF fingerprints	Insignificant result for generalization as a result of small device test set
Jahanirad et al. (2014)	Identifies the source communication computer device from recorded VoIP call	Lacks evaluation of explicitly achieved training and test samples with uncontrolled test set-up

2.4.4.1 Current state of audio source device identification

Despite the inability for quantitative comparison, Table 2.13 and Table 2.14 enable the identification of the present status of audio source device identification approaches in audio forensics, the discovery of their contribution toward optimization and the proposal of new directions for the future. Table 2.13 presents the reviewed techniques regarding data preparation and feature extraction elements. Table 2.14 presents the reviewed techniques in regard to feature analysis and decision-making elements. Meanwhile, the contribution and limitations of the respective studies are summarized in Table 2.15.

Table 2.16: Challenges in Audio Source Device Identification Approaches

Method	Challenges											
	Environment disturbances	Audio manipulation	Forensic Features	Computation cost	Inter/Intra similarity	Open set	Classifier benchmarking	Setup & aging	Audio content	Playback recording	Audio settings	Dataset Dependency
Kraetzer et al. (2007)	Yes	No	No	No	No	No	Yes	No	Yes	No	No	No
Buchholz et al. (2009)	Yes	No	Yes	Yes	Yes	No	No	No	Yes	Yes	No	Yes
Kraetzer et al. (2009)	Yes	No	No	No	No	No	Yes	No	Yes	No	No	Yes
Kraetzer et al. (2011)	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes
Kraetzer et al. (2012)	Yes	Yes	Yes	Yes	No	No	Yes	No	No	No	No	Yes
Vu et al. (2012)	Yes	No	No	No	No	Yes	Yes	Yes	Yes	No	Yes	Yes
Malik and Miller (2012)	Yes	Yes	Yes	No	Yes	No	No	Yes	No	Yes	No	No
Eskidere (2014b)	Yes	No	Yes	No	Yes	Yes	No	Yes	Yes	No	No	No
D. Garcia-Romero and Espy-Wilson (2009), D. Garcia-Romero and Espy-Wilson (2010), D. Garcia-Romero and Epsy-Wilson (2010)	Yes	No	Yes	No	No	No	No	Yes	Yes	Yes	No	No
Panagakis and Kotropoulos (2012a), Panagakis and Kotropoulos (2012b), Kotropoulos (2013)	No	No	Yes	No	No	Yes	No	Yes	Yes	No	No	Yes
Hanilçi et al. (2012), Hanilci and Ertas (2013)	Yes	Yes	Yes	No	No	Yes	No	Yes	Yes	Yes	No	Yes
Eskidere (2014a)	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	No	Yes
Hasse et al. (2013)	Yes	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	No	No
Jahanirad et al. (2014)	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	No	Yes

Table 2.17: Challenges in Communication Device Identification Approaches

Method	Challenges				
	Transmitting/Receiving side disturbances	Channel distortion/noise	Communication network	Multi-source audio signal	Recording device influences
(Hasse et al., 2013)	Yes	Yes	No	No	Yes
(Jahanirad et al., 2014)	Yes	No	No	Yes	Yes

2.4.4.2 Emerging trends of audio source device identification

The overall comparison of the challenges addressed, in regard to the field of audio source device identification are presented in Table 2.16 and Table 2.17. However, there are open issues and challenges still requiring more research in the future, summarized in this section as follows: (a) generalizing models to larger datasets and real-time scenarios, (b) measuring the features' vulnerability against convolutional influences such as speech contents, speakers, environmental disturbances, and communication channel distortion and noise, (c) addressing noise-robust feature extraction methods, (d) source playback device identification, (e) audio source device identification with applications for tampering detection, (f) source communication device identification from recorded calls based on different types of communications, (g) developing source separation techniques to separate the transmitting signal from the receiving signal, and (h) tracking or authenticating GSM-based devices based on the physical characteristics of the RF hardware.

2.5 Summary

This chapter has presented selected fundamentals on audio signals and audio mining techniques, in addition to an extensive review of audio source device identification. Meanwhile, for categorizing and analyzing existing approaches, four important stages in developing source device identification schemes were defined; that is, data selection and preprocessing, the extraction of device specific features, the implementation of accurate,

robust and computationally efficient machine learning techniques and performance evaluation. Various audio recording scenarios have been used to exploit characteristics of acquisition devices, such as microphone recording and call recording. Implementation of a feature extraction algorithm usually takes the form of modification of existing standard speech and speaker recognition features while the main characteristic is maintained, where possible. Evaluation and validation of an algorithm are an important step in demonstrating the effectiveness of the algorithm.

Subjective and/or objective quality assessment is usually the primary concern, and sometimes computational complexity is also examined, especially in the case of real-time applicability. Open issues for future research were then discussed. Dealing with newly-arising audio source communication device identification approaches, such as the identification of GSM-based mobile phones and VoIP-based mobile devices, is necessary in order to provide efficient solutions for future tracking or authenticating the call recording. Coping with various factors that affect the performance of audio source device identification (for example, its dependence on the environment, microphone setup and aging, and audio manipulations) still remains a challenge. Finally, a more profound understanding of call recording mechanisms will be desirable through multi-disciplinary research. Such understanding will need to be then incorporated effectively into audio source communication device identification approaches. The approaches provide efficient solutions for law enforcement agencies dealing with continuously increasing volumes of crimes committed through calls.

CHAPTER 3: ADOPTED SPECTRAL ANALYSIS TECHNIQUES

This chapter describes the concepts and justifies the spectral analysis techniques adopted to optimize acoustic features for audio source mobile device identification. The linear and nonlinear systems were investigated to model the mobile device frequency response on call recording signal. The study of the control system model of the mobile device transmission system incorporates the principles of methods, rules, and assumptions necessary for audio source mobile device identification, whereby the concepts are derived from these principles. The theoretical concerns pointed in this chapter are used as the justification basis for the implementation and evaluation of the proposed framework in Chapter 4 and 5, respectively. Figure 3.1 illustrates the relationship between this chapter and the remaining chapters of the thesis. Meanwhile, Chapter 2 and 7 are missing in this diagram because the contents of these chapters are related to all parts of the thesis. Thus, it is possible to summarize the importance of this chapter within the thesis context:

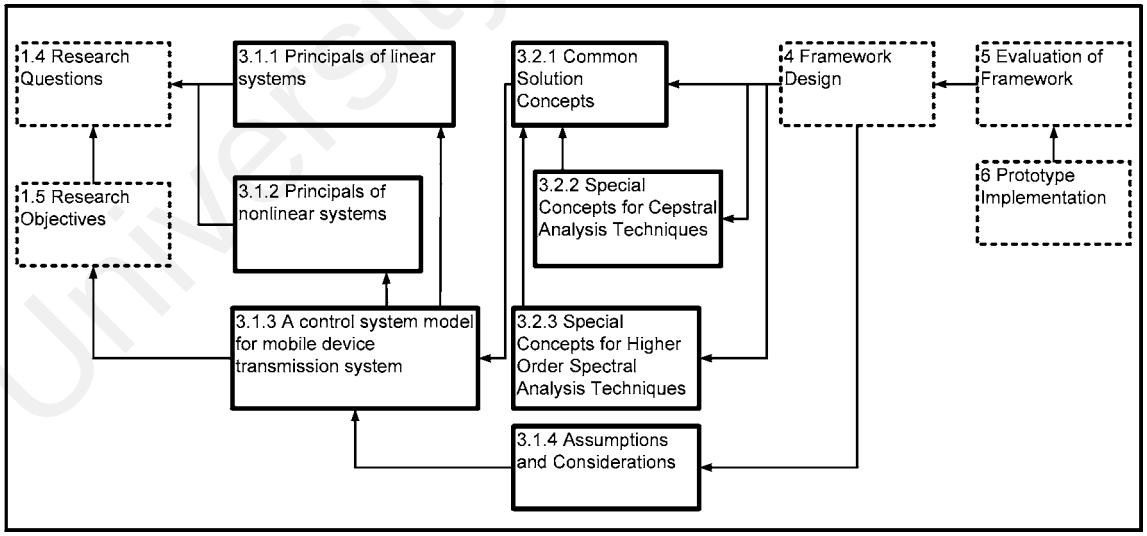


Figure 3.1: Organization of Chapter 3 Compare to Thesis Contents

- First, with respect to the research questions described in Section 1.4, principles of methods, rules, and assumptions utilized by linear and nonlinear systems are analyzed in Section 3.1.1 and 3.1.2. A control system model for mobile device transmission

systems is studied in Section 3.1.3 giving rise to the research objectives in Section 1.5. Subsequently, the model allows defining assumptions and considerations in Section 3.1.4 for the purpose of designing and developing the framework.

- Second, common spectral analysis techniques are adopted to capture signal variations due to mobile device frequency response on random stationary call recording signals. The special concepts are developed to identify its response function corresponding to different linear and nonlinear subsystems in the mobile device transmission system. As a result, within this study, the concepts are extended in Section 3.2.2 and 3.2.3 to justify the need of cepstral analysis and higher-order spectral analysis techniques to optimize acoustic features for audio source mobile device identification.
- Third, the concepts are utilized for design and evaluation of the framework and implementation of the prototype in the remaining chapters.

3.1 Mobile Device Transmission System

This section aims to investigate the use of linear and non-linear models for determining the mobile device frequency response on the call recording signal. Moreover, the methodology behind the investigations that are made in this study is considered to address the research problems formulated for a mobile device identification.

3.1.1 Principles of Linear Systems

Systems are described by their response to a particular input, which provides both a specification of the system and the basis of how the response to more complex inputs may be built up. This section describes the principles of linear systems and investigates whether the general nonlinear call recording signal could be the output of the linear system whose input is the stationary random process. A system is defined as linear if it satisfies the superposition principle. Let $x_1(n)$ and $x_2(n)$ be input sequences, then the system is denoted as linear if and only if

$$T\{a_1x_1(n)+a_2x_2(n)\}=a_1T[x_1(n)]+a_2T[x_2(n)] \quad \textbf{(3.1)}$$

, where a_1 and a_2 are random constants, and T is the discrete time (Chitode, 2008).

Moreover, the impulse response function represented as $h(t, t_1)$ which is the response at time t due to the application of a unit impulse at time t_1, as shown in Figure 3.2. Furthermore, there are additional properties that are ideal for linear systems as follows.

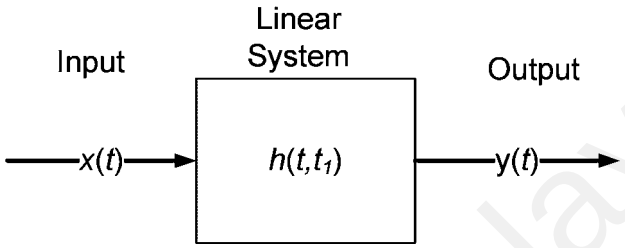


Figure 3.2: Linear System Representation

(a) The system is represented as time invariant if $h(t, t_1) = h(t - t_1)$, whereby it considers the delay time between t and t_1.

(b) The system is denoted as casual if $h(t, t_1) = 0$ for $t_1 > t$. This means the system only responds after the impact is applied.

(c) The system is stable if its impulse response is absolutely summable

$$\sum_{t=-\infty}^{\infty}|h(t)|<\infty. \quad \textbf{(3.2)}$$

These properties are combined in terms of a linear, time-invariant (LTI), causal, stable system to achieve a relationship with respect to the system's input, $x(t)$, to its output, $y(t)$, which is given by (Hammond & White, 2008):

$$y(t)=\int_{-\infty}^{t}h(t-\tau)x(\tau)d\tau=\int_{0}^{\infty}x(t-\tau)h(\tau)d\tau=h(t)*x(t) \quad \textbf{(3.3)}$$

In Eq. 3.3 the operator $*$ is denoted as convolution that is defined using the integral relationships. This relationship can be represented in the frequency domain as

$$Y(\omega)=F_1[h(t)*x(t)]=H(\omega)X(\omega) \quad \textbf{(3.4)}$$

Where $H(\omega)$ is represented as linear system frequency response and $F_1[.]$ is denoted as the one-dimensional Fourier transform.

3.1.2 Principles of Nonlinear Systems

This section describes principles of the nonlinear system and investigates whether the call recording signal could be the output of the nonlinear system whose input is the stationary random process. The system is nonlinear if there is no relationship between its input and output. However, the subsystems in nonlinear systems can be modeled through extensions of the linear convolution relationship using Volterra expansions. The discrete p^{th}-order Volterra system is denoted by p+1 terms of the Volterra series. Let $y(t)$ be the response of the discrete time-invariant p^{th}-order Volterra filter for input signal of $x(t)$ which is given by (Hammond & White, 2008):

$$\begin{aligned} y(t) &= h_0 + \sum_{i=1}^{p} H_i[x(t)] \\ &= h_0 + \sum_i \sum_{(\tau_1,\tau_2,\ldots,\tau_3)} h_i(\tau_1,\tau_2,\ldots,\tau_3)x(t-\tau_1)\cdots x(t-\tau_i) \end{aligned} \quad \textbf{(3.5)}$$

Equation 3.5 is determined using the Tylor series expansion, where $H_i[.]$ represents the ith-order Volterra operator, $h_i(\tau_1, \tau_2,\ldots,\tau_i)$ are the Volterra kernels of the system that are limited and discrete at each τ_i and are symmetric functions of their arguments. If the system is casual $h_i(\tau_1, \tau_2,\ldots,\tau_i)=0$ for any $\tau_i < 0$. Let $x(t)$ be the pure random process with zero mean, then the operator $H_1[.]$ will represent a general linear model, whereas for $H_i[.] > 1$, the operator such as $H_2[.]$ refers to quadratic model and the operator $H_3[.]$ denotes the cubic model. As a result, the time series signal y(t) is denoted by the second-order Volterra model which is given by (Nikias & Petropulu, 1993),

$$y(t) = \sum_{\tau_1} h_1(\tau_1)x(t-\tau_1) + \sum_{\tau_1}\sum_{\tau_2} h_2(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2) \quad \textbf{(3.6)}$$

In Eq. 3.6, the identification problem is to determine the impulse response $h_1(\tau)$ and the kernel $h_1(\tau_1, \tau_2)$. At this point, as shown in Figure 3.3 the second-order Volterra filter can

be modeled as a parallel connection of a linear system { $h_1(\tau)$ } and quadratic system { $h_1(\tau_1, \tau_2)$ }.

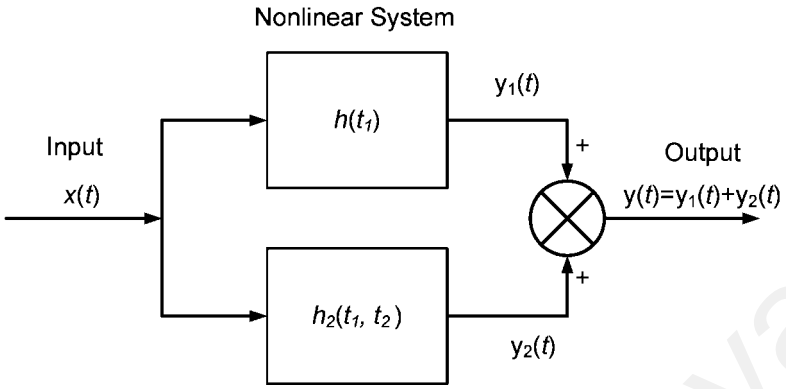


Figure 3.3: A Nonlinear System with Linear and Quadratic Subsystems.

3.1.3 A Control System Model for Mobile Device Transmission System

In general, a wireless communication signal processing pipeline consists of three basic components, including the transmitter that is responsible for signal conditioning to a proper form for conveying the signal through a channel, the channel that denotes the medium through which the signal propagates to the receiver and the receiver that regenerates the transmitted signal. The wireless communication system employs advanced signal processing methods in almost every aspect of the functioning of transmission and reception of information on any communication device. Meanwhile, a control system model describing the wireless communication signal processing is required to study different sources of signal variation in the audio signal during the call recording process. Figure 3.4 shows the control system block diagram based on the communication type and call recording that was performed within this study. It is evident that the model consists of different subsystems; each induces specific influence on the call recording signal. The audio signal can be modeled as either continuous or discrete signals in time or frequency domain. As a result, the study considered the frequency-domain representation of the signals for the control system modeling of the call recording

process. This is because this representation is more reliable for the modeling of the influence factors.

In the beginning, the model assumes that the audio delivered to the mobile device and the recording stationary contains speech. In the figure, the human speech production system is simplified as detailed in (Beigi, 2011b). Hence, $U(f)$ is the frequency transform of $u(t)$ in time-domain that is the input transmitting from human brain to drive the speech production system into producing a specific segment of speech. The transfer functions for the human nervous system and the neuro-muscular dynamics are combined and labeled as G_c, whereby the transfer function for vocal tract characteristics and dynamics is represented as G_v. Thus, G_c models the speech contents and G_v models the speakers' characteristics. The relationship for the speech production system, is given in (Eq. 3.7):

$$Y(f) = G_v(f)G_c(f)U(f). \quad \textbf{(3.7)}$$

As the speech signal is propagated through the air medium, it is usually distorted by various environmental factors prior to arriving at the microphone. According to (Pawera, 2003), there are three main sources of disturbances by environment: reflections, reverberation, and acoustic noise. The type of reflections differs with respect to the reflection period. The short-term reflections reach the microphone only fractionally later (0.8 to 20 ms) than the direct sound and generate discolouration. The medium-term reflections occur with delay times of usually more than 40 ms and enhance the volume of the direct sound. The long-term reflections occur with delay times longer than 80 ms and generate echoes. The reverberation takes place if multiple long-term reflections occur in the sound field, and the reflections reach to an energy intensity equal to one of the direct sounds. The influence due to all three environmental disturbances is determined in (Eq. 3.8):

$$Y_E(f) = e\int_f D(f)Y(f)df * F_{reverb}(f) + N_{envi}(f). \quad (3.8)$$

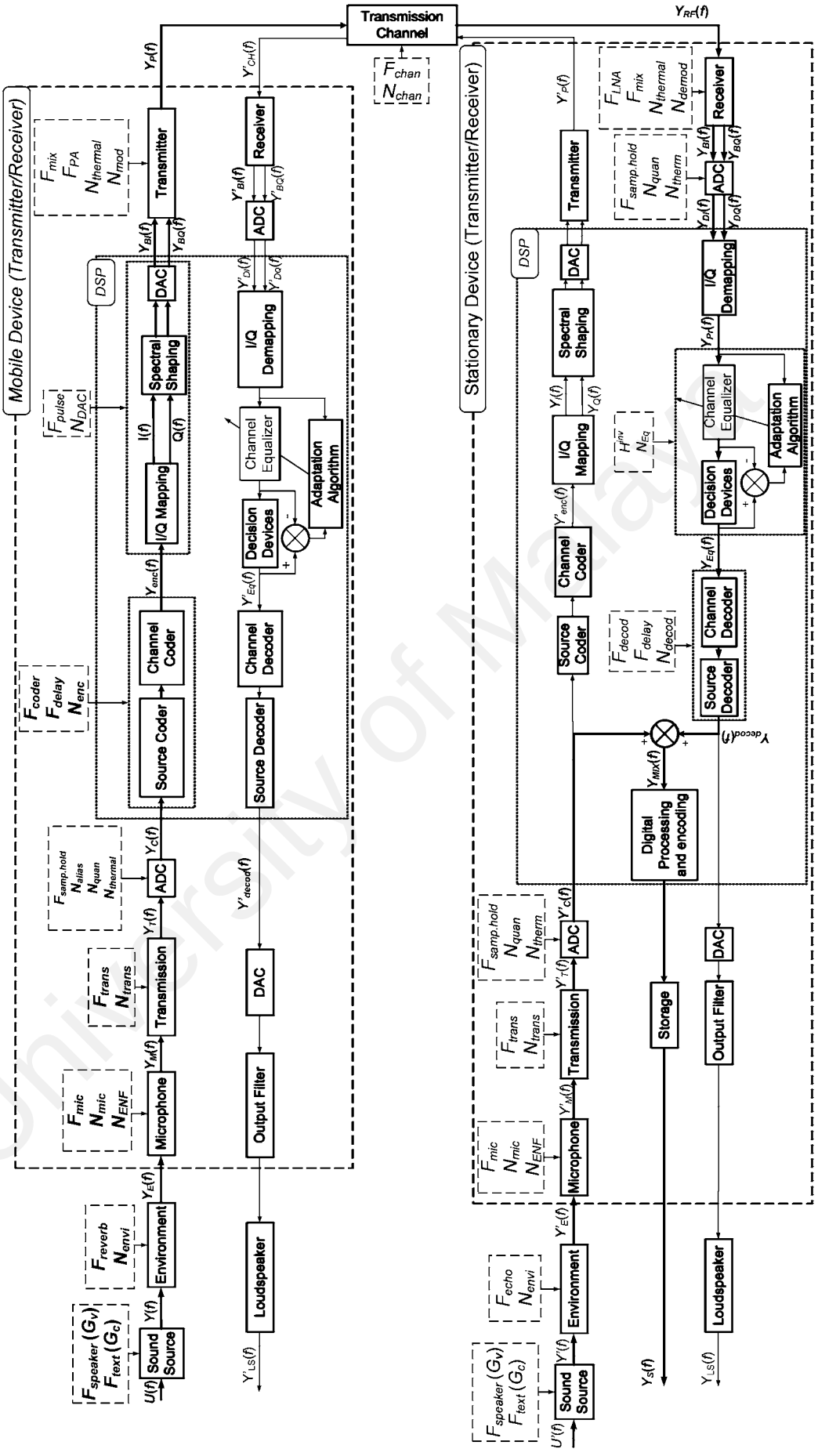


Figure 3.4: Mobile Device Transmission Process Pipeline-A Control System Model

In Eq. 3.8, $D(f)$ represents the discolouration function that produced from short-term reflections, e represents the enhancement factor introduced by medium-term reflections, and the convolution with $F_{reverb}(f)$ demonstrates the possible distortion from the echoes and / or reverberation (Takala & Hahn, 1992). Moreover, N_{envi} denotes the additive noise producing by the environment.

Afterward, the noisy, distorted signal $Y_E(f)$ passed over the microphone and modeled as:

$$Y_M(f) = \int_h F_{mic}(f) Y_E(f) df + N_{mic}(f) + N_{ENF}(f). \quad \textbf{(3.9)}$$

Where the function $F_{mic}(f)$ models the frequency response function of the microphone for frequency coefficients with index h; $N_{mic}(f)$ represents the thermal noise that the microphone generates, and $N_{ENF}(f)$ represents the ENF influence (ENF is the transmission frequency of power line, which has been utilized to authenticate time and location of the recording (Ojowu et al., 2012)). Kraetzer (Mai, 2013) determined the microphone frequency response with respect to the characteristics of the membrane in the microphone with its unique vibration behavior and interaction with the other parts of the microphone. They also considered other influences, including the orientation of the microphone to sound sources, the microphone mounting and possible aging phenomena of the microphone. Using a different approach, Malik and Miller (2012) modeled the microphone response with nonlinear systems using higher-order spectra. However, modeling the mobile device frequency response with respect to the microphone response may not be sufficient for audio source mobile device identification, especially for the case of call recording scenario.

The electronic signal passed over the microphone undergoes distortion and noise during transmitting to ADC, as given in (Eq. 3.10):

$$Y_T(f) = \int_h F_{trans}(f) Y_M(f) df + N_{trans}(f) \quad \textbf{(3.10)}$$

Where $F_{tran}(f)$ simulates the nonlinear distortion during the transmission of the signal from the microphone to recording device and $N_{tran}(f)$ represents the thermal noise generated by the transmission environment. As shown in Figure 3.5, the process of the ADC is modeled as,

$$Y_C(f) = \int_0^{f_{sample}} F_{samp.hold}(f) Y_T(f) df + N_{quan}(f) + N_{alias}(f) + N_{thermal}(f) \quad \textbf{(3.11)}$$

In Eq. 3.11, the upper limit f_{sample} represents the sampling frequency; $F_{samp.hold}(f)$ denotes the sampler response function; $N_{quan}(f)$, $N_{alias}(f)$ and $N_{thermal}(f)$ represents quantization, aliasing and thermal noise of the ADC.

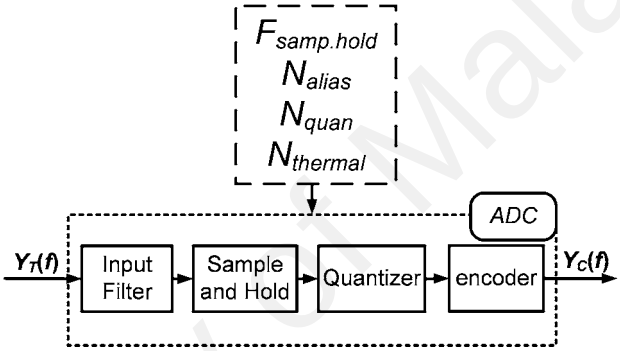


Figure 3.5: The ADC Process

A digital signal processor (DSP) block in the mobile device transmitter side performs a variety of encoding processes to the digital signal that converts a raw stream of data to another stream of data that is suitable for transmission, and that enhances the transmission process in a noisy channel. The encoding process includes source coding and channel coding, as given in (Eq. 3.12):

$$Y_{enc}(f) = e\int_0^N F_{coder}(f) Y_C(f) df * F_{delay} + N_{enc}(f). \quad \textbf{(3.12)}$$

In Eq. 3.12, $F_{coder}(f)$ denotes the speech coding algorithm for bit rate time of N; e represents the enhancement factor imposed by the channel coder; $N_{enc}(f)$ denotes the processing noise generated during the encoding, and the convolution with $F_{delay}(f)$ demonstrates the possible distortion from the delay caused by the speech coder.

The encoded signal is mapped to low frequency and low power signal known as the baseband signal $Y_B(f)$ that contains In-Phase $I(f)$ and Quadrature $Q(f)$ baseband data.

$$Y_{B_{I,Q}}(f) = \int_0^N F_{Pulse}(f)I(f)df + \int_0^N F_{Pulse}(f)Q(f)df + N_{DAC}. \quad \textbf{(3.13)}$$

Meanwhile, the pulse shaping is required before digital modulation to determine the bandwidth of the transmitted passband signal and also determine the amplitude variations of the signal envelope. Hence, $F_{pulse}(f)$ represents the pulse shaping frequency response and $N_{DAC}(f)$ represents the noise that is generated during the conversion of the digital baseband signal to its analog representation.

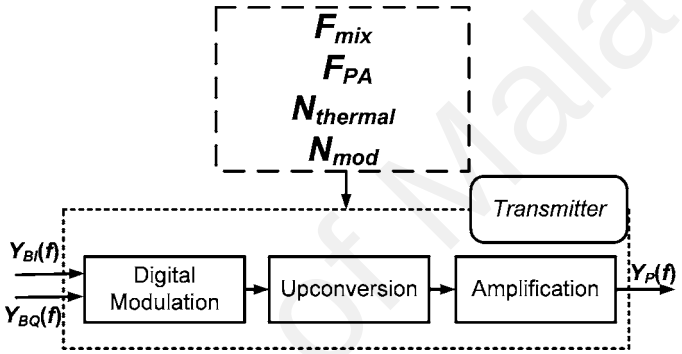


Figure 3.6: Basic Processes in Wireless Communication Device RF Transmitter

As demonstrated in Figure 3.4 and Figure 3.6, the baseband signal is the output of the DSP block and is converted to a passband signal that has a high frequency and a high power after the modulation process. Digital modulator is the important component in mobile device RF transmitter because the classification of the wireless communication devices varies with respect to the digital modulator architecture. Digital modulation techniques are categorized corresponding to the amplitude variability of the modulated signal into linear and nonlinear techniques. The linear technique modulates the amplitude or the phase of the signal, whereas the nonlinear technique modulates the frequency of the signal. For a selection of the modulation technique in wireless and mobile system applications a tradeoff has to be made between spectral efficiency, power efficiency, robustness against channel impairments and cost. A general model for a digitally modulated signal is given by

$$Y_P(f) = \left[\int_M F_{Mixer}(f) Y_{B_I}(f) df + \int_M F_{Mixer}(f) Y_{B_Q}(f) df \right] * F_{PA} + N_{Mod}(f) + N_{thermal}(f) \quad \textbf{(3.14)}$$

In Eq. 3.14 a mixer is utilized as a product device for frequency up-conversion process at the transmitter side and denotes by $F_{Mixer}(f)$, in addition, M is the number of levels, phases or frequencies of the signal. The outputs of the mixers are then added together to determine the modulated signal and then arrived in power Amplifier (PA) that is connected to an antenna through a matching circuit. Therefore, the convolution with $F_{PA}(f)$ demonstrates the possible amplitude and phase distortion from the PA. $N_{Mod}(f)$ represents the processing noise generated during the digital modulation, and $N_{thermal}(f)$ represents the thermal noise generated by the transmitter circuits.

PAs are located at the end of the transmitter pipeline to produce a signal with a power suitable for transmission through an antenna. In general, it is assumed that the received signal is corrupted by additive white Gaussian noise, and the signal distortion due to the channel response is modeled by intersymbol interference. This interference occurs in high-speed communications where the data symbols closely follow each other and time dispersion results in an overlap of successive symbols. However, in reality, there are several different sources of noise and interference that may restrict the performance of a communication system. The most common sources of distortion and noise in a mobile environment include receiver antenna thermal noise, interference from electromagnetic devices, radiation noise, background noise, echo and most importantly multi-path and fading (Vaseghi, 2008). The Eq. 3.15 summarizes the noise $N_{chan}(f)$ and distortions $F_{chan}(f)$ induced by the transmission channel,

$$Y_{RF}(f) = \int_h F_{chan}(f) Y_P(f) df + N_{chan}(f) \quad \textbf{(3.15)}$$

At this stage, the RF front end of the stationary device captures the transmitted RF signal. As can be seen in Figure 3.7, the RF front end performs amplification, filtering and down-conversion of the received RF signal to some intermediate frequency. Equation

3.16 describes the processes in the RF front end of the receiver, whereby the low-noise amplifier (LNA) is responsible for the amplification of the received signal, which usually has the very low power. The low frequency and low power RF signal, $Y_{RF}(f)$ and the Local-Oscillator (LO) signal, $Y_{LO}(f)$ are passed through the mixer to perform the down conversion. Therefore, the convolution with $F_{LNA}(f)$ demonstrates the possible amplitude and phase distortion from the LNA. Again, $N_{Pr}(f)$ represents the processing noise generated during the amplification and down-conversion process, and $N_{thermal}(f)$ represents the thermal noise generated by the RF front end receiver antenna.

$$Y_{IF}(f) = \left[\int_M F_{Mixer}(f) Y_{RF}(f) df + \int_M F_{Mixer}(f) Y_{LO}(f) df \right] * F_{LNA}(f) + N_{\Pr}(f) + N_{thermal}(f) \quad \textbf{(3.16)}$$

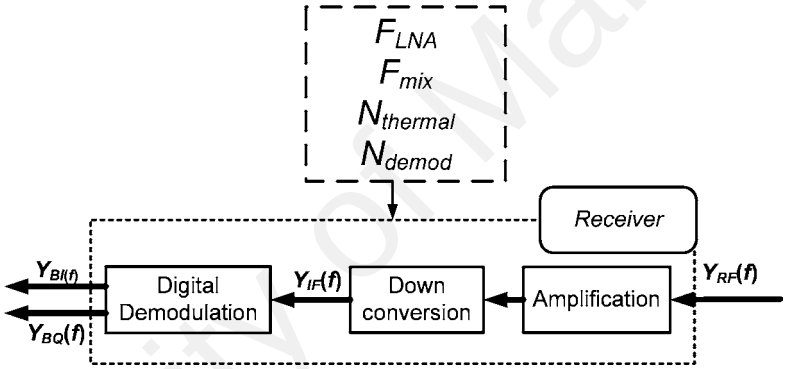


Figure 3.7: Basic Processes in Wireless Communication Device Receiver

In mobile communication at this stage, channel assignment strategy is required through three different approaches, namely frequency division multiple access (FDMA), time division multiple access (TDMA) and code division multiple access (CDMA), for frequency, time and code division multiple access. FDMA system employs different carrier frequencies to transmit the signal for each user. TDMA uses distinct time slots to transmit the signal to different users. CDMA utilizes different code to transmit the signal for each user. Hence, the IF filter is utilized for the selection of the desired frequency channel at a fixed frequency which is the IF. Let the demodulation block demodulates the generated IF signal to baseband signal with respect to selected frequency channel using another mixing process, as given by (Eq. 3.17):

$$Y_{BI,Q}(f) = \int_{0}^{f_{IF}-f_{LO}} F_{mix}(f)Y_{IF}(f)df + \int_{0}^{f_{IF}+f_{LO}} F_{mix}(f)Y_{LO}(f)df + N_{demo} \quad \textbf{(3.17)}$$

At the end of the demodulation, the ADC block converts the analog signal into its digital representation.

$$\begin{aligned} Y_{D_I}(f) &= \int_{0}^{f_{sample}} F_{sample}(f)Y_{B_I}(f)df + N_{quan}(f) + N_{thermal}(F), \\ Y_{D_Q}(f) &= \int_{0}^{f_{sample}} F_{sample}(f)Y_{B_Q}(f)df + N_{quan}(f) + N_{thermal}(F). \end{aligned} \quad \textbf{(3.18)}$$

The DSP block front end is responsible for digital processing of the digital baseband data denoted as $Y_{D_{I,Q}}(f)$ using I/Q demapping process as

$$Y_{\mathrm{Pr}}(f) = Y_{D_I}(f) + Y_{D_Q}(f) \quad \textbf{(3.19)}$$

The signal resulting from Eq. 3.19 is denoted as $Y_{Pr}(f)$ and equalized through blind deconvolution technique. This technique includes channel identification through adaption algorithm and channel equalization through decision levels. The general form of channel equalization is described as

$$Y_{Eq}(f) = \int_{h} H^{inv}(f)Y_{\mathrm{Pr}}(f)df + N_{Eq}(f) \quad \textbf{(3.20)}$$

Where $H^{inv}(f)$ is denoted as the estimated inverse channel filter, $N_{Eq}(f)$ is denoted as processing noise during equalization, $Y_{Eq}(f)$ is denoted as the equalized signal. Subsequently, Eq. 3.21 describes the decoding process at the end of the DSP block.

$$Y_{decod}(f) = e\int_{0}^{N} F_{decod}(f)Y_{Eq}(f)df * F_{delay} + N_{decod}(f). \quad \textbf{(3.21)}$$

For call recording, the input signal $Y'_C(f)$, that is sampled and quantized in the stationary device for transmission to mobile device and $Y_{decod}(f)$, that is the decoded signal received from the mobile device transmitter, are added as

$$Y_{Mix}(f) = Y'_C(f) + Y_{decod}(f). \quad \textbf{(3.22)}$$

, where $Y_{Mix}(f)$ is the mixture of the transmitting and receiving side signal. The call recording software applies additional signal processing and then compresses $Y_{Mix}(f)$ using the encoder to reduce its size; the call recording signal can be expressed as

$$Y_S(f) = \int_0^N F_{codec}(f) Y_{Mix}(f) df + N_{\Pr}(f) + N_{thermal}(f). \quad \textbf{(3.23)}$$

Where, $F_{codec}(f)$ is the frequency response of the speech codec utilized by the call recording application; $N_{Pr}(f)$ is the processing noise during the encoding; $N_{thermal}(f)$ is denoted as thermal noise generated by electric conductor during the storage process.

3.1.4 Assumption and Considerations within this thesis

The control system described at the beginning of this section allows understanding the architecture of the systems and the components in mobile device transmitter that are responsible for generating nonlinear distortion. On the other hand, it is also important to understand how these systems can be modeled to identify mobile device frequency response. To discriminate nonlinear distortions, resulting from transmitting mobile device linear and nonlinear subsystems, with other sources of distortion, the considerations within the proposed framework are restricted with certain rules and protocols:

- The framework suppressed the influences by speech signal from transmitter and receiver side by eliminating the speech segments of the call recording signals (Eq. 3.7).
- The framework eliminated the influences of the environment, by utilizing the near-silent segments of the call recording signals (Eq. 3.8).
- The framework controlled the transmitter and receiver influences of the stationary device as well as the call recording application using the same stationary device and recording software (Eq. 3.16-3.23).

- The framework controlled the influences by speech coder using the same VoIP application for VoIP communication with the stationary device (Eq. 3.12).
- The framework assumed that the GSM cellular communication always utilizes the same speech coder for communication (Eq. 3.12).
- The framework controlled the influences via different obstacles, reflectors, and diffraction in its propagation path using the same four locations for positioning the mobile devices with respect to the stationary device (Eq. 3.15).
- The framework controlled the network architecture influences using the same wireless local area network (WLAN) for all VoIP communications (Eq. 3.15).

Prior works in microphone forensics focused on nonlinear distortions generated from the microphone for extracting microphone intrinsic fingerprints. However, from a forensic point of view authenticating call recording audio based on its microphone fingerprints hardly provides an admissible solution (Kraetzer et al., 2012; Malik & Miller, 2012). This is because the mobile device user may have used the external microphone other than the mobile device built-in microphone. Hence, as summarized in Eq. 3.14 the considerable sources of nonlinear distortion in the mobile device transmitter are the mixer and power amplifier devices. Thus, the nonlinear distortion generated by these devices could be used for modeling the mobile device frequency response.

3.2 Concepts for Optimizing Acoustic Features

To determine the source of audio recording several techniques have been developed to estimate the acquisition device's fingerprint using acoustic features. The weaknesses and challenges with state-of-the-art audio source device identification approach, as it is presented in Section 2.4, are exploited as the basis to optimize acoustic features using spectral analysis techniques. This is important to design robust source mobile device identification framework based on recorded call.

3.2.1 Common Concepts for Spectral Analysis Techniques

One of frequently used spectral analysis techniques has been the estimation of the power spectral density or in basic terms the power spectrum of discrete time deterministic or stochastic processes. The existing power spectrum estimation techniques may be represented in a number of different classes such as conventional Fourier type methods (Nikias & Petropulu, 1993). Meanwhile, the ongoing process is handled as a superposition of statistically uncorrelated harmonic components. The distribution of power between these frequency components is then estimated. Thus, phase relations among frequency components are eliminated. The power spectrum contains information that principally exist in the autocorrelation sequence. Although the power spectrum technique is sufficient for spectral analysis of a Gaussian process of known mean, in more practical conditions, it is required to employ higher-order spectra to obtain information regarding deviations from Gaussians and presence of nonlinearities. Higher-order spectra or in another term polyspectra is defined using higher-order statistics, to handle nonlinear signal processing framework.

3.2.1.1 Cumulant Spectra of random stationary signals

Let $\{X(t)\}$, $t = 0,\pm1,\pm2,\pm3,\cdots$ be a real stationary random process and its moments up to order *n* is given by (Nikias & Petropulu, 1993):

$$m_n^x(\tau_1,\tau_2,\ldots,\tau_{n-1}) \stackrel{\Delta}{=} E\{X(t).X(t+\tau_1)\ldots X(t+\tau_{n-1})\}. \quad \textbf{(3.24)}$$

In Eq. 3.24 $E\{.\}$ represents the expectation operation and the moments are computed with respect to time differences $\tau_1,\tau_2,\ldots,\tau_{n-1}$, where $\tau_i = 0,\pm1,\pm2,\pm3,\cdots$ for all *i*. Using the same approach, the *n*th-order cumulants of $\{X(t)\}$ are (*n*-1)-dimensional functions that are given by (Nikias & Petropulu, 1993):

$$C_n^x(\tau_1,\tau_2,\ldots,\tau_{n-1}) \stackrel{\Delta}{=} Cum\{X(t), X(t+\tau_1),\ldots, X(t+\tau_{n-1})\}. \quad \textbf{(3.25)}$$

Let the cumulant sequence satisfies the condition

$$\sum_{\tau_1=-\infty}^{+\infty}\cdots\sum_{\tau_{n-1}=-\infty}^{+\infty}\left|C_n^x(\tau_1,\tau_2,\ldots,\tau_{n-1})\right|<\infty,$$
$$or \qquad \textbf{(3.26)}$$
$$\sum_{\tau_1=-\infty}^{+\infty}\cdots\sum_{\tau_{n-1}=-\infty}^{+\infty}\left(1+\left|\tau_j\right|\right)\left|C_n^x(\tau_1,\tau_2,\ldots,\tau_{n-1})\right|<\infty,$$

for j=1, 2, 3,…, n-1 the nth-order cumulant spectrum $C_n^x(\omega_1,\omega_2,\ldots,\omega_{n-1})$ of $\{X(t)\}$ exists, is continuous, and is defined as the (n - 1)-dimensional Fourier transform of the nth order cumulant sequence. Therefore, the nth-order cumulant spectrum is determined (Nikias & Petropulu, 1993):

$$C_n^x(\omega_1,\omega_2,\ldots,\omega_{n-1})\overset{\Delta}{=}\sum_{\tau_1=-\infty}^{+\infty}\cdots\sum_{\tau_{n-1}=-\infty}^{+\infty}C_n^x(\tau_1,\tau_2,\ldots,\tau_{n-1})\exp\left\{-j\left(\omega_1\tau_1+\omega_2\tau_2+\cdots+\omega_{n-1}\tau_{n-1}\right)\right\} \qquad \textbf{(3.27)}$$

for i=1, 2, 3,…, n-1and $\left|\omega_1,\omega_2,\ldots,\omega_{n-1}\right|\le\pi$. In overall, $C_n^x(\omega_1,\omega_2,\ldots,\omega_{n-1})$ is complex with magnitude and phase that is given by (Nikias & Petropulu, 1993):

$$C_n^x(\omega_1,\omega_2,\ldots,\omega_{n-1})=\left|C_n^x(\omega_1,\omega_2,\ldots,\omega_{n-1})\right|\exp\left\{j\Psi_n^x(\omega_1,\omega_2,\ldots,\omega_{n-1})\right\}. \qquad \textbf{(3.28)}$$

Based on the concepts and definitions discussed in this section, the power spectrum is estimated using the second-order cumulant spectrum that is given by (Nikias & Petropulu, 1993):

$$C_2^x(\omega)=\sum_{\tau=-\infty}^{+\infty}C_2^x(\tau)\exp\{-j(\omega\tau)\}. \qquad \textbf{(3.29)}$$

Let $\left|\omega\right|\le\pi$, $C_2^x(\tau)$ will be the covariance sequence of $\{X(t)\}$. Moreover, the bispectrum is estimated using the third-order cumulant spectrum. For $\left|\omega_1\right|\le\pi$, $\left|\omega_2\right|\le\pi$ and $\left|\omega_1+\omega_2\right|\le\pi$ the bispectrum is defined as (Nikias & Petropulu, 1993):

$$C_3^x(\omega_1,\omega_2,\ldots,\omega_{n-1})=\sum_{\tau_1=-\infty}^{+\infty}\sum_{\tau_2=-\infty}^{+\infty}C_3^x(\tau_1,\tau_2)\exp\left\{-j(\omega_1\tau_1,\omega_2\tau_2)\right\}. \qquad \textbf{(3.30)}$$

Meanwhile, the relationship between third-order cumulant sequences of $X(t)$ and third-order moment sequences of $X(t)$ is defined by (Nikias & Petropulu, 1993):

$$C_3^x(\tau_1,\tau_2)=m_3^x(\tau_1,\tau_2)-m_1^x\left[m_2^x(\tau_1)+m_2^x(\tau_2)+m_2^x(\tau_2-\tau_1)\right]+2(m_1^x)^3 \qquad \textbf{(3.31)}$$

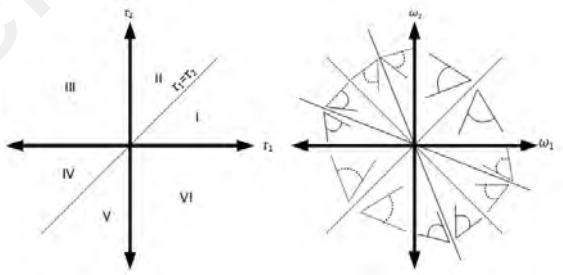
Hence, moments characteristics introduce important symmetry conditions as follows:

$$\begin{aligned} &C_3^x(\tau_1,\tau_2)=C_3^x(\tau_2,\tau_1)=C_3^x(-\tau_2,\tau_1-\tau_2)\\ &=C_3^x(\tau_2-\tau_1,-\tau_1)=C_3^x(\tau_2-\tau_1,-\tau_2)\\ &=C_3^x(-\tau_1,\tau_2-\tau_1). \end{aligned} \quad \textbf{(3.32)}$$

Accordingly, six symmetry regions (I-VI) are introduced, as shown in Figure 3.8 (a). Computing the third-order cumulants in any of the six regions allows determining the entire third-order cumulant sequence. Each region is characterized by its boundary in a way, for example, region I is an infinite wedge surrounded by the lines τ_1=0 and $\tau_1=\tau_2$; $\tau_1,\tau_2\geq 0$. Combining (Eq. 3.29) and (Eq. 3.31) give important symmetry conditions for bispectrum, which is defined by (Nikias & Petropulu, 1993):

$$\begin{aligned} &C_3^x(\omega_1,\omega_2)=C_3^x(\omega_2,\omega_1)\\ &=C_3^{x^*}(-\omega_2,-\omega_1)=C_3^x(-\omega_1-\omega_2,\omega_2)\\ &=C_3^x(\omega_1,-\omega_1-\omega_2)=C_3^x(-\omega_1-\omega_2,\omega_1)\\ &=C_3^x(\omega_2,-\omega_1-\omega_2). \end{aligned} \quad \textbf{(3.33)}$$

As a result, estimating the bispectrum in the triangular region $\omega_2\geq 0$, $\omega_1\geq\omega_2$, $\omega_1+\omega_2\leq\pi$ allows understanding complete characteristics of the bispectrum. Figure 3.8 (b) visualizes bispectrum for real processes, including its 12 symmetry regions.



(a) (b)

Figure 3.8: Symmetry Regions of: (a) Third-order Moment; (b) Bispectrum.

3.2.1.2 Linear versus Nonlinear Systems

Let a LTI system with input and output $\{X(t)\}$ and $\{Y(t)\}$

$$\begin{aligned} X(t)&=\sum_m A_m \exp\{(j\lambda_m t+\phi_m)\}\\ Y(t)&=\sum_m B_m \exp\{(j\lambda_m t+\theta_m)\} \end{aligned} \quad \textbf{(3.34)}$$

, where $\{\phi_m\}$ are independent, identically distributed (i.i.d) random variables, uniformly distributed over $[-\pi, +\pi]$ (Nikias & Petropulu, 1993). As described in Section 3.1.1, the output is the superposition of sinusoidal signals with the same frequencies as the input sinusoids.

Using a nonlinear system, the input signal $\{X(t)\}$ passes through a nonlinear system of a Volterra type, resulting in the output $Z(t)$ that is given by (Nikias & Petropulu, 1993):

$$Z(t) = \sum_{\tau_1=0}^{M-1} \cdots \sum_{\tau_{k=0}}^{M-1} h_{12\cdots N}(\tau_1, \tau_2, \cdots, \tau_k) X(t-\tau_1) X(t-\tau_2) \cdots X(t-\tau_k) \quad \textbf{(3.35)}$$

Equation 3.35 consists of sinusoidal signals of

$$\exp\left\{ j\left[\left(\sum_k \lambda_k \right) t + \left(\sum_k \phi_k \right) t \right] \right\}. \quad \textbf{(3.36)}$$

It is evident that the sinusoidal outputs in the nonlinear system are frequency and phase related to the input. For the second order Volterra filter in Eq. 3.35, $Z(t)$ is given by (Nikias & Petropulu, 1993):

$$Z(t) = \sum_m \sum_n C_m C_n \exp\{ j[(\lambda_m + \lambda_n)k + \phi_m + \phi_n] \} \quad \textbf{(3.37)}$$

The third-order cumulants for the output signal $Y(t)$ corresponding to the linear system given by Eq. 3.34 are identical to zero, whereas the third-order cumulants for the output signal $Z(t)$ corresponding to the nonlinear system given by Eq. 3.37 are different from zero. As a result, the existence of nonzero bispectrum in the output of a system with input $X(t)$ identifies the presence of a quadratic nonlinearity in the system.

3.2.2 Special Concepts for Cepstral Analysis Techniques

The cepstral analysis is an alternative approach to spectral analysis using the cepstrum of the power spectrum (power cepstrum) firstly, introduced by Bogert et al. (1963). The concept of the cepstrum estimation is developed as a special case of homomorphic filtering by Oppenheim (1969). Homomorphic filtering is a class of nonlinear signal processing technique, while their basic characteristic is that they use nonlinearities (mainly the logarithm) to transform convolved or nonlinearly related signals to additive

signals and then to process them by linear filters. The output of the linear filter is transformed afterward by the inverse nonlinearity (Pitas & Venetsanopoulos, 1990). Although the cepstrum of the signal is the straight forward nonlinear transformation, it is extremely strong in terms of properties and applications. Previous works in literature modeled microphone (or recording equipment) influence on speech recording by the convolution of its frequency response and the speech recording. The convolution means the spectrum of any recorded speech segment is the product of the spectrum of the speech recording signal and the device frequency response. Thus, one way to eliminate this convolution is to utilize homomorphic filtering to model the distortion caused by its nonlinear multiplicative transfer functions and its ability to convert it into additive terms.

It has been shown in Section 3.2, that the mobile device could be modeled idyllically using LTI systems in order to identify its frequency response on the call recording signal. Hence, this approach is unable to capture the nonlinearities induced by the nonlinear devices such as power amplifiers and mixers. Section 3.1.1 discusses principles of linear systems and describes the convolution of mobile device frequency response on the input signal spectrum in terms of Eq. 3.4. This indicates that each mobile device leaves its intrinsic fingerprints on the overall recorded call by modifying the spectrum of its corresponding input signal. The Cepstral analysis allows transferring the signal to cepstrum domain to eliminate the nonlinearity in Eq. 3.4. Meanwhile, in Section 3.1.4 the study suggested the use of near-silent segments to control speech and environment influences while modeling the mobile device frequency response. Hence, this section describes adopted techniques for power cepstrum estimation by assuming that the input signal is the random stationary signal. Let $Y(\omega)$ be Fourier transform of the call recording signal, $Y(\omega) = |Y(\omega)|.\exp\{j\phi_y\omega\}$, where the first term $|Y(\omega)|$ is the magnitude and the second term $\exp\{j\phi_h\omega\}$ is the phase of the signal. The complex cepstrum is estimated in

terms of the inverse Fourier transform of the log [$Y(\omega)$] given by (Nikias & Petropulu, 1993):

$$c_y(m) = \frac{1}{2\pi}\int_{-\pi}^{+\pi} \log[Y(\omega)]\exp\{j\omega m\}d\omega \\ = F_1^{-1}[\log[Y(\omega)] + j\phi_y(\omega)] \quad \textbf{(3.38)}$$

Not that F_1^{-1} [1] denotes the 1-d Fourier transform. Hence, the complex cepstrum of the signal in terms of Eq. 3.4 is defined as the inverse Fourier transform of log [$Y(f)$] and is written as,

$$F_1^{-1}[\log(Y(f))] = F_1^{-1}[\log(H(f)X(f))] = F_1^{-1}[\log(H(f))] + F_1^{-1}[\log(X(f))] \quad \textbf{(3.39)}$$

Furthermore, $c_y(m) = c_h(m) + c_x(m)$ and the nonlinear convolution becomes as the summation in a complex cepstrum domain. The feasibility of cepstral analysis in speech/speaker identification as well as acquisition device identification is due to this transformation. Finally, corresponding to the power spectrum of the signal, the power cepstrum is given by (Nikias & Petropulu, 1993):

$$p_y(m) = \frac{1}{2\pi}\int_{-\pi}^{+\pi} \log|Y(\omega)|^2 \exp\{j\omega m\}d\omega \quad \textbf{(3.40)}$$

3.2.2.1 Mel-frequency cepstral coefficients

MFCCs are one of the most attractive features in cepstrum domain and convey significant information about the structure of a signal. Thus, these features are widely used for speaker and speech recognition (Beigi, 2011a). The Mel scale proposed by Stevens et al. (1937) is an alternative nonlinear scaling of the frequency axis which models the nonlinear pitch perception characteristics of the human ear. It provides better resolution at low frequencies and less in high frequencies. For near-silent segments, we assumed the pitch exists in the low-intensity stochastic signal generated by the device, whereby the pitch of such a very short tone burst depends on the shape of its temporal envelope. In addition, the triangular filterbank helps to capture the energy at each critical band and gives a rough approximation of the spectrum shape and then smooths the

harmonic structure. In overall similar to PCA, the effect of filterbank analysis is to produce compact feature space by capturing the spectral envelope. However, there are alternative frequency warping approaches such as Bark, linear, etc. To justify the choice of Mel scale, in Chapter 5 the feature extraction algorithm has also implemented bark and linear filterbank with coarser scale (getting closer to actual DFT) for performance evaluation.

3.2.3 Special Concept for Higher-order Spectral Analysis Techniques

Polyspectra or higher-order spectra consist of higher-order moment spectra and cumulant spectra and can be defined for both deterministic signals and random processes. Nikias and Petropulu (1993) illustrated in their book that moment spectra can be very useful for analysis of deterministic signals (transient and periodic) whereas cumulant spectra are important in the analysis of stochastic signals (stationary, nonstationary). This is because: (a) higher-order cumulant spectra ($n > 2$) are zero if the process is Gaussian and nonzero cumulant spectra allows determine the level of non-Gaussianity; (b) cumulants allow to measure the level of statistical dependence in time series; (c) the cumulant spectrum of the sum of two independent, nonzero mean, stationary random processes equals the sum of their individual cumulant spectra. As a result, it is possible to model mobile device response function through analysis of the higher-order cumulant spectra of near-silent segments of the call recording signal (assumed to be stochastic). In overall, the main motivations behind employing HOSA for modeling the mobile device response function in this study are:

(a) As previously discussed for Gaussian processes all higher-order cumulant spectra ($n > 2$) are identically zero. Moreover, as illustrated in Section 3.1.3 the non-Gaussian input signal is received along with possible sources of additive Gaussian noise (i.e. acoustic noise, thermal noise, channel noise, processing noise), whereby a transform

to a higher-order cumulant domain theoretically suppresses the noise. For example, according to Nikias and Petropulu (1993), the bispectrum suppresses non-Gaussian noise with symmetric probability density function.

(b) HOSA reconstructs the true phase and magnitude response of signals or systems. The majority of previous works focused on second-order statistics, whereby the power spectrum domain eliminates phase information. An accurate reconstruction of the phase information for the power spectrum domain is only possible for the special case of minimum phase information, whereas for non-minimum phase signal it is possible to reconstruct true magnitude and non-minimum phase of the signal using higher-order spectrum domains.

(c) As discussed in Section 3.2 the mobile device could be modeled as the nonlinear system because it consists of nonlinear devices (i.e. microphone, PAs, mixers) in its wireless communication signal processing pipeline. Thus, higher-order spectrum domain allows to detect and characterize nonlinearities in call recording signal due to transmitting mobile device response function.

3.2.3.1 Power Amplifiers

The PA performs power amplification by multiplying the signal by a gain factor that results in an amplified signal whose power is much higher than the input signal. In practice, PAs have the maximum output power that is determined by the DC input power. Thus, if the PAs operate near its saturation region of its characteristics, the apparent gain of the PA reduces with increasing the input power and hence, the PA is considered nonlinear. Otherwise, the PA is considered linear if it is operated in a power range within the linear amplification range of its characteristics. In most cases, it is hard to avoid nonlinear distortion that is introduced at the output of the PA because to obtain the maximum power efficiency, the PAs are operated near its saturation region. The nonlinear distortions in PAs are characterized by two different curves known as Amplitude

Modulation–Amplitude Modulation (AM–AM) conversion characteristics and Amplitude Modulation–Phase Modulation (AM–PM) characteristics, as shown in Figure 3.9. In the AM-AM characteristics, the gain of the PA remains constant with increasing input power up to the saturation point where the gain drops. In the AM-PM curve, the phase of the output signal moved away from the input phase by an angle that depends on the input signal power. This curve allows determining the phase distortion that is introduced by the PAs.

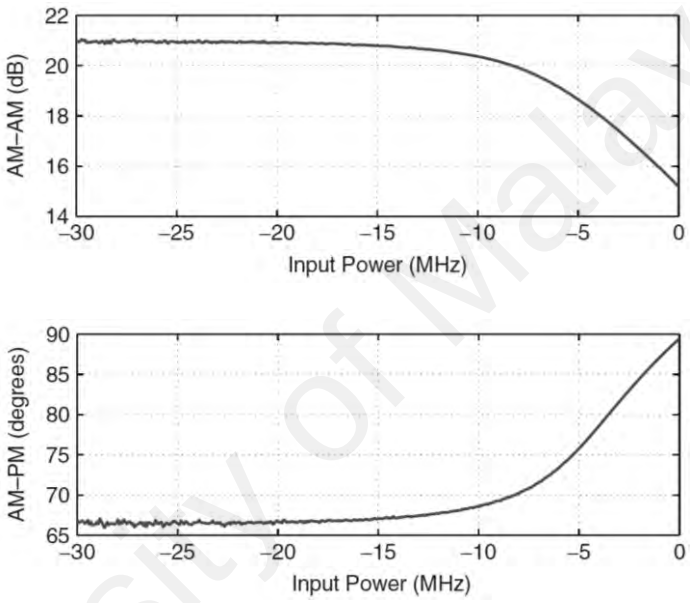


Figure 3.9: AM–AM and AM–PM Conversions (Gharaibeh, 2011).

3.2.3.2 Mixers

A mixer is a product device employed for frequency up-conversion process at the mobile device transmitter. Hence, it is intrinsically nonlinear for normal operation in terms of second order nonlinearity. However, mixers usually generate undesired higher-order nonlinearity in addition to the desired second order nonlinearity.

3.2.3.3 Quadratic Phase Coupling

Second-order nonlinearities are responsible for producing specific phase relations known as quadratic phase coupling (QPC). This phenomenon occurs due to the interaction between two harmonic components causing a contribution to the power at their sum

and/or difference frequencies. Hence, the QPC occurs only between harmonically related components. Three frequencies are harmonically related if one frequency is the result of sum or difference of the other two frequencies. The distortions generated due to nonlinear devices introduced in Sections 3.2.3.1 and 3.2.3.2, can be categorized into harmonic distortion, intermodulation distortion, and difference-frequency distortion. At this point, the intermodulation effect generates an output signal made of sums and differences of the input signals fundamental frequencies and their harmonics, that is $\omega_1 \pm \omega_2$, $\omega_2 \pm 2\omega_1$, $\omega_2 \pm 3\omega_1$, etc. Because the power spectrum of the signal lacks phase information, it is unreliable for identifying if the peaks at harmonically related positions are actually phase coupled. On the contrary, the third-moment sequence of the nonlinear signal is computed using only phase coupled components. As a result, a bispectrum is an effective tool for identifying QPC and distinguishing phase coupled components from non-phase coupled components. Section 3.2.3.6 discusses the techniques for identifying and measuring the nonlinearity based on the normalized bispectrum or bicoherence spectrum.

3.2.3.4 Bicoherence

Bicoherence is an effective function that can be computed through normalizing the bispectrum relating to the corresponding power spectrum. The bicoherence is the second-order coherency index and computed as (Nikias & Petropulu, 1993):

$$BIC^x \stackrel{\Delta}{=} \frac{C_3^x(\omega_1, \omega_2)}{[C_2^x(\omega_1).C_2^x(\omega_2).C_2^x(\omega_1 + \omega_2)]^{1/2}} \quad \textbf{(3.41)}$$

This function is very effective because it can identify and characterize nonlinearities in call recording signal through phase relations of their harmonic components. Later, in Section 3.2.3.5 the nth-order coherency index is utilized for determining the phase response of non-Gaussian linear processes, whereby the spectra of such processes are modeled by the same linear filter.

3.2.3.5 Test of Gaussianity and Linearity of the Signal

As discussed in Section 3.1.2, the bispectrum allows monitoring the Gaussianity and linearity of the signal. In Section 3.1.3, the control system model of the wireless communication between the mobile device transmitter and stationary receiver is discussed. It has been shown that the mobile device could be modeled by using both linear (i.e. ADC) and nonlinear (i.e. PA, mixer) systems in order to identify its frequency response on the call recording signal. Although the mobile device could be modeled as LTI system, its frequency response generates convolution on the recorded signal. Hence, it is expected that the call recording signal is inherently nonlinear. To investigate this statement, the hypothesis testing algorithm was adapted from (Hinich, 1982), to test Gaussianity and linearity of the call recording signal. For example, if the sample estimates of the bispectrum are asymptotically Gaussian, then sample estimates of the squared bispectrum are a chi-squared distribution with two degrees of freedom. As a result, if sample estimate of the bispectrum is zero, the statistic test determines the probability of false acceptance (PFA) by checking the consistency of the sample estimates of the squared bispectrum with chi-squared distribution. This metric is the probability of the wrong assumption of a nonzero bispectrum for a data. For example, if the PFA is only 0.95, the assumption of zero bispectrum is acceptable, or in another word, it is impossible to reject the Gaussianity assumption. Based on this discussion, a hypothesis testing problem for non-Gaussianity is given by:

H1: the bispectrum of $y(n)$ is nonzero;

H0: the bispectrum of $y(n)$ is zero.

If the hypothesis H1 holds, the new algorithm will test for linearity using a second hypothesis testing problem. The test is performed based on the fact that the process is linear and non-Gaussian if its bicoherence is a nonzero constant. Let the squared

bicoherence be chi-squared distributed with two degrees of freedom, the second hypothesis testing estimates and compares the sample interquartile range, R, of the squared bicoherence with the theoretical interquartile range of a chi-squared distribution with two degrees of freedom and noncentrality parameter. If the estimated interquartile range is much larger or much smaller than the theoretical value, then the linearity hypothesis will be rejected. Based on this discussion, a second hypothesis testing problem for linearity is given by:

H1': the bicoherence of *y* (*n*) is not constant;

H0': the bicoherence of *y* (*n*) is a constant.

If the hypothesis H0' holds, the process is linear. The HOSA Toolbox (Swami et al., 1995) implemented the aforementioned detection statistics for Gaussianity and linearity test using MATLAB routine glstat. A disadvantage of the Gaussianity test appears when the test is applied to each of the bifrequencies in the principal domain of the squared bicoherence plot. At this point, the PFA accumulates due to the existence of the spurious peaks in the principal domain. Thus, it overestimates the number of bifrequencies in which the bicoherence magnitude is significant. Appendix D1 details the observations achieved from applying this routine on different VoIP/GSM call recording signals recorded in different environments.

In order to reduce the PFA, Choudhury et al. (2008) suggested an alternative statistical test to check for the significance of bicoherence magnitude at each individual bifrequency. Let $bic^2_{significant}$ be the bicoherence values that satisfy

$$P\{bic^2(f_1, f_2) > \frac{c_\alpha^{x^2}}{2K}\} = \alpha, \quad \textbf{(3.42)}$$

, where *K* is the number of data segments used in the bicoherence estimation and $c_\alpha^{x^2}$ is the critical value calculated from the central χ^2 distribution table for a significance level

of α with two degrees of freedom. Thus, the hypothesis test for non-Gaussianity is reassessed based on the non-Gaussianity index (NGI) as given by (Choudhury et al., 2008):

$$NGI \stackrel{\Delta}{=} \frac{\sum bic^{2}{}_{significant}}{L} - \frac{c_{\alpha}^{x^2}}{2KL}, \tag{3.43}$$

where L is the number of $bic^2_{significant}$. Therefore, the hypothesis test is automated using a decision was made based on the following rule-based scenarios:

R0: if NGI ≤ 0, the signal is Gaussian

R1: if NGI > 0, the signal is Non-Gaussian.

Subsequently, Choudhury et al. (2006) modified the linearity test based on the newly defined metric known as nonlinearity index (NLI) to avoid false negatives due to the existence of few sharp peaks as given by (Choudhury et al., 2006):

$$NLI \stackrel{\Delta}{=} \hat{bic}^2_{\max} - \left(\overline{\hat{bic}^2_{robust}} + 2\sigma_{\hat{bic}^2, robust} \right), \tag{3.44}$$

where $\overline{\hat{bic}^2_{robust}}$ and $2\sigma_{\hat{bic}^2, robust}$ are correspondingly the robust mean and the robust standard deviation of the estimated bicoherence square. These values are calculated by excluding the highest and the lowest Q% bicoherence values. Therefore, the hypothesis test is automated using a decision was made based on the following rule-based scenarios:

R0: if NLI ≤ 0, the signal generating process is linear

R1: if NLI > 0, the signal generating process is nonlinear.

In Appendix D2, the study implements non-Gaussianity and nonlinearity hypothesis tests using two new indices of the NGI and the NLI, then evaluates and compares its observations with conventional methods implemented in HOSA.

3.2.3.6 Bicoherence-based Measure of Nonlinearity

The bicoherence based measure of nonlinearity is implemented in (Choudhury et al., 2008), using new index denoted as the total nonlinearity index (TNLI). Because it has been assumed that the call recording signal is nonlinear, it is required to measure nonlinearity as a metric, especially when there is a need to compare the level of nonlinearities for recorded calls received from the different unit, model, and brands of mobile devices. If a signal is detected as non-Gaussian and nonlinear through the detection statistics discussed in the previous section, then the total nonlinearity present in the signal can be measured through the following new index as given by (Choudhury et al., 2008):

$$TNLI \stackrel{\Delta}{=} \sum bic^2_{significant}, \tag{3.45}$$

In Eq. 3.45, the TNLI is limited between 0 and L, where *L* is the number of $bic^2_{significant}$. In Chapter 5, the study implements TNLI to measure and compare the nonlinearity of the call recording signals.

3.3 Summary

The main focus of this chapter was to study and compare the principles for implementing the audio source mobile device identification framework as follows:

(a) *Linear system*: A system is defined as linear if it satisfies the superposition principle.
(b) *Nonlinear system*: The system is nonlinear if there is no relationship between its input and output.
(c) *Control system model*: A control system model describes the wireless communication signal processing pipeline to study different sources of nonlinear distortions during the call recording process. To distinguish nonlinear distortions resulting from mobile device subsystems during transmission, the proposed framework followed a specific protocol to enforce specific restrictions in applied methodology.

Moreover, the following concepts have been adopted for identifying mobile device response function using call recording signal:

(a) *Cepstral analysis techniques*: The cepstral analysis is an alternative approach to spectral analysis using the cepstrum of the power spectrum (power cepstrum). The concept of the cepstrum estimation is developed as a special case of homomorphic filtering. If the mobile device is modeled as the linear system, the mobile device impulse response is modeled as the convolution on the input signal spectrum. Thus, one way to eliminate this convolution is to utilize homomorphic filtering to model the distortion caused by its nonlinear multiplicative transfer functions and its ability to convert it into additive terms.

(b) *HOSA techniques*: Polyspectra or higher-order spectra consist of higher-order moment spectra and cumulant spectra and can be defined for both deterministic signals and random processes. It is possible to model mobile device frequency response through analysis of near-silent segments of the call recording signal (assumed to be stochastic) using higher-order cumulant spectra. The mobile device is modeled as the nonlinear system, whereby the quadratic nonlinearity due to intermodulation distortion produces quadratic phase coupling between harmonically related components. Third-order cumulant spectra known as bispectrum is an effective tool for identifying QPC and distinguishing phase coupled components from non-phase coupled components.

CHAPTER 4: METHODOLOGY

Having studied the literature and adopted spectral analysis techniques to estimate the mobile device's frequency response on call recording signal, this chapter details the methodology to design the proposed framework. The aim of the framework is to estimate acoustic features based upon the assumptions and considerations together with selected techniques and models. Optimizing acoustic features are important as it provides the autonomous mode in the machine learning methods in audio source mobile device identification. To support the process, this chapter details the procedures in the data collection and test setup, pre-processing and data preparation, feature extraction and analyses process as well as the rationale behind their implementation. The discussion continues with a detailed description of the main and sub-models that support the proposed framework.

The proposed framework attempts to identify the individual, model or brand of mobile devices, by using audio mining techniques. As discussed in Section 2.3, the majority of audio source device identification approaches implemented six hierarchical audio mining stages. Selecting the suitable strategies for each stage to adopt in the framework is essential, especially when dealing with the technical aspects of the selected strategy.

The proposed framework is established with two different feature optimization strategies: the entropy of Mel-cepstrum coefficients (entropy-MFCC) and Zernike moments of Bicoherence (ZMBic). With the aid of the cepstral analysis techniques, as well as principles and concepts identified in the previous chapters, entropy-MFCC is determined as a strong feature set which supports the procedures in the pattern recognition process and aims to identify mobile device unit, model or brand. Furthermore, ZMBic is a novel feature set to offer a strongly robust pattern recognition strategy for identifying mobile device unit, model or brand. This feature set was computed through HOSA

techniques, whereby the methodology, concepts, and motivations were discussed in the previous chapter.

Finally, in order to establish the comparison between different strategies in proposed audio mining hierarchy, a multi-strategy audio source mobile device identification framework is proposed. As a first attempt to identify transmitting mobile devices using call recording, the framework employs both feature sets one at a time with other components, in order to satisfy the objectives of this study. Furthermore, in addition to the main objective to optimize acoustic features, the framework also addresses the limitations of the audio source device identification that were identified in previous chapters.

4.1 The Proposed Framework

The proposed framework includes six components that are discussed in each section. Section 4.1.1 explains the data collection and test setup. Section 4.1.2 discusses the data preparation step through precise preprocessing algorithms. Preprocessing increases the quality and quantity of the data instances to achieve high performance (Bhatt & Kankanhalli, 2011). Section 4.1.3 and 4.1.4 describes the proposed feature extraction process, including the computation steps of entropy-MFCC and ZMBic features in addition to selected features from literature research. Section 4.1.5 details supervised and unsupervised learning methods that were implemented for benchmarking classification and clustering algorithms, respectively. Moreover, this section outlines the open set source mobile device identification strategy for practical implementation. Finally, Section 4.1.6 entitles the adopted performance metrics for evaluating the proposed framework and comparison with state-of-the-art approaches. In general, the complete steps of the proposed source mobile device identification framework are shown in Figure 4.1.

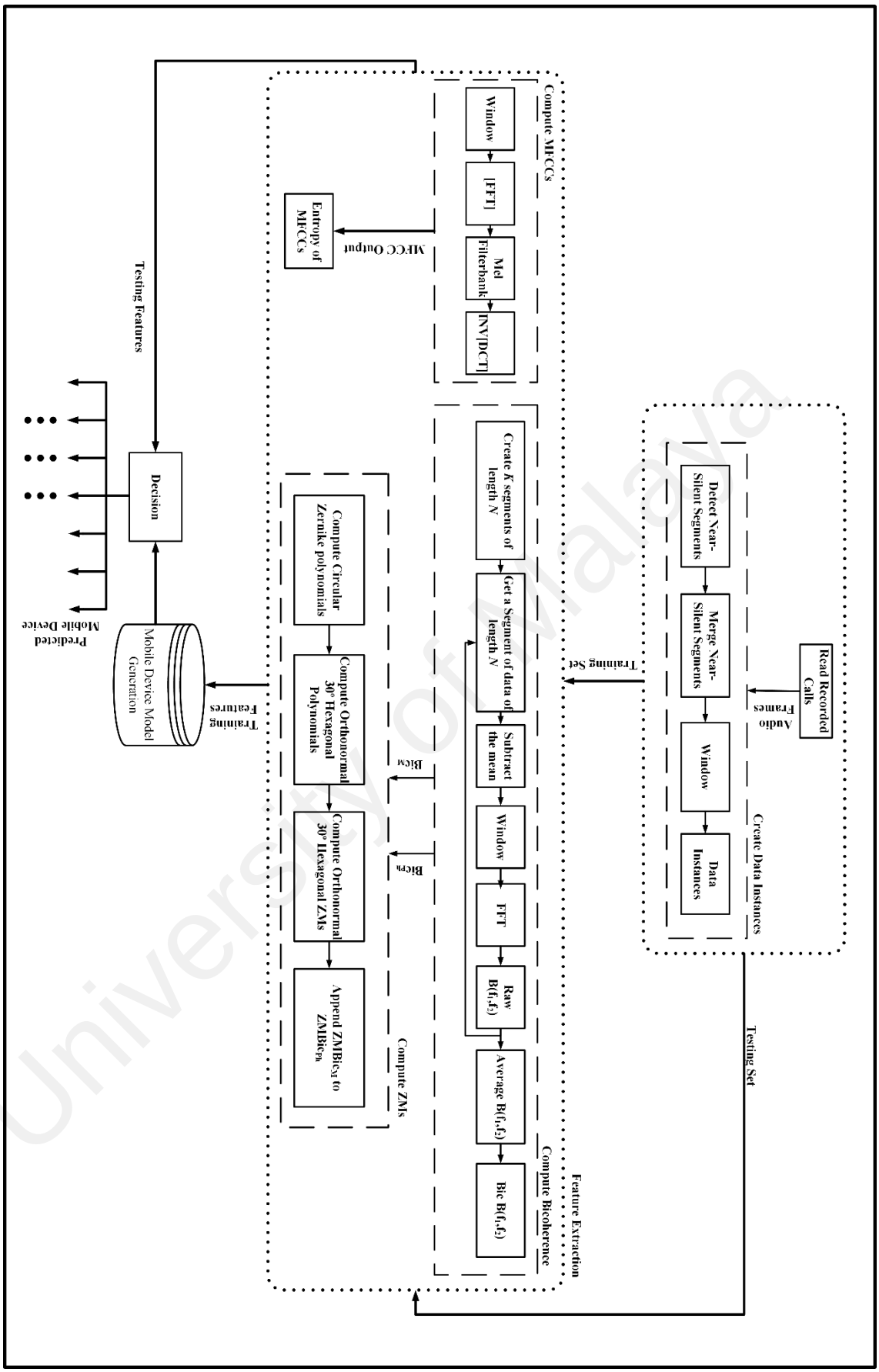


Figure 4.1: Control Flow Diagram of the Proposed Framework

4.1.1 Data Collection and Test Setup

The dataset includes four different call recording sets that were collected by using different test setups, as listed in Appendix B1. In conducting the experiments in Chapter 5, each experimental phase uses one or more types of the dataset with specific characteristics.

4.1.1.1 Dataset 1

The setup recorded a total of 25 Skype calls for each mobile device, whereby both the mobile device and the stationary were located in a truly silent office environment. The devices are listed in Appendix B2. The silent session eliminates the possible convolutions caused by speech signals generated by different speakers. The MP3 Skype call recorder v.3.1 freeware application (2013) records the signals in the '.mp3' format. The recorded files were converted to the '.wav' format prior to sampling and data preparation.

4.1.1.2 Dataset 2

This dataset was collected during VoIP communication with a total of 21 mobile devices. The controlled setup was similar to DS1 with the exception that a total of 25 call sessions were recorded in stereo mode, each with the duration of 10 seconds. The details of mobile devices for this dataset are listed in Appendix B3.

4.1.1.3 Dataset 3

An important aspect of developing and evaluating the source mobile device identification based on recorded call is the collection of a suitable benchmark database. The database should be able to capture characteristics of the model or possibly device rather than speech contents, recording environments or stationary device. Hence, the setup consists of one laboratory room for the stationaries in addition with two indoor and two outdoor locations in the main building of the Faculty of Computer Science and Information Technology, the University of Malaya for the mobile devices. A plan of these

recording locations is shown in Figure 4.2. Table 4.1 summarizes the description of the recording environments. Appendix B4 summarizes the 12 selected mobile device models, the number of respective devices, and basic mobile device specifications. The stationary devices' specification was listed in Appendix B1.

Table 4.1: Call Recording Environments in DS3

Environment ID	Description
A	Multimedia Research Lab
B	Corridor First Floor
C	Foyer Ground Floor
D	Main Door Block B
E	Cafeteria

Figure 4.2: Recording Locations and Setup for DS3

In overall, DS3 includes 12 mobile device models with 10 available devices for each model. This is the substantial effort in comparison to state-of-the-art works for audio acquisition device identification using a small subset of cell phone or mobile devices, as listed in Chapter 2, Table 2.13. Moreover, the DS3 dataset consists of both VoIP and cellular call recordings. For the VoIP call setup, the male operator logged on to the Skype

through the Dell or iMac stationaries, and then called the female operator. Subsequently, the female operator who was logged on to the Skype through the query mobile device answered the call, while the male operator recorded the call for the duration of approximately 150 seconds. For the cellular call setup, the male operator who possessed the Nokia Lumia stationary that is connected to the GSM called the carrier number on the mobile device, which is possessed by the female operator. Subsequently, the female operator answered the GSM call, while the male operator recorded the call for the duration of approximately 150 seconds. Meanwhile, the speech conversations were selected from the English practice texts, repeated by the operators, and recorded by each stationary one at a time corresponding to all possible mobile device locations. Therefore, DS3 includes three different recording sets:

RS1- VoIP to VoIP Skype calls from Dell Precision stationary to mobile devices located in environments B-E. The speech conversations were recorded in the mono '.wav' format by using PAMELA for Skype-Version 4.8.

RS2- VoIP to VoIP Skype calls from iMac stationary to mobile devices located in environments B-E. For Apple iPhone and Samsung Galaxy devices, the speech conversations were recorded with Vodburner in the mono '.wav' format, whereas for Sony Xperia devices the speech conversations were recorded with Callnote in the mono '.mp3' format. Vodburner was replaced with Callnote because Skype suddenly blocked access to the third party applications such as Vodburner. To eliminate the effects of the compression the files could be converted to the '.wav' format prior to sampling and data preparation.

RS3- Cellular to Cellular calls from Nokia Lumia 710 stationary to mobile devices located in environments B-E. The speech conversations were recorded using Windows phone call recorder application in stereo with the '.mp4' format.

In overall, the total of 1,440 calls was recorded, whereby the calls were received from 120 different units of mobile devices in four different environments by three different stationaries.

Analyzing the source mobile device identification framework requires a set $\mathcal{R}_{\text{train}} \subset \mathcal{R}$ of call recordings to train and a set $\mathcal{R}_{\text{test}}$ of call recordings to evaluate the detection performance of the employed machine learning algorithm. Let the utilized mobile device (*d*), stationary device (*s*) and environments (*e*) assign each call recording $r \in \mathcal{R}$ to its representative set:

$$\mathcal{R}_{train,\ test} = \{r_{i,j,k}^{(m)} \mid d_i \in \mathcal{D}_{train,test}^{(m)} \wedge s_j \in S_{train,test} \wedge e_k \in \mathcal{E}_{train,test}\}, \quad \textbf{(4.1)}$$

where $m \in \mathcal{M}$={*i4, i4S, i5, i5S, S3, S3M, S4, N3, N4, C, Z, Z1* } represents all mobile device models in the database and i={1, 2, 3, …, 10} represents the index of mobile device units per model. Furthermore, $s_j \in S$={*Dell, iMac, Nokia Lumia*} and $e_k \in \mathcal{E}$={*B, C, D, E*} denotes stationary device set and environment set for j=1…3 and k=1…4, respectively. A set $\mathcal{R}_{\text{train}}$ and $\mathcal{R}_{\text{test}}$ differ with respect to the objectives of the evaluation experiment. Table 4.2 demonstrates the representative sets of $\mathcal{R}_{\text{train}}$ and $\mathcal{R}_{\text{test}}$ corresponding to all experiments in this study.

Table 4.2: Description of the Recording Sets Assigned to Training and Test Sets

Experiment Scenario	$\mathcal{R}_{\textbf{train, test}}$
Intra/inter-mobile device model similarity	$\mathcal{R}_{train,test} = \{r_{i,j,k}^{(m)} \mid d_{1..10} \in \mathcal{D}_{train,test}^{(m)} \wedge s_1 \in S_{train,test} \wedge e_1 \in \mathcal{E}_{train,test}\}$
Mobile device model identification in closed set	$\mathcal{R}_{train,\ test} = \{r_{i,j,k}^{(m)} \mid d_i \in \mathcal{D}_{train,test}^{(m)} \wedge s_j \in S_{train,test} \wedge e_k \in \mathcal{E}_{train,test}\}$
Individual mobile device identification in closed set	$\mathcal{R}_{train,\ test} = \{r_{i,j,k}^{(m)} \mid d_i \in \mathcal{D}_{train,test}^{(m)} \wedge s_j \in S_{train,test} \wedge e_k \in \mathcal{E}_{train,test}\}$
Mobile device model identification in open set	$\mathcal{R}_{train} = \{r_{i,j,k}^{(m)} \mid d_{1..7} \in \mathcal{D}_{train,test}^{(m)} \wedge s_j \in S_{train,test} \wedge e_k \in \mathcal{E}_{train,test}\}$ $\mathcal{R}_{test} = \{r_{i,j,k}^{(m)} \mid d_{1..3} \in \mathcal{D}_{train,test}^{(m)} \wedge s_j \in S_{train,test} \wedge e_k \in \mathcal{E}_{train,test}\}$

4.1.1.4 Dataset 4

The uncontrolled test setup recorded speech conversations via VoIP/GSM calls by using different stationaries located in different environments. The calls were transmitted from 20 individual Apple iPhone devices in four different models, whereby the conversations were collected in the real-world basis scenario with different speakers located in unknown environments. The calls for few devices were made with VoIP Skype, whereas the calls for the majority of devices were made through cellular services. The idea was to predict the model of the transmitting mobile devices by using its corresponding recording as the testing dataset. Furthermore, DS3 could be used as the training dataset.

4.1.2 Preprocessing and Data Preparation

The preprocessing algorithm used two different strategies to create data instances from (a) speech recording signal, and (b) near-silent segments. This is to demonstrate the importance of the near-silent detection process prior to feature extraction for both feature sets.

4.1.2.1 Speech Recording Signal

The preprocessing stage includes sampling, framing, windowing and cleaning the signal. The speech signals become more distinct when environmental additive noise is eliminated. Thus, for entropy-MFCC features, the data preparation algorithm was utilized to enhance the audio signals through minimum mean-square error-based noise power estimation approach, as proposed in (Gerkmann & Hendriks, 2012). Spectral enhancement aims to remove the non-stationary noise corruptions produced by the environment. This method was originally proposed for speech enhancement to reduce the additive noise without reducing speech ineligibility. The method uses the speech presence probability approach with fixed non-adoptive *a priori* SNR. This value is selected based on the SNR that is typical in speech presence. The key advantage of this enhancement

technique is its low overestimation of the spectral noise power and its relatively low computational complexity. Figure 4.3 compares a noisy speech signal with a clean reference signal by using SNR function, in which the top and middle plots show and compare the speech signal against its reference clean signal, respectively. In addition, the bottom plot shows the global SNR of 22.7 dB and segmental SNR of 19.2 dB.

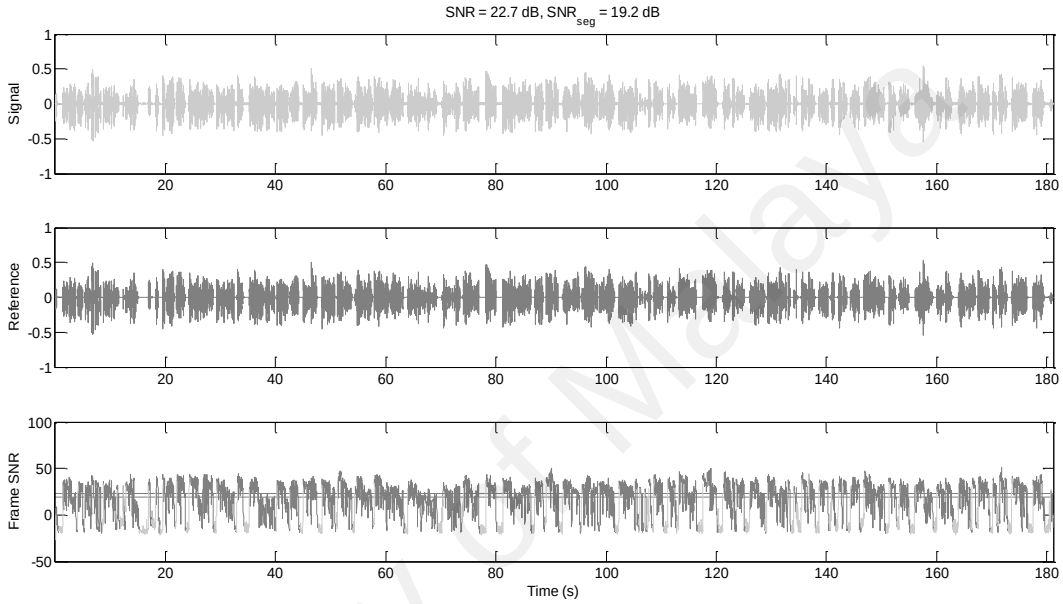


Figure 4.3: Visualization of the Segmental SNR

The power spectrum magnitude of the speech signals prior and after enhancement is computed via the FFT method and is illustrated in Figure 4.4. The FFT method computes the DFT of the input signal with FFT length of 1024 samples. The speech signals are from the call recording samples that were collected during the call from two different units of *Apple iPhone 5* located in four different environments with different noise conditions. Because all four environments are located in indoor and outdoor university campus, their SNRs are narrowed between 18 to 20 dB. In the overall, it is evident that the signals become more distinct when additive noise is eliminated. However, for extracting ZMBic features, the enhancement process is unnecessary because as discussed in Section 3.2.3, transferring to the second-order cumulant domain automatically suppresses the additive Gaussian noise.

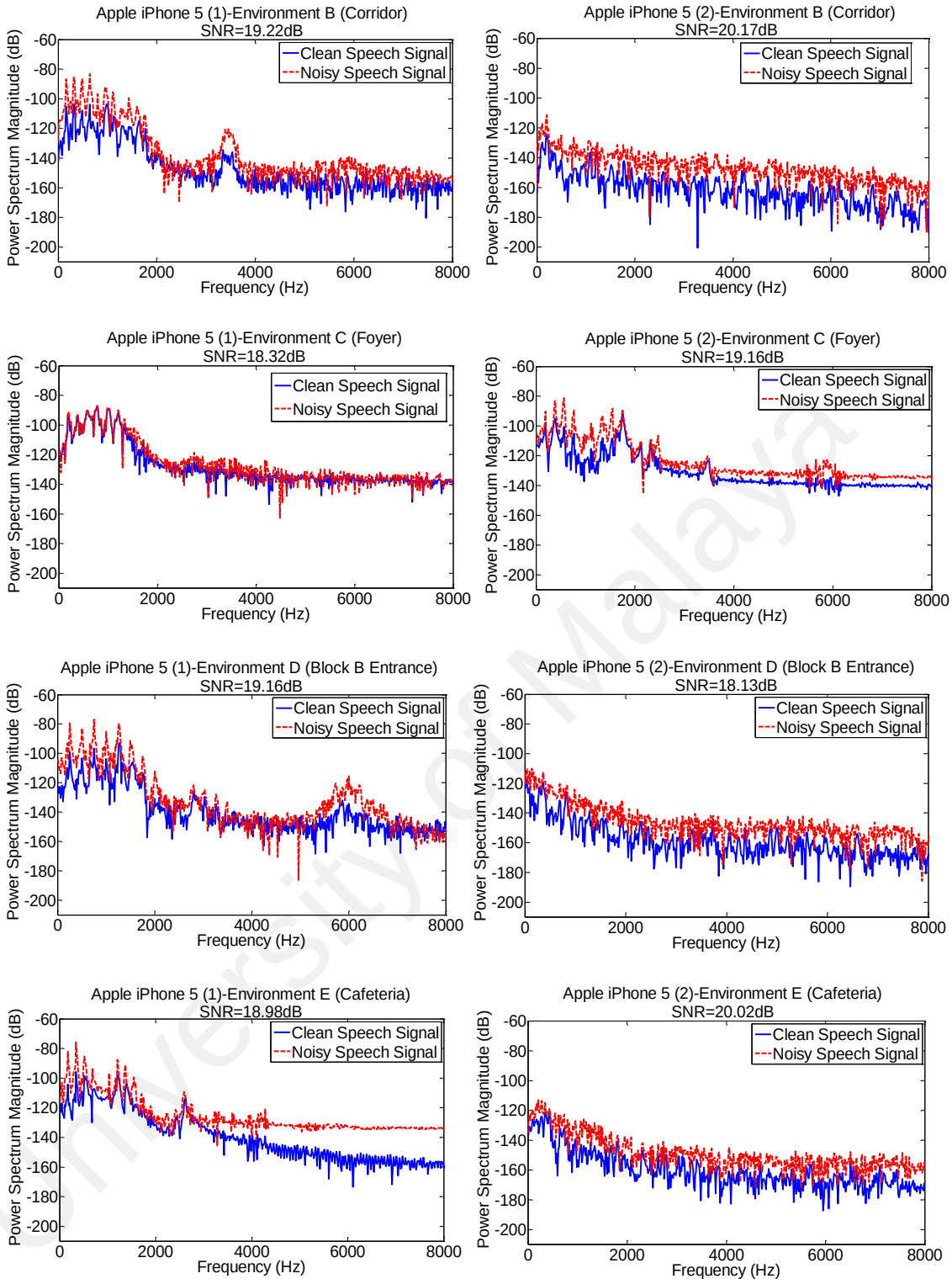


Figure 4.4: Power Spectrum Visualization of the Noisy versus Clean Signal

To allow comparison between the power spectrum and bispectrum of the speech signal, Figure 4.5 demonstrates the bispectrum of the audio frame of length one second corresponding to the call recording signals visualized in Figure 4.4. The signal bispectrum

was estimated via FFT direct approach with FFT length of 256 samples, Rao-Gabor optimal windows, 128 samples per segment and no overlap.

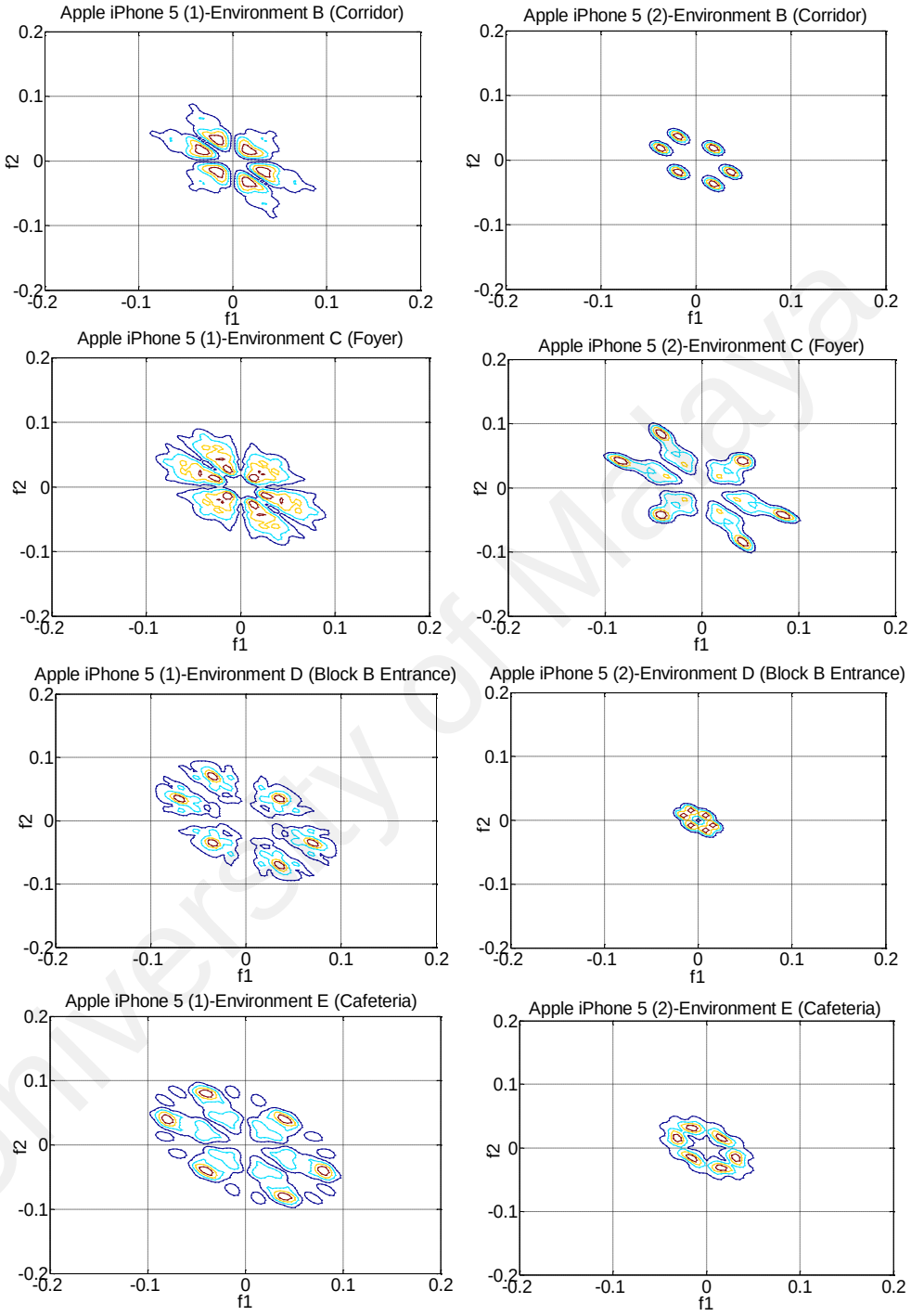


Figure 4.5: Bispectrum Visualization of the Speech Signal

In this stage, the data preparation algorithm breaks the signals into shortened audio frames of length one seconds. The short audio frames are the data instances for feature

extraction. The experimental results in Chapter 5, Section 5.2.2.1 will provide justification on the choices made during data preparation.

4.1.2.2 Near-Silent Segments

Near-silent detection algorithm uses simple segmentation of the recorded signal in order to extract the near-silent segments according to Figure 4.6. The framework has implemented the near-silent detection algorithm by using the frame by frame analysis of the clean speech signal. On the contrary, to the state-of-the-art silent removal algorithms such as the one implemented in (Giannakopoulos, 2010), the proposed near-silent detection algorithm in the current study extracts the near-silent segments and eliminates the speech segments. The method includes the following steps: (a) splits the speech signal into shortened audio frames of length 0.1 seconds, (b) analyzes each frame to identify frames with a maximum amplitude less than 0.03 as silent, and (c) creates the new signal without speech frames. The silent amplitude threshold of 0.03 was determined based on visualization. Finally, the new silent signal was broken into audio frames of length one seconds prior to feature extraction.

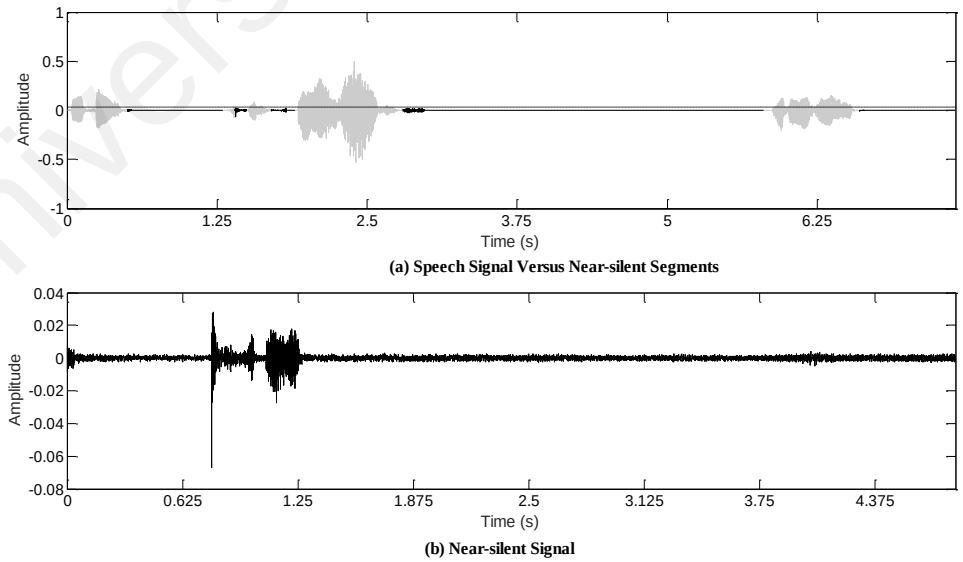


Figure 4.6: Visualization of Near-Silent Detection algorithm

The spectrum of the speech signal corresponding to a call recording sample of seven seconds and its detected silent segments are illustrated in Figure 4.6 (a), in which the

black color determines the near-silent segments, and the horizontal line shows the estimated amplitude threshold. Furthermore, the spectrum of the approximately five seconds of merged silent segments is appeared in Figure 4.6 (b).

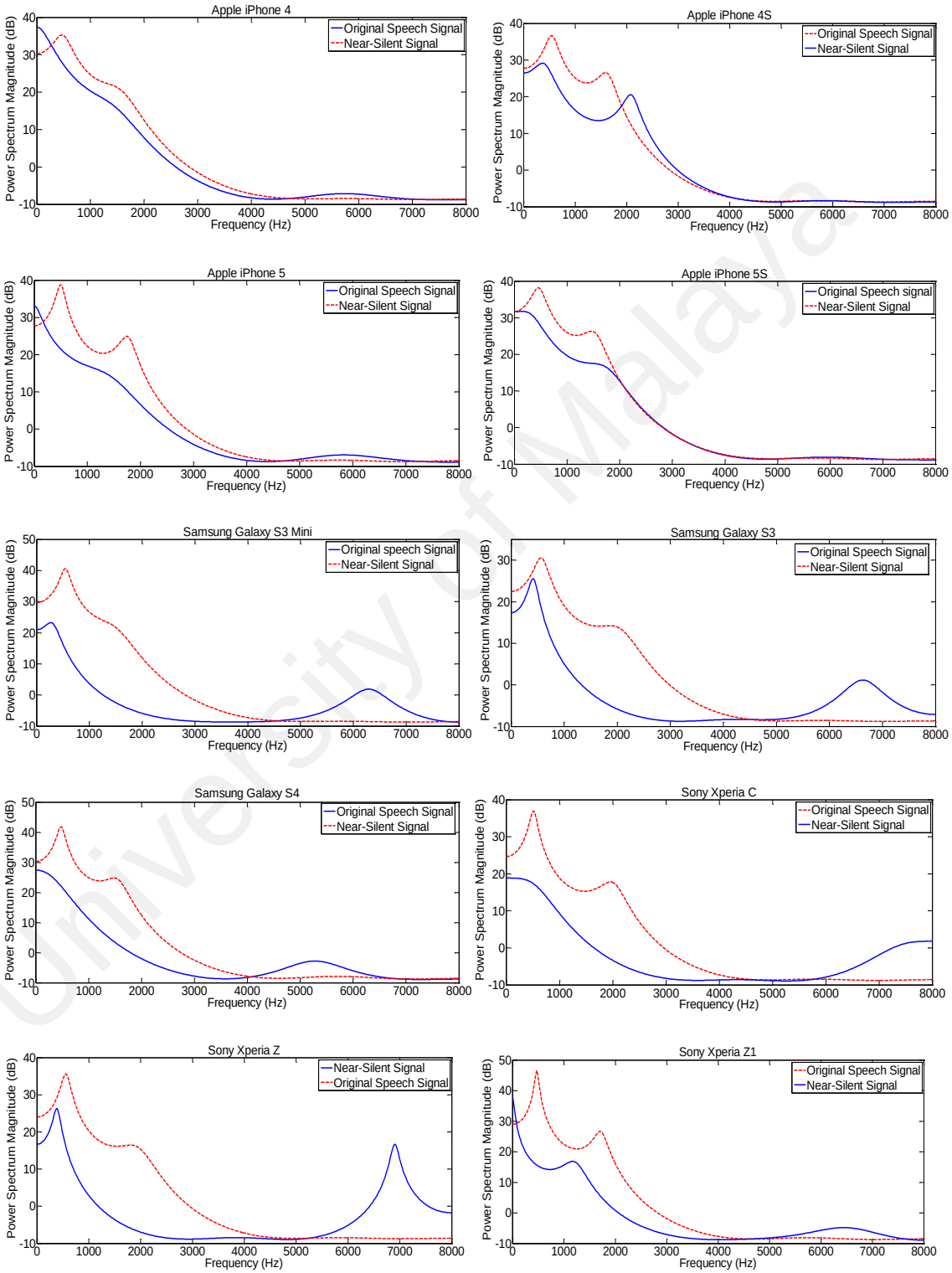


Figure 4.7: Power Spectrum Visualization of the Speech versus Near-Silent Signal

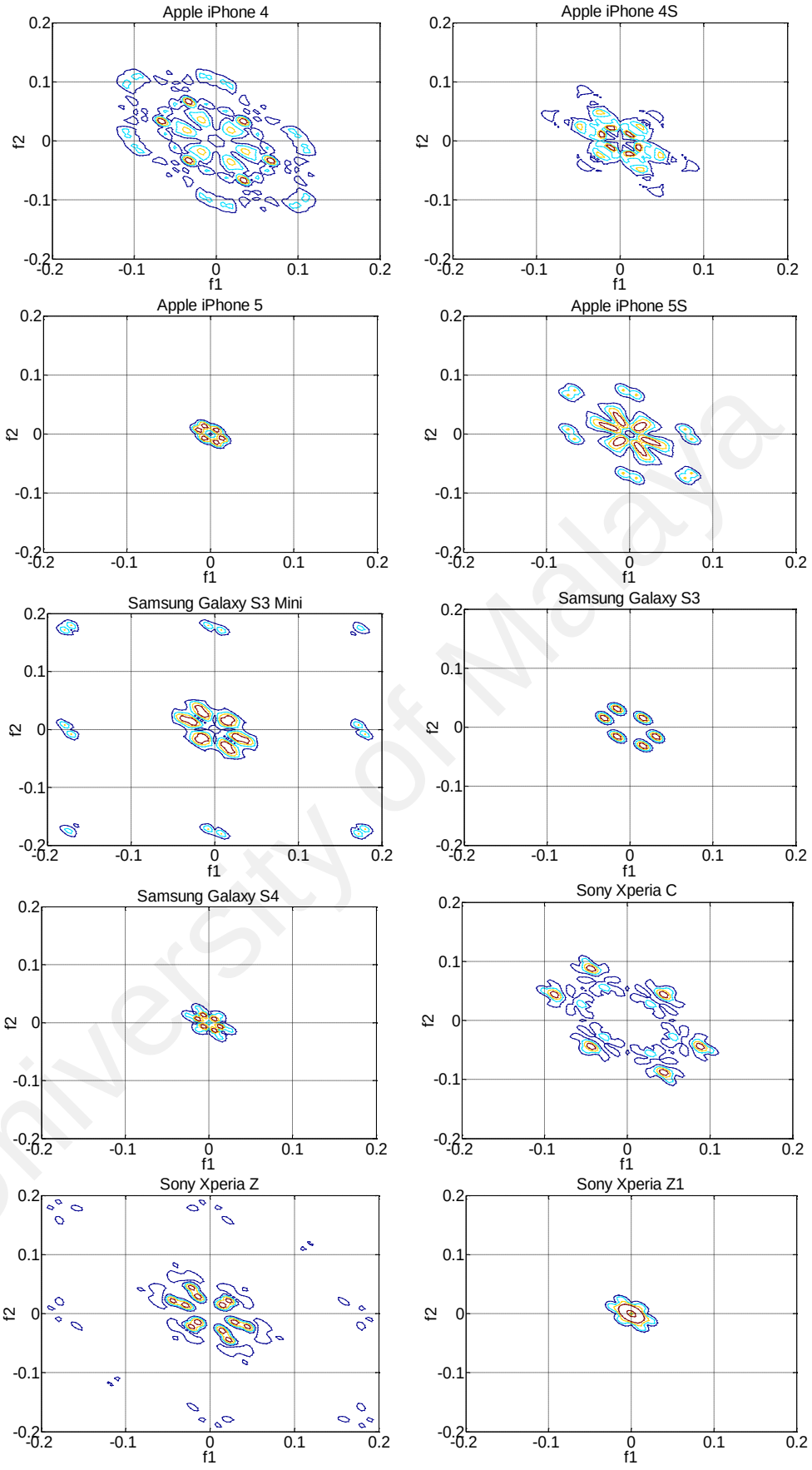


Figure 4.8: Bispectrum Visualization of the Near-Silent Signal

Figure 4.7 compares the power spectrum magnitude of the speech signals of length 170-180 seconds against its corresponding near-silent signals of length 35-50 seconds for 10 different models of mobile devices. For this figure, the power spectrum magnitude was estimated using multiple signal classification (MUSIC) method.

At first, the method determined the eigenvectors and eigenvalues of an estimate of the signal's correlation matrix and then estimated the reciprocal of a weighted sum of the magnitude squared of the FFTs of the eigenvectors in the noise subspace. Moreover, the method utilized the FFT length of 1024 samples, the hamming window, and no overlap. It is evident from Figure 4.7 that the near-silent spectrum corresponding to the recorded call from the specific mobile device model is more distinct in compare with the speech spectrum. Hence, it is plausible to utilize near-silent signal for extracting mobile device intrinsic fingerprints instead of the speech signal. Figure 4.8 visualizes the bispectrum of the near-silent signal corresponding to different mobile device models. In the previous section, the bispectrum of the speech signals was illustrated in Figure 4.4, whereby the signals were related to the recorded calls from two different mobile devices of the identical model located in four different environments. The illustration aimed to compare the bispectrum of the signal corresponding to the same device under different environmental disturbances.

However, Figure 4.7 and Figure 4.8 aim to demonstrate the spectrum and bispectrum differences, respectively among different mobile device models. By eliminating the speech segments and selecting the call recording samples that were recorded in the same environment (corridor), it is possible to assume that the differences are mostly due to response functions generated by mobile devices of different models. Furthermore, computing the entropy-MFCC and ZMBic features allow to determine the accurate inter-class and intra-class similarity distances corresponding to mobile devices of different model and individual mobile devices.

4.1.3 The Feature Extraction Using Cepstral Analysis Techniques

As discussed in Chapter 1, the first objective of this study is to optimize MFCC features to allow the automatic source mobile device identification using call recording. In Chapter 3, the study elaborated the methodologies for modeling the mobile devices as linear systems as well as the special concepts for cepstral analysis techniques to identify such a mobile device response function. Meanwhile, MFCCs are introduced as one of the most attractive feature sets in cepstrum domain that conveys significant information about the structure of a signal. However, because MFCC features are well known to model the context of the audio signal (e.g., speech), the resulting MFCCs from near-silent segments are low in values. Moreover, the Mel filterbank in MFCCs is designed to capture perception of audio signals. Thus, it is plausible to rearrange the Mel filterbanks in order to capture characteristics of the mobile device. Furthermore, the normalization technique is required to intensify the energy of MFCCs with respect to the maximum mutual information corresponding to the mobile device frequency response. Motivated by this, the proposed feature extraction algorithm used entropy of Mel-cepstrum coefficients. The use of entropy allows reducing the dimensionality of the feature space by converting the sequence of feature vectors to a single feature vector. According to information theory, entropy increases for silent segments that contain uncertainty and reduces for the speech segments that contain information (Beigi, 2011c). In order to measure entropy, the proposed framework took advantage of both extensive and non-extensive statistics approach. The traditional extensive statistics such as Shannon entropy (C. E. Shannon, 1949) normally hold the additive property (linear system), whereas the non-extensive statistics, proposed by Tsallis (Tsallis, 1988) allow characterizing the non-additive behavior of complex non-linear systems. The study utilized Tsallis entropy along with Shannon entropy in an attempt to increase robustness and the discriminating ability of the entropy-MFCC feature set. Overall as shown in Figure 4.9, the proposed feature

extraction algorithm: (a) computes the MFCCs, (b) computes the entropy of the feature spectrum, and finally (c) scales the entropy-MFCC features between zero and one. In Chapter 5, Section 5.4 the experimental setup evaluates the feasibility of entropy-MFCC features against entropy of LFCCs and BFCCs, in addition to other statistical moments of MFCCs and GSV of MFCCs through different types of classifier and clustering techniques that were usually used in machine learning applications [e.g., support vector machine (SVM), naïve Bayesian, and logistic regression].

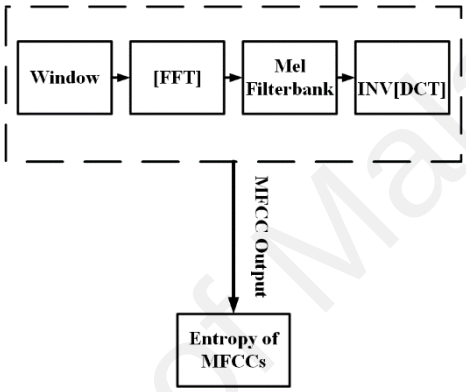


Figure 4.9: Flow Chart of Entropy-MFCC Extraction Technique

4.1.3.1 MFCCs

The complex cepstrum of the signal is defined as the Fourier transform of the log of the signal spectrum (Rabiner & Juang, 1993b). Meanwhile, Mel is short for the word "melody" and is equal to the one-thousandth of the pitch of a simple tone with a frequency of 1000 Hz and amplitude of 40 dB above the auditory threshold (Beigi, 2011b). This relationship is given in Eq. 4.2.

$$f_{mel} = \frac{1000}{\log(2)} \log(1 + \frac{f_{Hz}}{1000}) \quad \textbf{(4.2)}$$

Hence, the signal was transferred to a linear frequency scale by using FFT and was converted to the Mel frequency scale by using a filterbank. The filterbank consists of the triangular shape filters that are set in equidistant centers based on the Mel scale. Using this approach, the Mel-scale spectral magnitude is denoted as (Beigi, 2011b),

$$\left|{}_{l}\breve{H}_{m}\right| = \mathcal{M}_{mk}\left|{}_{l}H_{k}\right| , \tag{4.3}$$

where $\mathcal{M}_{m,k}$ is the $(m,k)^{th}$ element of the matrix $\mathcal{M}$, which is the mapping produced by the triangular filters from the linear frequency scale to the Mel scale, and $\left|{}_{l}H_{k}\right|$ is the magnitude of the spectra in the l^{th} frame. The log spectrum of the signal in the l^{th} frame was then computed as follows (Beigi, 2011b):

$${}_{l}C_{m} = \log\left(\left|{}_{l}\breve{H}_{m}\right|^{2}\right), \tag{4.4}$$

where M is the total number of triangular filters for $m=\{1, \ldots, M\}$ filter coefficients in the filterbank. At last, MFCCs were determined by computing the inverse discrete cosine transform of the short-time Mel frequency log spectrum of the signal, as given by (Beigi, 2011b):

$${}_{l}c_{n} = \sum_{m=0}^{M-1} \alpha_{m}\ {}_{l}C_{m} \cos\left(\frac{\pi(2n+1)m}{2M}\right), \tag{4.5}$$

where ${}_{l}C_{n}$ is the n^{th} MFCC coefficient for the l^{th} frame. In addition, the coefficients a_m are determined as

$$\alpha_{m} = \begin{cases} \frac{1}{M} & for\ m = 0 \\ \frac{2}{M} & \forall\ m \rangle 0 \end{cases}. \tag{4.6}$$

The log energy (average log energy of audio frames), and the first and second derivative of MFCC coefficients could be also included in MFCC feature vector. In overall, the first and second order derivatives are, to some extent, independent of the actual MFCC coefficients and were used to model the local dynamics of the signal. These derivatives are denoted by Delta and Delta-Delta cepstral coefficients. Dynamic features provide information regarding the way of feature vectors change in time. By default the filterbanks in MFCCs are designed to develop 12 zeroth order cepstral coefficients; thus, the feature extraction algorithm for the preliminary test in Section 5.2 computed all 12 MFCCs. Consequently, the Mel-cepstrum output consists of one frame per row, and each

frame includes 12 columns. Section 5.4.1.2 evaluates and compares the overall performance of entropy-MFCCs corresponding to zeroth order cepstral coefficients with and without the log energy, Delta and Delta-Delta coefficients.

Using a different approach, the proposed entropy-MFCC feature set in this study utilized a different number of triangular filters, *M*, in order to set the size of filterbanks prior to feature extraction and compute the different number of zeroth order MFCCs. After running several experiments, as will be discussed in Chapter 5, Section 5.4.1.1, the algorithm selected a most optimal setup with a total of 49 filters in filterbank and computed 48 cepstral coefficients.

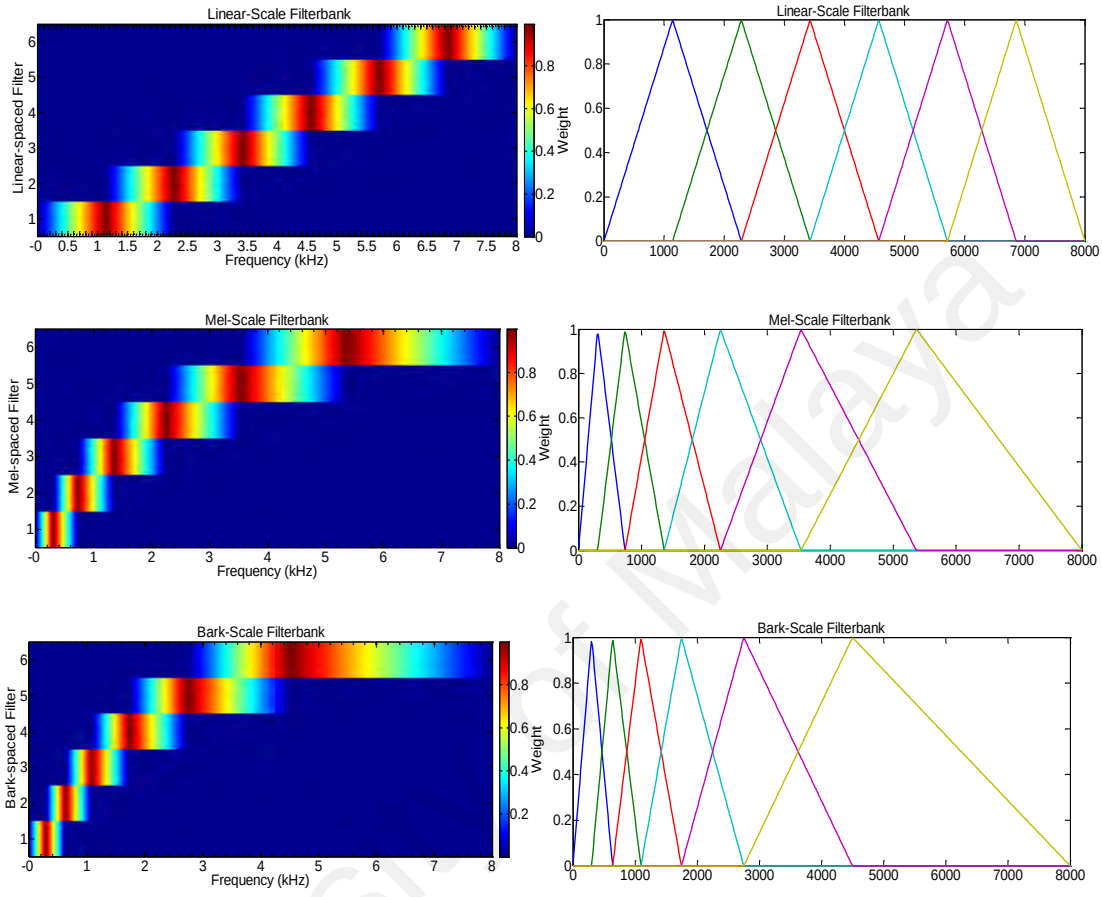
4.1.3.2 LFCCs and BFCCs

For computing the LFCCs, the feature extraction algorithm follows the same processes as illustrated in Figure 4.9 for computing MFCCs except that the Mel filterbank was replaced with the linear filterbank that computes the spacing center frequencies in units of Hertz. Similarly, for computing the BFCCs, the Mel filterbank was replaced with Bark scale filterbank. This is, in principle, a filterbank modeling of the hearing system proposed by Zwicker et al. (1957). The proposed method subdivided the space of frequencies into 24 basic critical bands, where each band has a center frequency and a bandwidth associated with it. Based on this study, one Bark corresponds to the width of one critical band over the whole frequency range and corresponds to nearly a 100 Mel pitch interval. Hence, center frequencies of the filters in Bark scale were computed by (B. J. Shannon & Paliwal, 2003)

$$f_{Bark} = 6\log_e\left(\frac{f_{Hz}}{600} + \sqrt{\frac{f_{Hz}}{600} + 1}\right) \quad \textbf{(4.7)}$$

Figure 4.10 demonstrates an example of the filterbank that consists of six triangular filters located in linear-scale, Mel-scale, and Bark-scale. The filterbanks were determined with FFT length of 1024 samples, a sampling frequency of 16000 as well as upper and lower frequencies of 0 and 8000 Hz, respectively. It is evident that filters located in low

frequencies have the narrow bandwidth whereas, for Mel and Bark scales, filter bandwidths are larger in high frequencies. On the contrary, in linear scale, the bandwidth of each filter is the same.



(a) **(b)**

Figure 4.10: Filterbanks Visualization: (a) Linear, Mel & Bark-Spaced Filters versus Frequency, (b) Calculated Triangular Filters Spaced in Linear, Mel & Bark Scales

4.1.3.3 Entropy

Entropy intensifies the energy of MFCC outputs. Figure 4.11 illustrates the MFCC output $_{l}C_{n}$ and its entropy, where *N* is the total number of frames. In order to optimize the acoustic features for source mobile device identification, at first, the proposed approach utilized the 12 zeroth order MFCCs as set by default and Shannon entropy for feature extraction. After evaluating the experiment results in Phase I (Section 5.2), the source mobile device identification framework was further optimized. In Phase II (Section 5.3), the optimized entropy-MFCC feature set was designed by rearranging the filterbank

spacing and generating a total of 48 zeroth order MFCCs. Later, the entropy of Mel cepstrum coefficients was determined by using both Shannon and Tsallis entropy generating a total of 96 entropy-MFCC features.

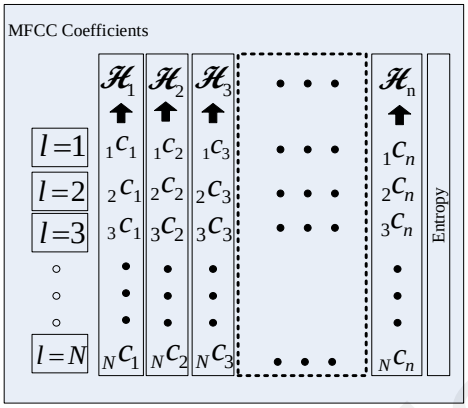


Figure 4.11: Entropy-MFCC Feature Extraction Steps

(a) ***Shannon Entropy***

The feature extraction based on Shannon entropy computed the entropy of MFCC vectors in two stages; first, it computed the probability mass function (PMF) of the MFCC coefficients and then; it computed the entropy $\mathcal{H}$ by using

$$\mathcal{H}_n = -\sum_{l=1}^{N} {}_l p_n \log_2 {}_l p_n, \tag{4.8}$$

where ${}_l p_n$ is the PMF of the n^{th} MFCC coefficient in frame l (C. E. Shannon, 1949).

(b) ***Tsallis Entropy***

For any real parameter q, Tsallis entropy is defined as (Tsallis, 2009):

$$\mathcal{T}_n = \frac{k}{q-1}\left(1-\sum_{l=1}^{N} {}_l p_n^q\right), \tag{4.9}$$

where parameter q is known as the entropic or non-extension index; k is a constant that in the signal processing is set to 1. In the limit that q goes to 1, the standard Boltzman-Gibbs-Shannon measure is achieved. In Section 5.4, the algorithm set q=0.1 after evaluating the entropy-MFCC feature set generated with different parameter settings.

4.1.3.4 Statistical Moments

Statistical moments measure the nature of random variables, their behaviors, and their functions. Despite the availability of the infinite set of statistical moments, this study

implemented the first five moments that are widely used in (Kinnunen et al., 2012; Senan et al., 2011) and (Molla & Hirose, 2004), to compare their performance against the proposed entropy-MFCC feature set. These moments are mean, standard deviation, variance, skewness, and kurtosis. In Chapter 5, Section 5.2.2.2(a) the experimental setup compares the performance of entropy-MFCC features for source mobile device identification with other statistical moments of MFCCs.

4.1.3.5 Gaussian Supervectors

Using a different approach adopted from speaker recognition (Campbell et al., 2006) and acquisition device identification (D. Garcia-Romero & Espy-Wilson, 2009; Hanilçi et al., 2012) literature research, the MFCC sequences could be normalized to supervectors using GMM-UBM. To allow comparison with the entropy-MFCC feature set proposed in this study, here, the following steps were implemented to compute GSV from MFCC sequences:

(a) A large set of training data corresponding to all mobile devices was used to train GMM-UBM in a machine learning fashion (Campbell et al., 2006).

(b) By training the GMM-UBM, the algorithm determined the density model denoted as, $\lambda_{UBM} = (\{w_k, m_k, \Sigma_k\})$ and mapped a call recording signal, parametrized in terms of sequences of MFCCs $X = \{x_t\}_{t=1}^{T}$ with $x_t \in \mathbb{R}^F$, into two supervectors. Meanwhile, w_k is denoted as the mixture weights, m_k is represented as the one-dimensional mean vector, Σ_k, is denoted as covariance matrix, M is the number of unimodal Gaussian densities, T is the total number of frames for k=1, …, M and t=1, …, T.

(c) The first supervector is represented as the supervector of counts and was generated by merging the soft-counts of the GMM. Mathematically, for the GMM-UBM λ_{UBM} and a feature vector x_t, the role of mixture k for the observation frame x_t, at time t, is defined as, which is given by (Daniel Garcia-Romero, 2012):

$$\gamma_{tk} = \frac{w_k \, \mathcal{N}(x_t; m_k, \Sigma_k)}{\sum_{j=1}^{M} w_j \, \mathcal{N}(x_t; m_j, \Sigma_j)}. \quad \textbf{(4.10)}$$

Afterward, the algorithm determined soft-count for mixture *k* via adding the roles of all frames (Daniel Garcia-Romero, 2012):

$$N_k = \sum_{t=1}^{T} \gamma_{tk}. \quad \textbf{(4.11)}$$

As a result, the supervector of counts was computed as $N = [N_1, N_2 \ldots N_K]^T$.

The second supervector known as the supervector of means, was computed for each mixture component as the weighted average of the observed data, whereby the weights were computed with respect to the roles of the mixture for each frame (Daniel Garcia-Romero, 2012):

$$\mu_k = \frac{1}{N_k} \sum_{t=1}^{T} \gamma_{tk} x_t. \quad \textbf{(4.12)}$$

(d) Finally, the algorithm determined the supervector through merging the means for each mixture component as: $\mu = [\mu_1^T \mu_2^T \ldots \mu_K^T]^T$.

4.1.4 The Feature Extraction Using HOSA Techniques

The second objective of this study is to optimize bicoherence features to allow the automatic source mobile device identification using call recording. Hence, in Chapter 3, the study discussed the methodologies for modeling the mobile devices as nonlinear systems. In addition, Chapter 3 entitled the motivation as well as the special concepts for HOSA techniques to identify such a mobile device response function. Therefore, bicoherence is introduced as an effective function to identify and characterize nonlinearities in call recording signal through phase relations of their harmonic components. Based on this fact, it has been assumed that each mobile device leaves characteristic artifacts in the resulting recording that can be examined using bicoherence magnitude and phase spectra. As a result, both components of the bicoherence are required for the aim of mobile device identification. However, additional normalization

technique is required to reduce memory and the computational overhead involved in utilizing the full two-dimensional bicoherence magnitude and phase of call recording signal.

It could be observed from Figure 3.8 (b) that the bispectrum of signal consists of 12 regions of symmetry. Because bicoherence is only the normalized bispectrum, examination of the bicoherence in only one region ($\omega_1 > 0$, $\omega_2 > 0$) could give sufficient information while reducing the dimensionality of the spectrum. The other regions of the (ω_1, ω_2) plane are redundant. In addition, ZMs are the candidates from image moments literature research to capture the mobile device dependent magnitude and phase artifacts for the selected region. ZMs were first developed by Teague in the early 1980s based on Zernike polynomial functions (Teague, 1980). Zernike polynomials are a set of polynomials that are orthogonal on a unit disk, which named after Zernike (1934) the inventor of phase contrast microscopy. ZMs are known as the effective descriptor in the field of pattern recognition (Farokhi et al., 2015), image retrieval (Wang, Yu, et al., 2011), image reconstruction (Starck & Hilton, 2008), vehicle identification (Nagarajan & Devendran, 2012; Zeng et al., 2014) , audio watermarking (Chen & Xiao, 2013; Wang, Ma, et al., 2011), audio information retrieval (Li et al., 2013). ZMs are statistical measures designed to remain constant after some transformations, such as object rotation, scaling, and translation. These motivations introduced ZMs as a good candidate for robust and reliable pattern recognition applications. In addition, ZMs are superior in compare with regular moments such as scale-invariant Hu moments (Hu, 1962). Hu moments derive a set of moment invariants, which are invariance toward the position, size, and orientation. On the contrary, ZMs provide simple rotation invariance, higher accuracy and less information redundancy due to utilizing orthogonal basis sets. As a result, the ZMs of the bicoherence (ZMBic) magnitude and phase spectrum are used as intrinsic mobile device fingerprints for source mobile device identification.

4.1.4.1 Bicoherence

In practice, the bicoherence of a signal has to be estimated from a finite set of the measurements. Bicoherence is the normalized bispectrum of the signal, whereby there are two broad nonparametric approaches for computing bispectrom: (a) the indirect method, based on estimating the cumulant functions and then taking the Fourier transform; and (b) the direct method, based on a segment averaging approach. Because the proposed framework in this study utilized the direct method, the description of the indirect method is out of the scope of this study. The method is shown in the control flow diagram in Figure 4.1 and includes the following steps, as given by (Choudhury et al., 2008):

(a) The shortened near-silent audio frame of length one second resulting from preprocessing stage is denoted by $x(k)$, whereby the data was broken into K segments of length M with no overlap for $k = 0, 1, \ldots, N-1$, and $i = 0, 1, \ldots, K-1$. Hence, the i^{th} segment of $x(k)$ is denoted by $x_i(k)$ for $k = 0, 1, \ldots, M-1$. For this framework, the algorithm utilized segments of length 128 samples.

(b) The mean of the i^{th} segment was determined as

$$\mu_i = \frac{1}{M}\sum_{k=0}^{M-1} x_i(k) \qquad \textbf{(4.13)}$$

Then, the mean was subtracted from each element in the segment, as given by

$$x_i'(k) = x_i(k) - \mu_i \qquad \textbf{(4.14)}$$

(c) The bicoherence estimation algorithm multiplied the zero-mean-centered segment of the data x_i' by a suitable data window $w(k)$ to control over the spectral leakage. Here, the framework utilized hamming window of length 128 samples. This is because the window length should be selected equal to the number of samples in each segment.

$$x_i''(k) = w(k)x_i'(k) \qquad \textbf{(4.15)}$$

(d) Then, the algorithm estimated the DFT, $X_i(f)$ with FFT length (*nfft*) of 256 samples for each segment as

$$X_i(f) = \sum_{k=0}^{M-1} x_i''(k) e^{-j2\pi kf/M} \quad \textbf{(4.16)}$$

, where f is the discrete Fourier frequency. Based on the computed DFT, the algorithm estimated the raw spectral estimates of $P_i(f)$ and the bispectral estimates of $B_i(f_1, f_2)$ as follows

$$\hat{P}_i(f) = X_i(f) X^{*}{}_i(f) \quad \textbf{(4.17)}$$

$$\hat{B}_i(f_1, f_2) = X_i(f_1) X_i(f_2) X^{*}{}_i(f_1 + f_2) \quad \textbf{(4.18)}$$

(e) Afterward, the resulting raw estimates from all K segments were averaged to obtain the bispectrum estimates given by

$$\hat{P}(f) = \frac{1}{K} \sum_{i=0}^{K-1} \hat{P}_i(f) \quad \textbf{(4.19)}$$

$$\hat{B}(f_1, f_2) = \frac{1}{K} \sum_{i=0}^{K-1} \hat{B}_i(f_1, f_2) \quad \textbf{(4.20)}$$

(f) In the end, the bicoherence magnitude and phase matrix of size [*nfft* × *nfft*] were estimated recalling the Eq. 3.41 in Chapter 3, for the estimated bispectrum in Eq. 4.20,

$$Bic^2(f_1, f_2) \triangleq \frac{|B(f_1, f_2)|^2}{E[|X(f_1)X(f_2)|^2]\ E[|X(f_1 + f_2)|^2]} \quad \textbf{(4.21)}$$

4.1.4.2 Zernike Moments

The feature extraction approach independently computes the ZMs of the bicoherence magnitude and phase image using Zernike polynomials. The algorithm fits orthonormal 30-degree hexagonal Zernike polynomials to the bicoherence surface data provided. The 30-degree hexagonal polynomial proposed by Mahajan and Dai (2006) is the most fitting candidate because the bicoherence is typically rotated octagonal in shape, as shown in Figure 3.8 (b). The algorithm employs Zernike circle polynomials as the basis functions for the orthogonalization process, in a way that the hexagonal polynomial is a linear combination of the circular polynomial. In addition, this relationship allows obtaining the Zernike coefficients in terms of the orthonormal coefficients, whereby the hexagonal polynomials were determined up to the eighth order.

Let Z_j (ρ , θ) be Zernike circle polynomials, these polynomials form a complete orthogonal set over the interior of the unit circle $0 \le \rho \le 1$, and are given as (Mahajan & Dai, 2007):

$$\begin{cases} Z_{even\ j}(\rho,\theta) = \sqrt{2(n+1)}R_n^m(\rho)\cos m\theta, & m \neq 0, \\ Z_{odd\ j}(\rho,\theta) = \sqrt{2(n+1)}R_n^m(\rho)\sin m\theta, & m \neq 0, \\ Z_j(\rho,\theta) = \sqrt{n+1}R_n^0(\rho), & m = 0 \end{cases} \quad \textbf{(4.22)}$$

Where $\rho = x^2 + y^2$, and $\theta = \tan^{-1} y/x$ are the polar coordinates and $R_n^m(\rho)$ are the radial polynomials defined as (Mahajan & Dai, 2007),

$$R_n^m(\rho) = \sum_{s=0}^{(n-m)/2} (-1)^s \frac{(n-s)!}{s!(\frac{n+m}{2}-s)!(\frac{n-m}{2}-s)!} \rho^{(n-2s)} \quad \textbf{(4.23)}$$

The index *n* denotes the order of the polynomial, *m* represents the azimuthal frequency and *j* is a polynomial-ordering number and is a function of both *n* and *m*. In Cartesian coordinates (*x* , *y*); the aberration function for a hexagonal surface is represented in terms of polynomials $H_j(x , y)$, which are orthonormal over the surface (Mahajan & Dai, 2007):

$$W(x,y) = \sum_j a_j H_j(x,y), \quad \textbf{(4.24)}$$

where a_j is the aberration coefficient of the polynomial $H_j(x , y)$. The orthonormality of the polynomials is given by

$$A^{-1} \int_{hexagon} H_j(x,y) H_{j'}(x,y) dxdy = \delta_{jj''}, \quad \textbf{(4.25)}$$

where A=3√3/2 is the area of a unit hexagon inscribed inside a unit circle. Finally, the relative value of the coefficients of the circle polynomials whose linear combination yields an orthonormal hexagonal polynomial H_k is defined as (Mahajan & Dai, 2007):

$$c_{j,k} = -\frac{2}{3\sqrt{3}} \int_{hexagon} Z_j H_k dxdy. \quad \textbf{(4.26)}$$

As shown in Figure 4.12, the 30 degree hexagonal region of integration includes a rectangle FBCE and two congruent triangles FAB and CDE with limits of integration

[-√3/2, √3/2; 1, -1], [0, √3/2; -(√3-x), (√3-x)], [-√3/2, 0; -(√3+x), (√3+x)], respectively. The adopted algorithm for computing the ZMs is demonstrated in Figure 4.13.

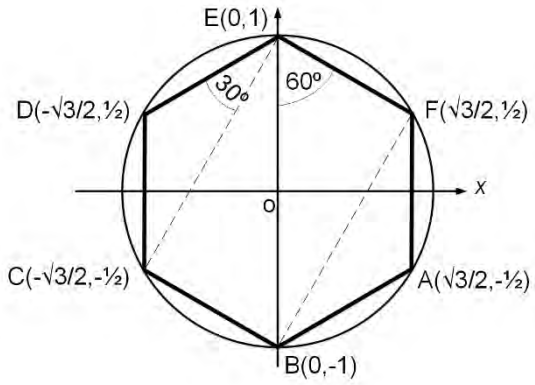


Figure 4.12: Unit Hexagon Rotated 30 Degree Clockwise

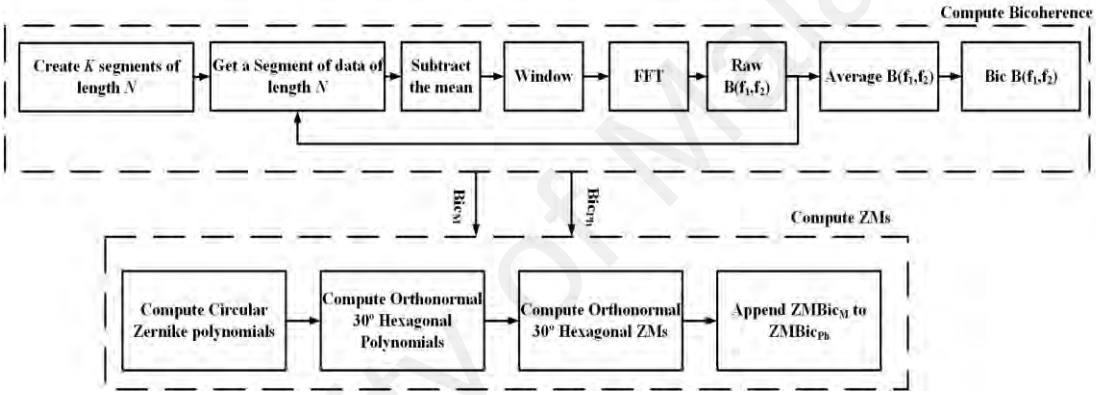


Figure 4.13: Control Flow Diagram of ZMBic Feature Extraction Algorithm

Assuming the two-dimensional bicoherence magnitude and phase output, $Bic_M(x, y)$, $Bic_{Ph}(x, y)$, respectively, for each output the algorithm at first determined the Zernike circle polynomials, Z_j and then utilized these values to compute the orthonormal 30 degree hexagonal polynomials, H_j, for $j=\{1, \ldots, 28\}$ coefficients, as listed in Appendix A.1. The algorithm then computed ZMs of the bicoherence magnitude, $Z_{M\,j}$, and the bicoherence phase, $Z_{Ph\,j}$ as given by (Mahajan & Dai, 2007):

$$Z_{M\,j} = Z_{M\,nm} = \frac{n+1}{\pi} \int_x \int_y Bic_M(x,y) H_{M\,mn}(x,y)\,dx\,dy, \quad x^2 + y^2 \le 1 \qquad \textbf{(4.27)}$$

$$Z_{Ph\,j} = Z_{Ph\,nm} = \frac{n+1}{\pi} \int_x \int_y Bic_{Ph}(x,y) H_{Phnm}(x,y)\,dx\,dy, \quad x^2 + y^2 \le 1 \qquad \textbf{(4.28)}$$

All 56 coefficients from ZMs of the bicoherence magnitude and phase spectrum were computed for all data instances. At last, the algorithm scaled all data instances in the range [0,1] to increase computational time and the classifier performance.

4.1.4.3 Scale-Invariant Hu Moments

Malik and Miller (2012) employed distance similarity measures of the scale-invariant Hu moments of the bicoherence magnitude for automatic microphone identification. Hu moments are invariant under the action of translation, scaling, and rotation. As discussed at the beginning of this section, ZMs have many advantages over regular moments. In order, to prove the superiority of the performance of the ZMs, this work has also computed seven scale-invariant Hu moments of the bicoherence magnitude. For extracting the scale-invariant Hu moments of the bicoherence magnitude, $Bic_M(x, y)$, the algorithm at first computed the geometric moments as given by:

$$m_{mn} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x^n y^m Bic_M(x,y)\,dx\,dy \tag{4.29}$$

Afterward, the central moments were computed as given by:

$$\mu_{mn} = \int_{x}\int_{y} Bic_M(x,y)(x-\bar{x})^n(y-\bar{y})^m\,dx\,dy, \tag{4.30}$$

where, $\bar{x} = \frac{m_{10}}{m_{00}}$, $\bar{y} = \frac{m_{01}}{m_{00}}$. In the next step, the algorithm computed the normalized central moments as follows (Theodoridis & Koutroumbas, 2009):

$$\eta_{nm} = \frac{\mu_{mn}}{\mu_{00}^{\gamma}}, \quad \gamma = \frac{m+n+2}{2}. \tag{4.31}$$

The detail of these moments appears in Appendix A.2.

4.1.5 Feature Analysis and Validation Process

In order to determine the employed source transmitting mobile device of a call recording under investigation, a machine learning algorithm is required to train the classifier with features of call recordings collected corresponding to each individual

mobile device, mobile device model or brand under investigation. Furthermore, the trained machine learning algorithm is able to identify the corresponding source mobile device of a call recording under investigation through detecting the closest match of feature values. As discussed in Chapter 2, in literature, different classifiers such as SVM, Naïve Bayesian, SL, NN and rotation forest are used for machine learning. Meanwhile, the main difference between the related works depends on the feature extraction and analysis approaches. To allow comparison with different state-of-the-art supervised and unsupervised learning techniques, this study, at first entitled analysis of feature-based mobile device identification through closed sets as implemented in data mining tool WEKA (Hall et al., 2009), where all questioned mobile device sources were considered known. Subsequently, the study considers the problem of unknown mobile devices and possible false detection in so-called open sets. The proposed open set algorithm handles the unknown mobile devices via one-against-all SVM classifier and minimizing the training data error associated with it.

4.1.5.1 Selected Supervised Learning Methods

Classification is known as a supervised learning method (Bhatt & Kankanhalli, 2011). However, to determine which approach is the most efficient for a particular problem, systematic methods are required to evaluate how different methods work and to compare these methods with one another. The framework utilized the MATLAB based library for support vector machines (LIBSVM) classifier to classify the data instances with an *RBF* kernel (Chang & Lin, 2011). This is because LIBSVM can perform well whenever applied to pattern recognition approaches. Multiclass classification problems can be implemented through different classification techniques. In Section 5.4.2 and 5.5.2, for classifier benchmarking, the study analyzed the performance of the optimized entropy-MFCC and ZMBic feature instances based on the majority of classifiers implemented in data mining

tool WEKA (Hall et al., 2009). The specifications on these classifiers are given in Table 2.4 and Table 2.5.

4.1.5.2 Selected Unsupervised Learning Methods

The unsupervised learning method is the general term for clustering algorithms. In this case, no class exists for the prediction, and the data instances are divided into groups based on the relationships between features, such as distance-based similarity measures, as well as hierarchical and incremental relationships (Xu & Wunsch II, 2005). The study evaluated the reliability of the source mobile device identification by using the unsupervised learning techniques such as density-based spatial clustering of applications with noise (DBSCAN) (Hao et al., 2011) and EM-based clustering (Abbas, 2008).

4.1.5.3 Open Set SVM Classifier

The majority of similar approaches in literature research were investigated in a closed set scenario, with the assumption that the audio recording under investigation was processed by one of the *n* known mobile devices available during the training. However, sometimes it is impossible to be confident that an audio recording is processed by one of the mobile devices under investigation. Further, it is necessary to model the source mobile device identification problem in an open set scenario, whereby in a real-world environment, there is access to a limited set of suspect mobile devices. Hence, an open set scenario represents a real-world environment much better in compare with the closed set one. In Costa et al. (2014), the authors utilized the decision boundary carving algorithm for open set source camera attribution. This scenario adopted a classification model with respect to the selected available classes while taking the unknown variables under investigation. The proposed open set source mobile device identification framework in this study matches a call recording to its specific source by using acoustic features, whereby this work has access to a limited set of mobile devices for training, and a call recording can be processed and transmitted from any mobile devices, including

mobile devices to which this work never had access. In general, the algorithm handles the unknown mobile devices via SVM classifier though minimizing the training data error associated with it. The open set classification algorithm employed in this study consists of the following steps:

(a) The algorithm at first built a multi-class SVM classifier using a LIBSVM library with RBF kernel from the training set of instances considering all class labels. Mathematically, let the training data be (x_i , y_i) for $i = 1 \ldots N_{Tr}$, with $x_i \in \Re^d$ and $y_i \in l = \{1, \ldots, N_c\}$, ($N_{Tr}$ represents the number of data instances, and N_c represents the number of classes). Let x be the training data matrix in which the n^{th} row of x corresponds to the row vector x_i^{Tr}.

(b) The classifier at first considers the test data instances (x_i , y_i) for $i = 1 \ldots N_{Te}$, with $x_i \in \Re^d$ and $y_i = 1$, then the classifier predicts the label for the test data instances and determines the prediction accuracy, acc_1. Then, set acc_1 to maximum prediction accuracy $\mathrm{acc}_{\max}^1$. The algorithm continues to compute SVM predict by replacing the $y_i = 2$ with $y_i = 1$, and subsequently, compares the prediction accuracy acc_2 against $acc_{\max}^1$. Next, if $acc_2 > acc_{\max}^1$ the maximum prediction accuracy is updated to $acc_{\max}^2$, otherwise it remains as $acc_{\max}^1$.

(c) The algorithm continues to compute the prediction accuracy for all possible class labels and each time compares it against the maximum prediction accuracy. Eventually, if $acc_{\max}^l < 100/N_c$ the algorithm sets the class label as zero and recognizes the mobile device model as unknown, otherwise the algorithm recognizes the class label that achieves the best prediction accuracy as the mobile device model.

4.1.6 Detection Performance Metrics

The closed set experiments in Chapter 5, based on classifiers in Section 4.1.5.1 requires the following metrics and parameters for evaluation:

(a) *Identification accuracy* (*ACC*)=*Correct classified instances/All classified instances*

(b) *Kappa Statistics* (*KS*): measures the agreement between the predicted and observed categorization of the dataset, it is useful to evaluate classifiers among themselves and it is more robust than *ACC*.

(c) *ROC Curve*: determines the cost of misclassification error for each individual class by plotting the true positive rate on the vertical axis against the TNR on the horizontal axis.

The performance of the numeric predictions is measured based on the testing data as in (Eq. 4.32–3.35). For all measures, p_i is the numeric value of prediction and a_i is the actual value for the i^{th} instance, where $i = 1,2,3,\cdots N$ and N is the total number of test instances.

(d) *RMSE*: computes the square root to determine the same dimensions as the predicted value itself.

$$\sqrt{\frac{(p_1 - a_1)^2 + \cdots + (p_N - a_N)^2}{N}} \quad \textbf{(4.32)}$$

(e) *Mean absolute error (MAE)*: treats all sizes of error evenly according to their magnitude.

$$\frac{|p_1 - a_1| + \cdots + |p_N - a_N|}{N} \quad \textbf{(4.33)}$$

(f) *Root relative squared error (RRSE)*: computes the total squared error and normalizes it through dividing by the total squared error based on a simple predictor. The predictor is the average of the actual values from the training data that is represented by $\overline{a}$.

$$\sqrt{\frac{(p_1 - a_1)^2 + \cdots + (p_N - a_N)^2}{(a_1 - \bar{a})^2 + \cdots + (a_N - \bar{a})^2}} \quad \textbf{(4.34)}$$

(g) *Relative absolute error (RAE)*: considers the total absolute error, with the same normalization approach, as in (Eq. 4.35).

$$\frac{|p_1 - a_1| + \cdots + |p_N - a_N|}{|a_1 - \bar{a}| + \cdots + |a_N - \bar{a}|} \quad \textbf{(4.35)}$$

Selected metrics and parameters appear in abbreviated form in Chapter 5.

Although obtaining the metrics and parameters for benchmarking unsupervised learning techniques is sometimes difficult, the performance of the clustering algorithms discussed in Section 4.1.5.2 was measured by using the following metrics, as detailed in (Witten et al., 2011d).

(a) *Incorrectly clustered instances*: number of instances assigned incorrectly to the clusters

(b) *Unclustered instances*: number of instances that are not assigned to any cluster

(c) *Log likelihood (LL)*: measures the goodness of fit; a larger value indicates that model fits the data better.

(d) *MDL metric*: determines the MDL score for k clusters and N instances as detailed in (Hao et al., 2011),

$$MDLScore = -LL + \tfrac{k}{2} \log N. \quad \textbf{(4.36)}$$

The MDL score is smaller for strong clustering techniques. Nevertheless, this value increases for less strong clustering techniques. The aforementioned parameters appear in Chapter 5 in abbreviated form.

4.2 General Tools

To perform the experiments in Chapter 5, this study utilized both MATLAB and WEKA software for their signal processing, data mining, and machine learning applications. The reasons for utilizing MATLAB were its interactive environment for

numerical computation, visualization, and programming. Its high-level language, tools, and built-in math functions allow experiencing different approaches. The reasons for utilizing WEKA were its pre-existing collections of machine learning algorithms and data preprocessing tools, openness and being free to use. In order to easily and efficiently apply SVM classifier to the audio source mobile device identification framework, this study employed open-source LIBSVM library. Furthermore, various open-source skype recording applications were used including MP3, PAMELA, VodBurner, and Callnote. For GSM call recording, the free windows phone GSM call recorder were utilized. In addition, for playing, converting and editing the audio files VLC media player and Audacity were utilized. The reasons for using these applications were their openness, public availability and being free to use. The details of the applications are discussed as follows:

(a) *MATLAB*: Cleve Moler started developing MATLAB in the late 1970s. In 1984, Cleve Moler, Steve Bangert, and Jack Little continued its development by rewriting MATLAB in C and finding MathWorks. MATLAB is a high-level language for numerical computation, visualization, and application development and interactive environment for iterative exploration, design, and problem-solving. It contains mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration, and solving ordinary differential equations. MATLAB utilizes built-in graphics for visualizing data and tools for creating custom plots. A key advantage is that MATLAB employs functions for integrating MATLAB based algorithms with external applications and languages such as C, C++, Java and .NET. Hence, for this study MATLAB provides a simple interface to LIBSVM that was originally written in C. This study applied MATLAB R2015a for developing the final prototype in this study because MATLAB provides tools for building applications with custom graphical interfaces.

(b) *WEKA*: Developed at the University of Waikato in New Zealand, which stands for *Waikato Environment for Knowledge Analysis.* The tool is written in Java language and distributed under the terms of the GNU General Public License. It offers a uniform interface to various learning algorithms, in addition to methods for pre- and postprocessing and for evaluating the result of learning algorithms on any type of dataset. The workbench consists of main data mining algorithms: regression, classification, clustering, association rule mining, and attribute selection. It also visualizes the dataset and provides data preprocessing tools. WEKA applies a selected learning method to a dataset and analyzes its output to learn more about the data. Using a different approach, it generates predictions on new instances using learned models. This study utilized WEKA version 3.6.10 to perform performance evaluation (Phase I, III and IV) during source mobile device identification in a closed set.

(c) *LIBSVM*: This package has been actively developed since 2000 to help users to easily apply SVM to their applications. LIBSVM is an efficient implementation of SVM algorithm written in C language. This study employed LIBSVM Version 3.20 for preliminary test evaluation in Phase I and II as well as open set evaluation in Phase V.

(d) *MP3 Skype Call Recorder*: The software is free with minimal limits attached for private, non-commercial use. It performs automatic or manual recording options. The stored records are in the compact '.mp3' format. This study utilized MP3 Skype Recorder Version 3.1 during collection of DS1 and DS2.

(e) *PAMELA FOR SKYPE*: The freeware Pamela basic is free, reliable and easy to use Skype audio and video recording software for Windows systems. Although the basic Version 4.9.0.56 only allows for 15 minutes per audio call recording, it is sufficient for the call recording sessions of three minutes during collection of DS3. An

advantage is that PAMELA allows saving the call recordings in the raw and uncompressed '.wav' audio format.

(f) *VodBurner*: The Mac version of the software (Version 1.1.0.203 (Mac OS X)) was available until 11th August 2014, when Microsoft/Skype had withdrawn support SkypeKit-the technology was used to build Vodburner for Mac. It was able to capture video and audio to '.mp4', '.mov', '.m4a' and '.wav' for free with absolutely no time. VodBurner for Mac was easy to use, it utilized built-in Skype and there was no need to install Skype on Mac. It allowed to log into Skype through the VodBurner interface, and it was possible to make and receive calls directly from within VodBurner. This study utilized Mac version of the VodBurner for the iMac stationary and collected audio call recordings in WAV format for DS3. After stopping its operation, the software was replaced with Callnote.

(g) *Callnote*: Developed by Kanda Software, Callnote is a free application for both windows and Mac systems that easily records audio Skype calls and saves the recordings to the Evernote or Dropbox account. This study utilized Mac's version of the Callnote (Version 3.1.21 (Mac OS X)) for the iMac stationary and collected audio call recordings received from Sony Xperia mobile devices. Unfortunately, Callnote saves the call recordings in MP4 format and later should be converted to WAV format.

(h) *Windows Phone GSM Call Recorder*: The application is free to use and designed for windows phone to allow recording of GSM calls. It saves GSM calls in the stereo channel MP4 format and allows to upload files to OneDrive through sharing the files. This study employed this application for Nokia Lumia 710 stationary while collecting GSM call recordings for DS3.

(i) *VLC Media Player*: It is a free and open-source media player, encoder and streamer made by the volunteers of the VideoLan community. The software uses its internal codecs, works on essentially every popular platform and reads the majority of file

formats. This study utilized VLC media player (Version 2.2.1) for playing back the audio files and for converting the '.mp3' and '.mp4' files to uncompressed '.wav' format.

(j) *Audacity*: The software is a free program written by a worldwide team of volunteer developers. The projects are hosted on Google Code and SourceForge. It is available for Windows, Mac, and GNU/Linux (and other Unix-like systems). Audacity (Version 2.0.5) was employed for editing the call recordings.

4.3 Design Assumptions and Rationale

In order to expedite and perform the experiments in this study accurately, along with the particular model and tactics planned in the proposed framework, it is necessary to make some adjustments and assumptions. Precisely, given the lack of specific information on internal signal processing circuits and electrical components utilized by different mobile device manufacturers, some assumptions had to be made on modeling the mobile device response function in the proposed control system model for the communication and call recording pipeline. Using the subsystems given for the control system model discussed in Section 3.1.3, the experiments in this chapter were handled based upon a set of assumptions on the influence factors and the mathematical model of the mobile device response function. The assumptions are required to justify the proposed spectral analysis techniques in determining the mobile device intrinsic fingerprints, so the mathematical modeling does not necessarily reflect the actual response of the mobile device communication signal processing pipeline. Therefore, the study computed entropy-MFCC and ZMBic feature sets in order to estimate mobile device response function based on the hypothesis that the mobile device response function induces nonlinearity on the non-stationary random input signal. To investigate this hypothesis, the Gaussianity, and linearity of the call recording signal were tested in Appendix D1 using the hypothesis testing algorithm by Hinich (1982), as discussed in Section 3.2.3.5.

Using a different approach, the Gaussianity and nonlinearity of the call recording signals were tested in Appendix D2 through the proposed hypothesis testing methods by Choudhury et al. (2006).

4.4 Summary

This chapter has focused on the conceptual framework for optimizing acoustic features, in order to develop a robust framework that allows the automatic source mobile device identification. Its description has included an introduction of the main audio mining components, strategies, frameworks and the rationale behind their implementation, as well as their operational characteristics. In summary, this chapter highlighted the key points of this study and provided the detail within the framework, in addition to the discussion on how they can be combined as one significant framework. It is essential to understand the interrelationship between those approaches and the model in compiling the overall process in optimizing acoustic features in an attempt to get a useful result from the source mobile device identification process.

Having established the proposed framework using multiple approaches and models, the next chapter presents several evaluations of the framework, which is followed by a detailed discussion of them. It is important to understand that the results provide a verification of the usefulness and suitability of the framework in facilitating the automatic mode in the mobile device identification process.

CHAPTER 5: EXPERIMENTAL RESULTS

The novelty of this study is to develop a source mobile device identification framework based on recorded calls transmitted from different mobile devices. The framework optimizes acoustic features using spectral analysis techniques in order to capture the signal variations due to mobile device frequency response on call recording signal. The ultimate aim is to handle closed and open set audio source mobile device identification. Hence, it is important to perform evaluation study in order to highlight the feasibility and suitability of the framework. Having proposed the multi-strategy source mobile device identification framework, it is necessary to design a systematic evaluation phase in order to achieve verification of its feasibility and fitness, specifically for the second part of the framework, which is feature optimization.

This chapter entitles five evaluation phases with respect to the proposed framework, which aims to evaluate it in terms of its effectiveness and performances in relation to the models and strategies selected. This evaluation study needs to investigate the effectiveness and performance of the proposed framework in order to satisfy its feasibility and fitness, in particular, the ability of the framework to facilitate the open set scenario. Meanwhile, the open set problem identifies the source of unknown mobile devices and reduces possible false detection.

The first phase investigates the reliability of the entropy-MFCCs through visualization and preliminary tests for individual mobile device, model and brand identification. To make evaluation and comparisons with the results from other studies, this phase analyses the effectiveness of the selected state-of-the-art features for this framework. With the first phase results, the second phase investigates the statistical properties of the optimized entropy-MFCCs and ZMBic feature sets for intra- and inter-mobile device model identification. The third and fourth phase extend the closed set evaluation by investigating

different strategies in cepstrum- and bispectrum-based feature extraction for source model and individual mobile device identification. In particular, both phases apply the critical test in regard to different influence factors for evaluation, as opposed to the preliminary test evaluation applied in the first phase. The phases evaluate the relationship between different feature extraction and classification algorithms for benchmarking. The fifth phase evaluates the fitness of using an open set classifier to the proposed source mobile device identification framework. This phase investigates the performance of the proposed framework by measuring the final accuracy and false detection error rates. At last, the chapter concludes with a summary.

5.1 General Description

There are three stages of the evaluation study in this chapter, where each stage has its specific objectives with different results, discussion, and conclusion. Despite this, some requirements are common in the experimental procedures.

For performance, evaluation (Phase I) call recording setups corresponding to sets of different mobile devices, including mobile devices of the same brand and model are used to establish:

(a) Whether the call recording-based source mobile device identification is actually possible with the introduced approach?

For performance evaluation (Phase II-IV) practical investigations are performed within this study on:

(a) Which aspect of the type and number of feature vectors in training has an influence on the detection performance?
(b) Which classifiers (from a pre-existing collection provided by WEKA) are suitable for implementing the source mobile device identification?

(c) Which features from the audio acquisition device identification literature research are suitable for source mobile device identification based on the recorded call?

(d) How is classification using content selection as well as content dependent and independent training and testing influences the detection performance in source mobile device identification?

(e) What are the influences of:

- The recording environment
- The recording stationary
- The transmission channel

(f) Is it possible to obtain robustness against common audio post-processing operations (normalization, MP3 conversion, and denoising)?

For evaluation performance (Phase V) practical investigations are performed within this study on:

(a) Is it actually possible to identify the source mobile device of the call recording that processed by the mobile device other than the ones utilized during the training with the introduced open set approach?

5.2 Performance Evaluation- Phase I: Preliminary Test

The first step of the evaluation study investigates the use of entropy-MFCCs as a mobile device intrinsic fingerprints for source mobile device identification and aims to achieve two objectives:

(a) To propose entropy of Mel-cepstrum coefficients from near-silent segments as an optimizing method to extract intrinsic mobile device features, where it remains robust to the characteristics of different speakers.

(b) To compare the performance of the proposed features for inter- and intra-mobile device identification.

Hence, two main experiments were conducted based on the application scenario of source mobile device identification corresponding to the call recording samples of DS1 and DS2 as described in Section 4.1.1. The evaluation was represented as a preliminary test because it mainly aims to evaluate the reliability of the method.

The experiments utilized 12 zero order MFCCs, whereby the log energy, and the first and second derivative of MFCC coefficients could also be included in MFCC feature vector. However, preliminary results show fewer contributions of the log energy, as well as the first and second order cepstral coefficients in achieving the identification accuracy rates. Hence, the Mel-cepstrum output consists of one frame per row, and each frame includes 12 coefficients. Finally, the algorithm generates a total of 12 entropy-MFCC features by computing the entropy of the MFCCs. The following sections include the description of the experiments, results, and discussion.

5.2.1 Experiment 1

For DS1, the experiment generated a total of 1000 data instances from the recorded calls from each mobile device. All 12 entropy-MFCC features were computed by using

the generated data instances to obtain the data subset with a length 1000 for each mobile device. The method randomly selected 700 data instances for training and utilized the remaining 300 instances to test the data subset. The experiment thus obtained 7000 training and 3000 testing data from the 10 mobile devices. This stage was repeated 10 times, and the average accuracy was computed. Table 5.1 shows the average confusion matrix generated by running 10 experiments using the 10-class SVM classifier. The diagonal values of the matrix represent the respective classification accuracies of the 10 mobile devices, whereas the non-diagonal values indicate the misclassification among the mobile devices. A high average classification accuracy of 99.72% was achieved for all mobile devices. The percentage of misclassification among the mobile devices was negligible (less than 0.27%). The mobile devices of the same model (Galaxy Note 10.1-A, B and Galaxy Note II-A, B) obtained high average accuracy rates of 99.74% and 99.76%, respectively.

Table 5.1: Confusion Matrix of Intra-Mobile Device Identification for Entropy-MFCCs Based on SVM Classifier

Total average accuracy rate 99.72%		***Predicted* (%)**									
		GNA	***GNB***	***GN***	***GNIIA***	***GNIIB***	***GT***	***iPadA***	***iPadB***	***Asus***	***HTC***
Actual	***GNA***	99.7	0.1	*	*	*	*	0.1	*	*	*
	GNB	*	99.8	*	*	*	*	*	*	0.1	*
	GN	*	*	99.8	*	*	*	*	*	*	*
	GNIIA	*	*	0.1	99.8	*	*	*	*	*	*
	GNIIB	*	*	0.2	*	99.7	*	0.1	*	*	*
	GT	*	*	*	*	*	99.8	*	*	*	*
	iPadA	*	*	0.1	*	*	0.17	99.6	*	*	*
	iPadB	*	*	*	*	*	*	*	99.7	*	0.23
	Asus	*	0.27	*	*	*	*	*	0.1	99.6	*
	HTC	*	*	0.17	*	*	*	*	0.13	*	99.6
The cell marked with an asterisk indicates a value of less than 0.1%.											

In an alternative approach, Figure 5.1 visualizes the classification results by using the Euclidean distance similarity method. Each color represents the class label of the dataset associated with each mobile device. The unfilled markers represent data instance from the training data subset, and the filled markers represent data instance from the testing data subset. It is evident that, the Euclidean distance method clusters both the training and

testing data subsets into 10 groups. This observation is consistent with the results obtained by the ten-class SVM classifier. It could be inferred therefore that the proposed entropy-MFCC features are the effective feature set in the source mobile device identification using recorded VoIP calls.

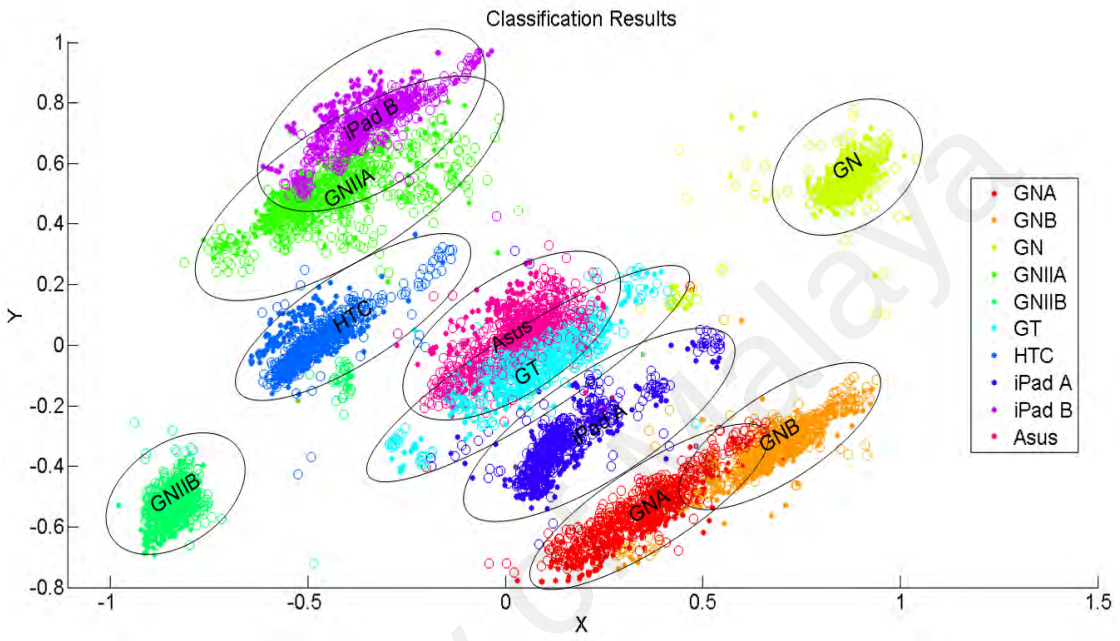


Figure 5.1: Clustering of Training (Unfilled Markers) and Testing (Filled Markers) Data Subsets by Using the Euclidean Distance Method

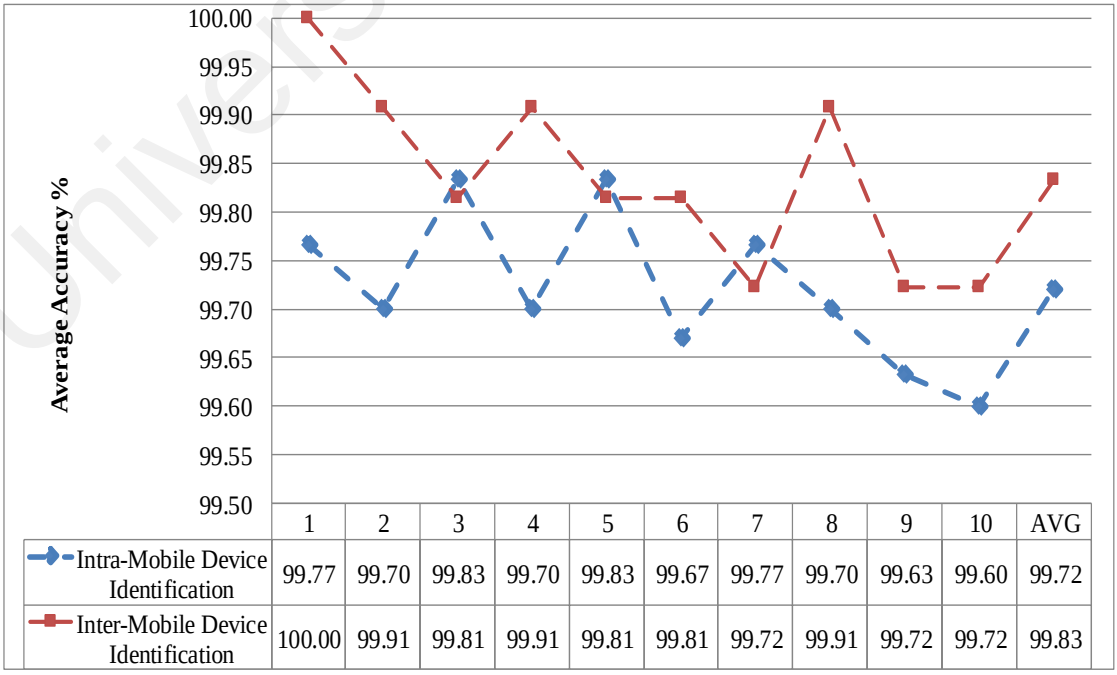
At this stage the experiment utilized DS1 to verify the feasibility of the proposed method for identifying the source brand of the mobile devices that their VoIP calls were transmitted to the stationary, whereby the stationary recorded only the near-silent conversations. As listed in Appendix B2, this dataset consists of 10 mobile devices in four different brands. Hence, in order to obtain consistency among the number of training and testing data subsets, a total 630 and 270 data instances were selected as training and testing data subsets for each class respectively, resulting a total of 900 data instances for each class.

In this case, the data instances corresponding to mobile devices of the same brand were assigned to the same label. The 4-class SVM classifier was built using the training data subset and the prediction vector was generated using the testing data subset. Table 5.2

shows the average confusion matrix resulted from running 10 experiments using a 4-class SVM classifier. The high average identification accuracy of 99.83% was achieved for inter-mobile device identification, which was approximately 1% higher than the result for intra-mobile device identification. Moreover, Figure 5.2 compares the variation of the average accuracy for intra and inter-mobile device identification for each trial experiment. In the majority of cases, the inter-mobile device identification revealed a considerably higher average accuracy than the intra-mobile device identification. This is because, the classifier consists of less number of classes, and the signal variations corresponding to mobile devices of the different brand are more distinctive.

Table 5.2: Confusion Matrix of Inter-Mobile Device Identification for Entropy-MFCCs Based On SVM Classifier

Total average accuracy rate 99.83%		***Predicted* (%)**			
		Galaxy	***Apple***	***Asus***	***HTC***
***Actual* (%)**	***Galaxy***	99.82	*	*	*
	Apple	0.19	99.78	*	*
	Asus	*	0.15	99.82	*
	HTC	*	*	*	99.93
Note: the cell marked with an asterisk symbol has value of less than 0.1%					



	1	2	3	4	5	6	7	8	9	10	AVG
Intra-Mobile Device Identification	99.77	99.70	99.83	99.70	99.83	99.67	99.77	99.70	99.63	99.60	99.72
Inter-Mobile Device Identification	100.00	99.91	99.81	99.91	99.81	99.81	99.72	99.91	99.72	99.72	99.83

Figure 5.2: Average Accuracy Rates for Inter- and Intra-Mobile Device Identification against Increase of the Experimental Trials

5.2.2 Experiment 2

The experiments evaluate the feasibility of the source mobile device identification method through classification accuracy, robustness, and computational efficiency. This process justifies the choices made to handle datasets, features, training and testing instances, classification, and evaluation technique. The first experiment focused on the data preparation approach. The second experiment employed the most common combinations of statistical moments of MFCCs, such as mean, standard deviation, variance, skewness, and kurtosis (Beigi, 2011d). By modifying the feature extraction algorithm in Figure 4.9, "Mean-MFCC," "Stdev-MFCC," "Var-MFCC," "Skew-MFCC," and "Kurt-MFCC" were employed. These feature sets are popular among works on musical instrument classification (Senan et al., 2011), as well as speaker verification (Alam et al., 2011; Kinnunen et al., 2012) and identification (Molla & Hirose, 2004), and were thus adopted for comparison with entropy-MFCC features. The remaining experiments determined the classification performance for individual mobile device models and brands, respectively.

5.2.2.1 Experiment on data preparation approaches

This experiment used the data instances prepared with the enhancement technique as described in Section 4.1.2.1 against data instances prepared from the original audio signals to justify the choices made against its alternatives. All 21 mobile devices (as detailed in Appendix B3) were employed with 1000 data instances from each to evaluate the performance of the five classification algorithms through 10-fold cross-validation; the results are listed in Table 5.3. The clean data instances obtained the best performance with 99.91% identification accuracy and RRSE of 4.37% by using the naïve Bayesian classifier. The result shows that the environmental noise distortions in the original data instances slightly reduce classification accuracy to 99.80% with respect to the clean data instances. Although the effect of de-noising on classification accuracy is minimal, it

increases the computational time, particularly for the linear logistic regression model. This finding also suggests the robustness of the entropy-MFCC features against environmental noise distortions.

Table 5.3: Performance Comparison of Entropy-MFCC Features from Enhanced and Original Audio Signals

	Entropy-MFCC					***Noisy Entropy-MFCC***				
Classifiers	***MAE***	***RMSE***	***RAE***	***RRSE***	***ACC***	***MAE***	***RMSE***	***RAE***	***RRSE***	***ACC***
SVM	0.0003	0.0186	0.38%	8.72%	99.6%	0.0005	0.0221	0.54%	10.39%	99.49%
Neural Network	0.0008	0.0179	0.86%	8.42%	99.6%	0.0008	0.019	0.84%	8.91%	99.55%
Naïve Bayesian	0.0001	0.0093	0.10%	4.37%	99.9%	0.0002	0.0139	0.22%	6.54%	99.80%
Rotation Forest	0.0009	0.0154	1.03%	7.25%	99.8%	0.0011	0.0168	1.17%	7.91%	99.73%
Linear Logistic Regression	0.0005	0.0223	0.56%	10.47%	99.4%	0.0007	0.0258	0.77%	12.10%	99.28%

5.2.2.2 Experiment on Entropy-MFCC features

This experiment was conducted to indicate the contribution of entropy and MFCCs in identification performance as discussed in Section 3.2.2. To justify the selection of entropy, the experiment compared the performance of entropy-MFCC features for source mobile device identification with other statistical moments of MFCCs, as adopted in (Kinnunen et al., 2012; Senan et al., 2011) and (Molla & Hirose, 2004). As a result, five feature sets of "Mean-MFCC", "Stdev-MFCC," "Var-MFCC," "Skew-MFCC," and "Kurt-MFCC" were computed. The various moments of MFCCs were concatenated to a single feature vector. This feature vector contains 60 features that were reduced to 48 by using best-first search method. The best-first method traverses the feature space to find the best subset by evaluating each one through the SVM classifier. This search method uses Greedy hill climbing with backtracking algorithms (Goldberg, 1989). The experiment evaluates the performance of all feature sets by using five classification and two clustering algorithms via 10-fold cross-validation. In the next step, the experiment eliminated the logarithmic transformation of MFCCs from (Eq. 4.4) to compute the discrete cosine transform of Mel-filterbank energies (DCT of MFBE). The identification

performance based on DCT of MFBE was obtained and compared against entropy-MFCCs. This comparison was to study the effect of frequency domain features with multiplicative components on identification performance.

(a) ***Classifying mobile devices based on entropy-MFCCs and statistical moments of MFCCs***

The experiment determined the classification results for different statistical moments of MFCCs, the combined feature set, and its best-first selected features, as summarized in Table 5.4. The highest accuracy rate was always determined when the feature set was used in rotation forest classifier. This classifier achieved an accuracy of 95.10% for "Mean-MFCC" feature set. Meanwhile, for most classifiers, the highest accuracy rate was obtained with combined feature set. However, feature selection produces a small improvement in classification accuracy. This result was compared against the performance of the entropy-MFCC features as appeared in Table 5.5. The entropy-MFCC feature set always outperforms statistical moments of MFCCs with higher accuracy rates. This outcome agrees with the comparison of ROC curves that were obtained from these feature sets.

Table 5.4: Performance of Statistical Moments of MFCCs

Statistical Moments of MFCCs							
Classifiers	***Mean***	***Stdev***	***VAR***	***Skew***	***Kurt***	***Combined Set***	***Combined with Best-first***
SVM	32.83	28.18	30.08	31.35	21.54	69.02	69.62
Neural Network	39.43	28.75	30.78	31.33	21.99	88.42	73.35
Naïve Bayesian	29.29	26.96	27.0	32.90	18.60	60.12	61.18
Rotation Forest	95.10	40.96	42.79	30.57	17.83	89.48	84.36
Linear Logistic Regression	28.27	26.63	33.48	30.79	22.30	84.45	68.75

Figure 5.3 compares the overall ROC curves of the Rotation Forest classifier among all feature sets and label class. The ROC area for the entropy-MFCC features was close to one, but the value was smaller for other feature sets. This finding indicates that for entropy-MFCC features, the false positive rate is close to zero, and the true positive rate is close to one. Moreover, the ROC area for the "AllSet" features that were produced with

the combined statistical moments of MFCCs was significantly smaller than the entropy-MFCCs. Overall, because in near-silent segments, fewer contents exist to be modeled by MFCCs, their value was not large enough to represent the strong discrimination among mobile devices. Meanwhile, entropy intensifies the value of MFCCs in near-silent segments and increases the classification accuracy.

Table 5.5: Performance Comparison of Entropy-MFCC Features and Entropy-[DCT of MFBE] Based on Model

Entropy-MFCC						***Entropy-DCT of MFBE***				
Classifiers	***MAE***	***RMSE***	***RAE***	***RRSE***	***ACC***	***MAE***	***RMSE***	***RAE***	***RRSE***	***ACC***
SVM	0.0002	0.0158	0.28%	7.42%	99.74%	0.0003	0.0183	0.37%	8.60%	99.65%
Neural Network	0.0006	0.0153	0.72%	7.16%	99.68%	0.0007	0.0164	0.74%	7.68%	99.65%
Naïve Bayesian	0	0.0030	0.01%	1.41%	99.99%	0	0.0037	0.02%	1.73%	99.99%
Rotation Forest	0.0004	0.0181	0.39%	8.50%	99.80%	0.0009	0.0154	0.99%	7.21%	99.84%
Linear Logistic Regression	0.0005	0.0223	0.56%	10.47%	99.63%	0.0006	0.0232	0.62%	10.9%	99.42%

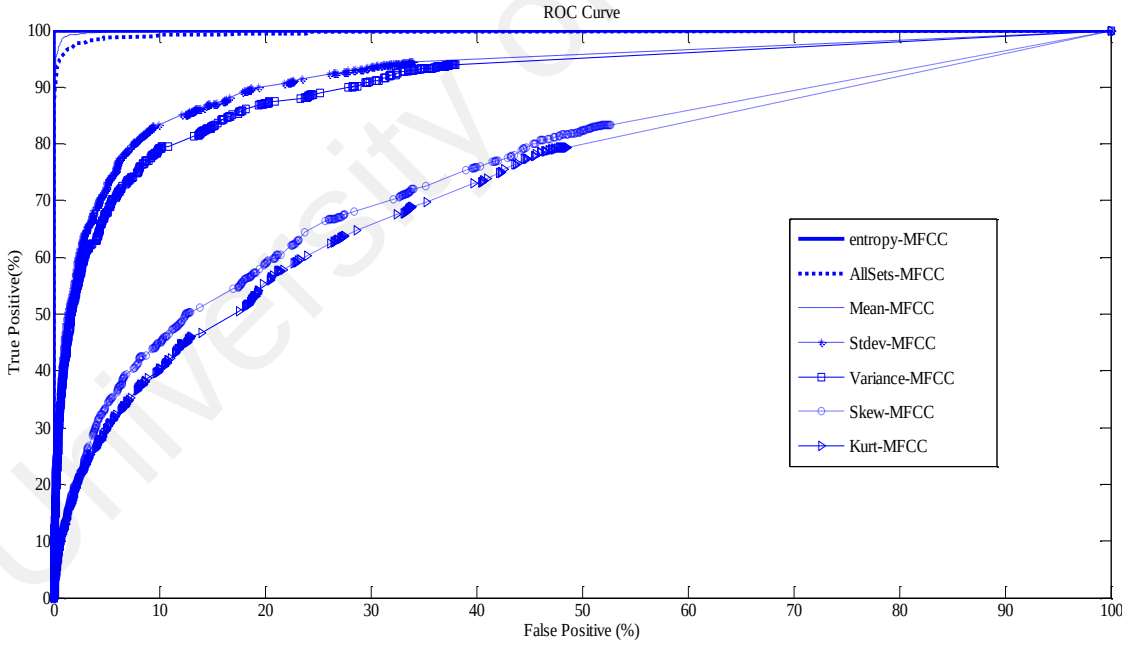


Figure 5.3: Overall ROC Curves of Rotation Forest Classifier Using Different Feature Sets on the Class of Labels

Figure 5.4 illustrates the classifier benchmarking for Entropy-MFCC feature set, where vulnerability is the performance reduction due to replacing "Entropy-MFCC" with "Stdev-MFCC". As can be seen, the Rotation Forest and SVM classifier exhibited the lowest increase in error rates. This observation suggests the robustness of both classifiers

against loss of accuracy rates. Naïve Bayesian classifier generally achieved high identification accuracy at the shortest computation time but with the lowest robustness. Rotation Forest achieved the second-best identification accuracy and the best robustness, but the computation time was considerably slower. Moreover, the performance of the SVM classifier was comparable with the Rotation Forest classifier.

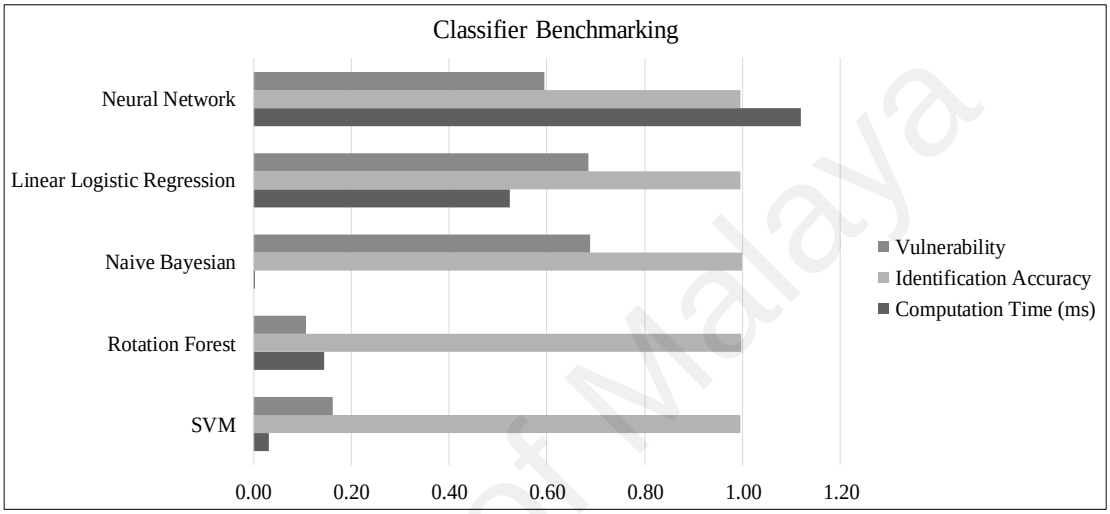


Figure 5.4: Classifier Benchmarking Based on Vulnerability, Identification Accuracy and Computation Time

(b) ***Classifying mobile devices based on entropy-MFCCs and entropy-[DCT of MFBE]***

The experiment determined the performance of the entropy-MFCC feature set against entropy-[DCT of MFBE], as detailed in Table 5.5. The result shows both feature sets performed comparably. This is because entropy intensifies the energy of the silent segments and moreover, by extracting the features from near-silent segments, convolution due to speech segments is eliminated. The result proves the contribution of entropy to the improvement of the performances of both MFCCs and DCT of MFBE features that were proposed in (Hanilçi et al., 2012) for cell-phone identification. In this work, Hanilçi et al. determined that DCT of MFBE features reduces the accuracy rates for identification.

(c) ***Performance in clustering mobile devices based on entropy-MFCCs***

This experiment re-evaluates the performance of all feature sets with probabilistic-based (EM) and nearest-neighbor-based (DBSCAN) algorithms. However, only the

entropy-MFCC feature set can diverge to assessable results. Table 5.6 summarizes the results by using DBSCAN clustering based on the entropy-MFCC feature set with a minimum neighbor distance of $\varepsilon = 0.4$ and minimum cluster size of 200. This algorithm identified 21 clusters with respect to the total mobile devices with only one incorrectly clustered instance. However, 880 instances out of 21000 instances were unclustered. The EM algorithm inserts the number of clusters beforehand and then determines incorrectly clustered instances, LL, and MDL metrics. Thus, 4841 instances out of 21000 instances were incorrectly clustered. Smaller MDL indicates strong clustering techniques. DBSCAN assigned more instances to its correct cluster, which makes it the better choice.

Table 5.6: Clustering Performance Based on Entropy-MFCCs

EM Algorithm			***DBSCAN Algorithm***		
ICI	***LL***	***MDL***	***ICI***	***UCI***	****GC***
4841	24.10	21.28	1	880	21
**All instances=21000. GC=Generated Clusters, ICI: incorrectly classified instances*					

5.2.2.3 Intra-mobile device identification by using SVM

Source mobile device identification in previous experiments performed well with the SVM classifier in terms of identification accuracy, robustness, and computational efficiency. This experiment analyzes the identification accuracy of individual mobile devices based on the entropy-MFCC feature set and SVM classifier. The confusion matrix in Table 5.7 shows the correct and incorrect classified instances in diagonal and non-diagonal cells, respectively. Moreover, the proposed method can distinguish among mobile devices of the same model, such as Galaxy Note 10.1 (A, B), Galaxy Note II (A–C), and iPhone 5 (A, B). Minimal misclassifications occurred among mobile devices of different models and brands, which may be a result of signal loss during Skype communication. Overall, the performance was satisfactory for ideal environments, which indicates a promising result when employing entropy-MFCC features in real-world scenarios.

Table 5.7: Confusion Matrix of SVM Based on Intra-Mobile Device Identification

ACC= 99.74%		*Predicted*																				
		G1	***G2***	***G3***	***G4***	***G5***	***G6***	***GT***	***GM***	***GS***	***H1***	***H2***	***Ip1***	***Ip2***	***I1***	***I2***	***I3***	***I4***	***I5***	***A***	***S***	***N***
Actual	***G1***	997	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
	G2	0	998	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	G3	0	0	996	0	0	1	0	0	0	0	2	0	0	0	0	0	0	0	0	1	0
	G4	0	1	0	998	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	G5	1	0	0	0	996	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	G6	0	0	0	0	0	1000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GT	0	0	0	0	0	0	997	0	0	0	0	0	0	0	0	0	0	0	2	1	0
	GM	0	2	0	0	0	0	0	997	0	0	0	0	0	0	0	0	0	1	0	0	0
	GS	0	0	0	0	0	0	0	0	997	0	0	0	0	1	0	0	0	0	1	1	0
	H1	0	0	0	2	0	0	0	0	0	997	0	0	0	0	0	0	0	0	0	0	1
	H2	0	0	1	0	0	0	0	0	0	0	998	0	0	0	0	0	0	0	0	1	0
	Ip1	1	0	0	1	0	0	0	0	0	0	0	998	0	0	0	0	0	0	0	0	0
	Ip2	1	0	0	0	0	0	0	0	0	0	0	0	997	1	0	0	0	0	0	1	0
	I1	0	0	0	0	0	0	0	0	0	0	1	0	1	996	0	1	1	0	0	0	0
	I2	0	0	0	0	0	0	0	0	1	0	0	0	0	1	998	0	0	0	0	0	0
	I3	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	997	1	0	0	0	0
	I4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	1	996	0	0	0	0
	I5	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	998	0	0	1
	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	999	1	0
	S	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	997	0
	N	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	998

5.2.2.4 Inter-mobile device identification

This experiment represents the mobile devices of the same brand in one class. Five classification algorithms were used among six classes with 10-fold cross-validation for the evaluation of the entropy-MFCC features. Table 5.8 shows that the Rotation Forest classifier performed better than all the other classifiers for inter-mobile device identification. Furthermore, for most classifiers, the overall performance slightly improved with respect to the classification results based on models. However, in terms of Naïve Bayesian classifier, the classification accuracy was reduced from 99.99 to 99.69% and the error rates was increased. Meanwhile, SVM classifier achieved comparably close accuracy rates with Rotation Forest, but its computation time was faster. Thus, the experiment revisits the performance of the SVM classifier for each particular brand.

Table 5.9 shows the confusion matrix that resulted from 10-fold cross-validation by using a six-class SVM classifier. The last row of the confusion matrix is the number of predicted instances and the last column is the total number of instances with respect to each class. The larger misclassifications exist among Samsung and Apple with 9000 and 7000 data instances in each class respectively. An average identification accuracy of 99.75% was achieved for inter-mobile device identification, which is approximately similar to the results for intra-mobile device identification by using the same classifier.

Table 5.8: Performance of Entropy-MFCC Features for Inter-Mobile Device Identification

Classifiers	***MAE***	***RMSE***	***RAE***	***RRSE***	***ACC***
SVM	0.0008	0.029	0.37 %	8.56 %	99.75 %
Neural Network	0.0015	0.0291	0.66 %	8.58 %	99.69 %
Naïve Bayesian	0.0235	0.0953	10.21 %	28.12 %	97.16 %
Rotation Forest	0.0015	0.018	0.64 %	5.32 %	99.93 %
Linear Logistic Regression	0.0024	0.0362	1.02 %	10.67 %	99.57 %

Table 5.9: Confusion Matrix of SVM Based Inter-Mobile Devices Identification

ACC= 99.75%		***Predicted***						
		Galaxy	*HTC*	*Apple*	*Asus*	*Sony*	*Nokia*	***Total***
Actual	*Galaxy*	8984	3	10	1	2	0	9000
	HTC	6	1990	2	0	0	2	2000
	Apple	14	3	6981	0	1	1	7000
	Asus	4	0	0	996	0	0	1000
	Sony	0	0	2	0	998	0	1000
	Nokia	0	0	2	0	0	998	1000
	Total	9008	1996	6997	997	1001	1001	

5.2.3 Discussion

Prior work has focused on source recording device identification from both speech and non-speech recording. Hanilçi et al. (2012), for example, used cell phone devices as an ordinary tape recorder to collect speech recordings. Although these studies proved that the MFCCs extracted from the speech recording is the most effective feature set to capture device specific features, the results lack evaluation on the robustness of MFCCs. This is because MFCC features are contextualized by the speech contents, speaker's characteristics and the environment. The experiments in this section improved the robustness of MFCCs by computing the entropy of MFCCs from near-silent segments for the source mobile device identification framework.

It is therefore speculated that by using all selected classifiers, entropy-MFCC feature set exhibits high performance against statistical moments of MFCCs. Meanwhile, for near-silent segments, the classification results obtained for the combination of entropy with frequency domain features are in exceptionally good agreement with the entropy-MFCC feature set. These findings proved the significance of eliminating convolution due to speech signals. Furthermore, in terms of classifiers, Rotation Forest and SVM classifier

achieved the best performance with respect to the classification accuracy, robustness, and computational efficiency. Some aspects of the proposed method compares well with existing research on acquisition device identification (Buchholz et al., 2009; D. Garcia-Romero & Epsy-Wilson, 2010; Hanilçi et al., 2012; Kraetzer et al., 2007; Kraetzer et al., 2012; Kraetzer et al., 2011; Panagakis & Kotropoulos, 2012b). However, this method adds an advantage to the previous approaches in the following ways: (a) entropy-MFCC features are extracted from near-silent frames, (b) entropy of Mel-cepstrum output intensifies the energy of MFCCs for near-silent frames, (c) blind identification of mobile devices over the call. This study, therefore, indicates that Entropy-MFCC features identify the distinguishing pattern in mobile devices of the same model.

5.2.4 Conclusion

Most notably, this study is the first to identify traces of the source transmitting devices by detecting the near-silent segments in a recorded conversation. Both experiments found evidence to suggest that the proposed framework can identify different source brand/model and individual mobile devices in a more practical experimental setup by using the communication through any type of service provider, such as cellular, VoIP, PSTN, and their combinations and subsets. However, some limitations are worth noting. Although the experiments in this section found promising results based on the silent recording, the proposed method was not reassessed based on speech recording. The experiments in the following sections include a follow-up work designed to evaluate the accuracy when speech is processed and transmitted by the mobile device through either VoIP or cellular networks and recorded by different stationaries. This way, it would be possible to determine the effects of speech contents, speaker characteristics, as well as the stationary devices on identification accuracy. In addition, it is possible to perform the quantitative comparison with current state-of-the-art approaches.

5.3 Performance Evaluation-Phase II: Intra- and Inter-Model Similarity

Discriminating between individual mobile devices of the same model is the most challenging problem for source mobile device identification. Hence, for individual mobile device identification, the intra-mobile device model similarity should be minimized. On the other hand, for source mobile device model identification, it is important to discriminate among different mobile device models instead of individual devices. Thus, the optimal features for source mobile device model identification should be selected in a way that the intra-mobile device model similarity is high, whereas the inter-mobile device model similarity between feature values corresponding to different mobile device models should be minimized.

As specified in Appendix B4, different models of mobile devices of the same manufacturer are equipped with the similar chipset, CPU, Wi-Fi and Cellular technology and achieved close audio quality test results. In general, devices of the same series are more difficult to separate. However, they are less difficult than the individual devices. The performance evaluation in Phase I revealed promising result for both intra- and an inter-mobile device identification; yet, the dataset (DS1 and DS2) were small and only a few mobile devices of the same model were available. Hence, Phase II aims to provide practical investigations on inter- and intra-mobile device model similarity based on optimized entropy-MFCC and ZMBic feature sets and large DS3 dataset, in order to achieve the following objectives:

(a) To visualize statistical properties of both optimized entropy-MFCC and ZMBic feature sets by using the histogram and 3-D bar plot of the covariance matrix.
(b) To evaluate and compare the performance of both feature sets for inter- and intra-mobile device model identification.

(c) To utilize the average square Euclidean distance metrics in order to plot several visualization diagrams, which compare the intra- against inter-mobile device model distances between feature sets.

5.3.1 Statistical Properties of Entropy-MFCCs

The statistical properties of the entropy-MFCC vectors have a direct influence on the performance of the source mobile device identification module. An observation-based analysis was applied to distinguish call recording signals from mobile devices of different models based on the histogram of entropy-MFCC features. Figure 5.5 demonstrates the histogram of one randomly selected feature ($\mathscr{H}_{12}$) from **[entropy-MFCC]$_{\text{Shannon}}$** feature set against an **[entropy-MFCC]$_{\text{Tsallis}}$** feature with the same index ($\mathscr{T}_{12}$) shown in Figure 5.6. Both features were extracted from near-silent signals in 12 groups corresponding to all 12 mobile device models in DS3.

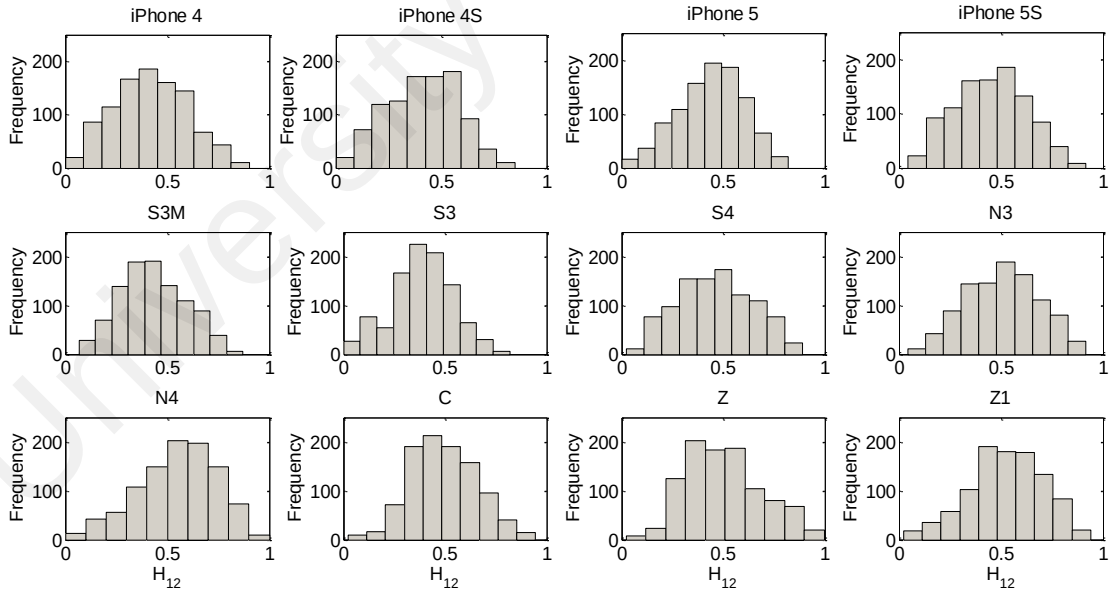


Figure 5.5: Histograms of Feature $\mathscr{H}_{12}$ for Each Mobile Device Model

It is evident that the statistical properties of both $\mathscr{H}_{12}$ and $\mathscr{T}_{12}$ extracted from near-silent recordings are discriminative among mobile devices of the different model. This observation proves the hypothesis for extracting transmitting mobile device fingerprints

from near-silent segments of the call recording signal. In addition, it can be seen that the Tsallis entropy produces less zero valued features and intensifies the value of MFCCs to more extend. This proves that more device specific information was extracted from Mel cepstrum spectrum by using the Tsallis entropy.

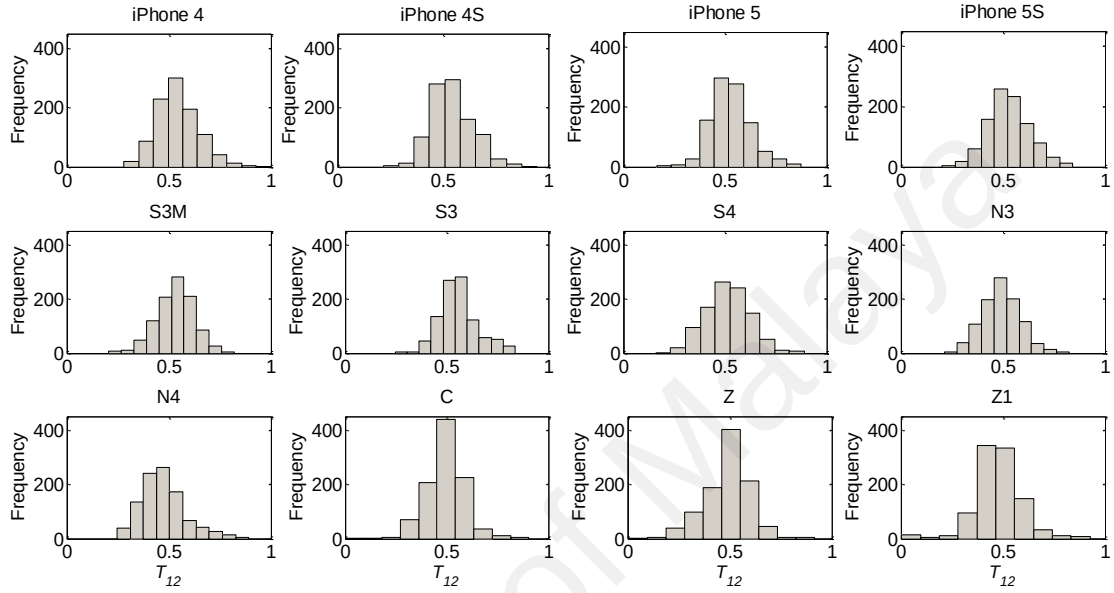


Figure 5.6: Histograms of Feature $\mathcal{T}_{12}$ for Each Mobile Device Model

Using a different approach, the correlations among the entropy-MFCC features within a feature vector were investigated by obtaining the covariance matrix of entropy-MFCCs from the data instances generated from a call recording received from Apple iPhone 4. The feature extraction algorithm in this study generates two subsets of entropy-MFCC features $\{\mathcal{H}_l\}_{i=0}^{48}$ and $\{\mathcal{T}_l\}_{i=0}^{48}$ corresponding to Shannon and Tsallis entropy. The feature set contains 49 zeroth order cepstral coefficients, including the zeroth coefficient. Hence, the covariance matrix of each entropy-MFCC subset with [49 × 49] elements was computed, to allow comparison. Furthermore, because of the large differences in the magnitude of the variance of some entropy-MFCC features compared with those of other entropy-MFCC features, this approach applied a square root operation to the covariance elements to compress the dynamic range and then plotted the absolute value of the covariance of entropy-MFCCs. As shown in Figure 5.7 and Figure 5.8, the diagonal elements represent

the standard deviation of each entropy-MFCC rather than the variance, which indicates that in both cases, the entropy-MFCCs have imbalanced energies. On the other hand, the off-diagonal covariance elements have lower magnitude. This means that both subsets of the entropy-MFCC features are weakly correlated with each other. This week correlation suggests that the optimized entropy-MFCC feature set is a good candidate for source mobile device identification.

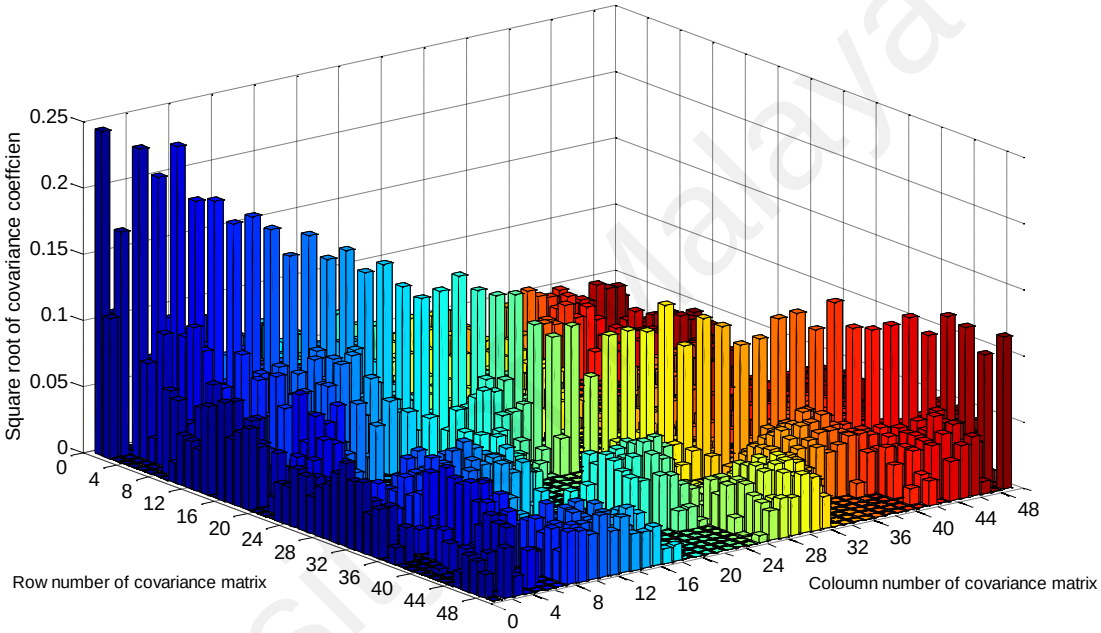


Figure 5.7: 3D-Bar Plot of the Absolute Value of the Covariance Elements of Entropy-MFCCs $\left\{\mathcal{H}_i\right\}_{i=0}^{48}$.

5.3.2 Statistical Properties of ZMBics

The statistical properties of the ZMBic vectors have also been examined in order to justify its influence on the performance of the source mobile device identification module. An observation-based analysis at first demonstrated the histogram of the ZMBic features corresponding to call recordings received from mobile devices of different models. Figure 5.9 demonstrates the histogram of the ZMBic features based on the bicoherence magnitude $\mathcal{Z}_M$ against the bicoherence phase $\mathcal{Z}_{Ph}$ in Figure 5.10, whereby they extracted from near-silent signals in 12 groups corresponding to all 12 mobile device models in

DS3. It is evident that the statistical properties of $\mathcal{Z}_{M22}$ extracted from near-silent segments of the call recording signal (in regard to mobile devices of the different model) are more distinguishable than $\mathcal{Z}_{Ph22}$.

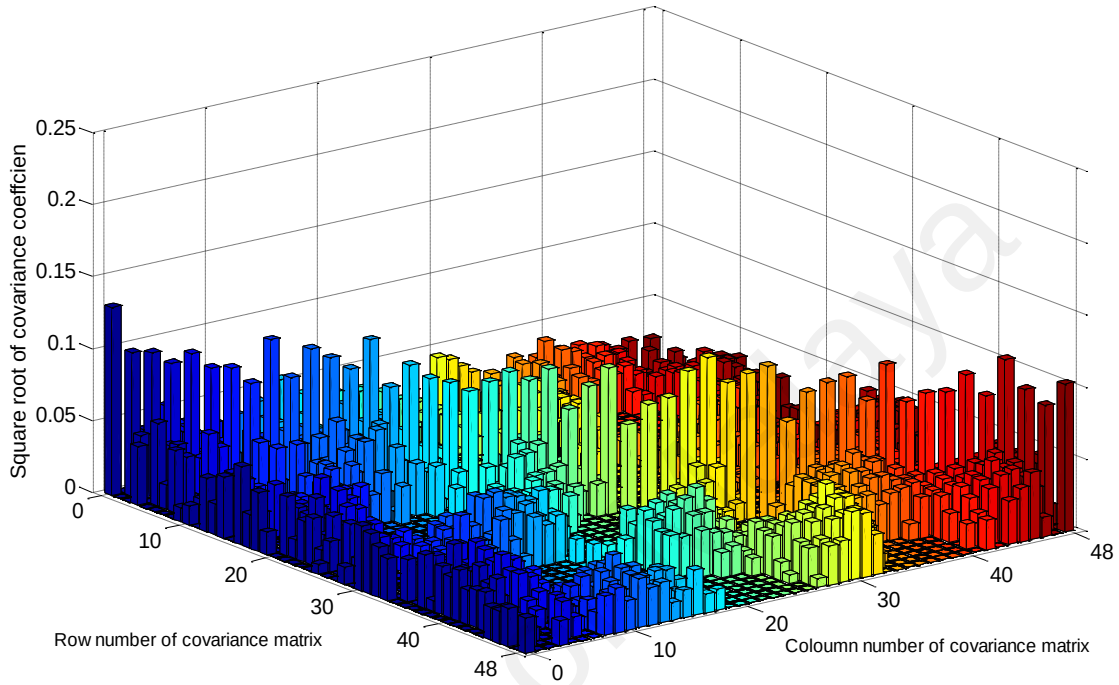


Figure 5.8: 3D-Bar Plot of the Absolute Value of the Covariance Elements of Entropy-MFCCs $\{\mathcal{I}_i\}_{i=0}^{48}$

For mobile devices of the same series such as Apple iPhone, the histogram of the $\mathcal{Z}_{Ph22}$ follows a more similar pattern, yet it is distinctive. Hence, ZM of bicoherence magnitude should be more feasible for source mobile device identification.

Using a different approach, the amount of correlations between the ZMBic features within a feature vector were investigated by obtaining the covariance matrix of the ZMBic from the data instances generated from call recordings in DS3 database. The feature extraction algorithm in this study generates two subsets of the ZMBic features $\{\mathcal{Z}_{Mi}\}_{i=1}^{28}$ and $\{\mathcal{Z}_{Phi}\}_{i=1}^{28}$ corresponding to bicoherence magnitude and phase spectrum. Hence, the covariance matrix of each ZMBic subset with [28 × 28] elements was computed, to allow

comparison. Furthermore, due to the same reason discussed in the previous subsection, a square root operation was applied to the covariance elements to compress the dynamic range and then plot the absolute value of the covariance of the ZMBic feature set. In Figure 5.11 and Figure 5.12, the diagonal elements have imbalanced values, which indicate that the energy of the ZMBic features is different.

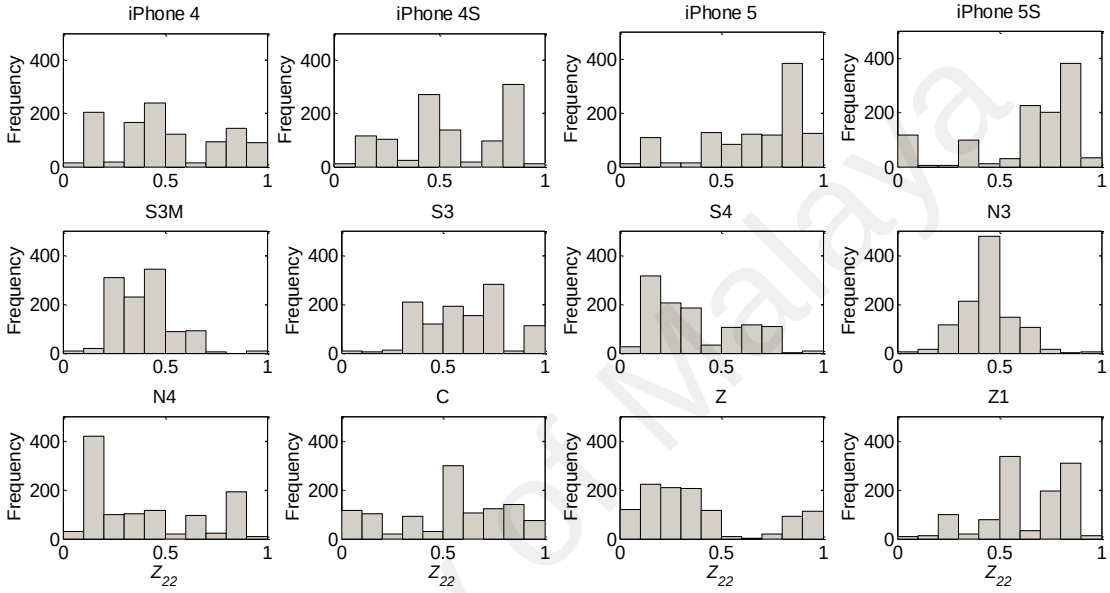


Figure 5.9: Histograms of Feature $Z_{M_{22}}$ for Each Mobile Device Model

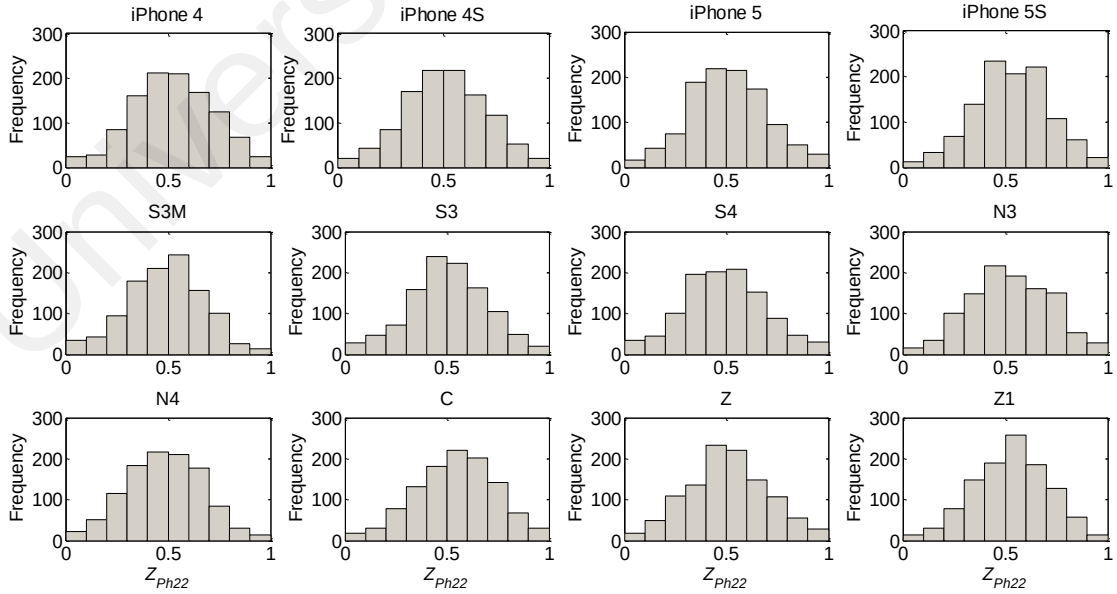


Figure 5.10: Histograms of Feature $Z_{Ph_{22}}$ for Each Mobile Device Model

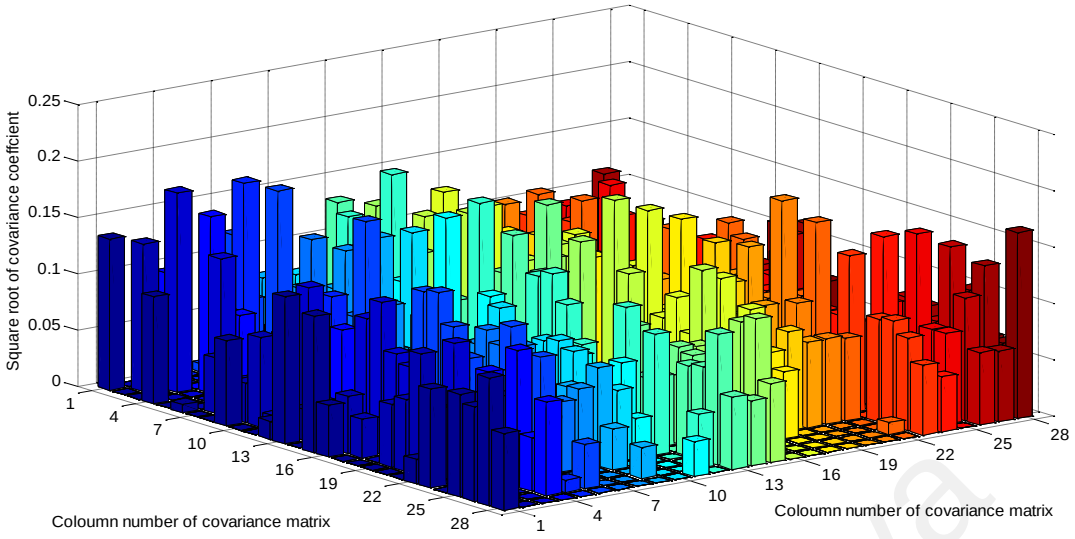


Figure 5.11: 3D-Bar Plot of the Absolute Value of the Covariance Elements of the $ZMBic_{M}$

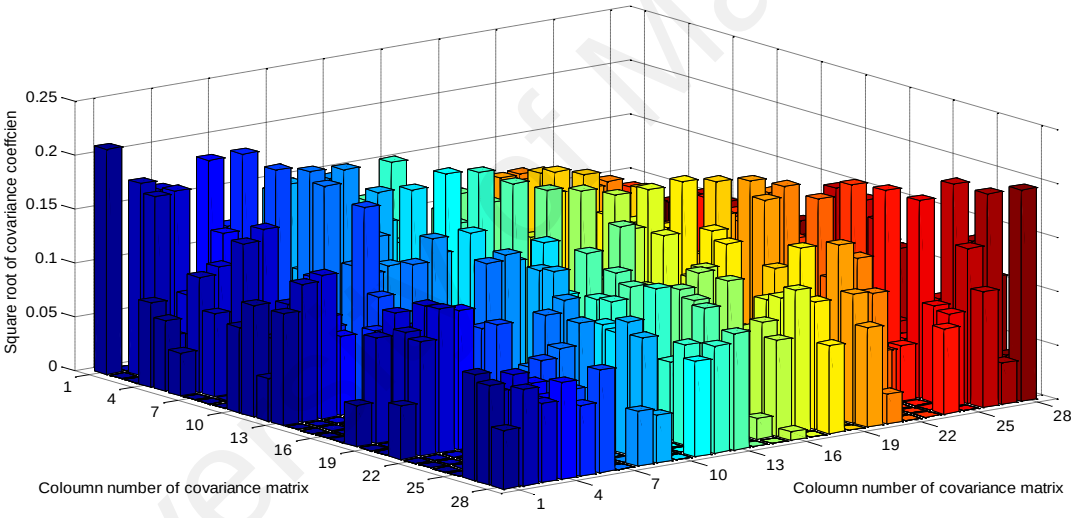


Figure 5.12: 3D-Bar Plot of the Absolute Value of the Covariance Elements of the $ZMBic_{Ph}$

On the other hand, the off-diagonal covariance elements have lower magnitude. This means that both subsets of the ZMBic features are weakly correlated with each other. This observation makes the proposed ZMBic feature sets a good candidate for source mobile device identification.

5.3.3 Intra and Inter-Model Similarity based on Entropy-MFCCs

For intra-mobile device model identification, the SVM classifier trained a set of data instances from a particular subset of call recording signals based on the individual devices

of one model. This subset consists of call recordings corresponding to Dell stationary and environment B from DS3. This way, the similarity among devices of the same series was investigated based on10 individual devices per model. For the experiments in this section, a total of 120 data instances with the length of 0.25 seconds were created corresponding to each device. In overall, the SVM classifier is trained with 60% of the representative data instances of each device for training, and the remaining data instances were used for testing. The test is repeated 100 times for each mobile device, and for each run, the call recordings are randomly selected for both the training and test set. Eventually, the experiment determines the number of correctly classified and misclassified instances by using the confusion matrix, correspondingly.

The classification results based on the confusion matrix are summarized in Table 5.10-Table 5.13 for intra-mobile device model identification in case of devices of model Apple iPhone 4, 4S, 5 and 5S. Next, the experiment labeled all the data instances corresponding to 10 individual devices from each model as the same class, then from each class randomly selected 60% of the instances as training set and the remaining instances as the test set. Table 5.14 shows the correct and incorrect classified instances in diagonal and non-diagonal cells for source inter-mobile device model identification in case of devices of the series Apple iPhone. In the overall, it is evident that the classification result obtained for individual mobile device identification is in exceptionally good agreement with source mobile device identification among mobile devices of the same series.

Table 5.10: Intra-Model Identification Performance of Entropy-MFCCs (iPhone 4).

ACC=85.8% $r_{i,1,1}^{(i4)}$		***Predicted***										
		1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***Total***
Actual	***1***	42	2	2	1	0	1	0	0	0	0	48
	2	0	39	3	1	0	1	1	1	1	1	48
	3	1	4	40	1	0	1	0	0	1	0	48
	4	0	1	3	41	1	1	0	0	1	0	48
	5	1	0	1	0	46	0	0	0	0	0	48
	6	1	1	1	0	0	42	1	1	1	0	48
	7	0	0	1	0	0	1	42	2	1	1	48
	8	0	1	1	0	0	1	0	40	4	1	48
	9	0	1	0	0	0	2	0	3	40	2	48
	10	0	1	1	0	0	1	1	2	2	40	48
	Total	45	50	53	44	47	51	45	49	51	45	

Table 5.11: Intra-Model Identification Performance of Entropy-MFCCs (iPhone 4S).

ACC=89.8% $r_{i,1,1}^{(i4S)}$		***Predicted***										
		1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***Total***
Actual	***1***	45	0	0	0	1	1	1	0	0	0	48
	2	2	44	0	0	1	0	0	0	0	1	48
	3	1	0	42	0	2	0	1	2	0	0	48
	4	0	0	0	47	1	0	0	0	0	0	48
	5	0	0	0	0	46	0	2	0	0	0	48
	6	0	0	0	0	1	43	0	2	1	1	48
	7	2	0	1	0	2	0	40	0	0	3	48
	8	0	0	1	0	1	1	0	40	3	2	48
	9	0	0	0	0	1	2	0	3	42	0	48
	10	1	0	1	0	1	0	0	0	0	45	48
	Total	51	44	45	47	57	47	44	47	46	52	

Table 5.12: Intra-Model Identification Performance of Entropy-MFCCs (iPhone 5).

ACC=84.4% $r_{i,1,1}^{(i5)}$		***Predicted***										
		1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***Total***
Actual	***1***	44	0	1	2	0	0	0	1	0	0	48
	2	0	43	0	2	0	0	1	1	1	0	48
	3	0	0	37	2	1	0	0	5	0	3	48
	4	2	1	1	38	0	0	1	4	0	1	48
	5	0	0	1	0	44	1	0	1	0	1	48
	6	0	0	0	1	1	43	0	2	0	1	48
	7	0	1	0	1	0	0	44	2	0	0	48
	8	0	0	3	0	1	0	1	41	0	2	48
	9	0	1	0	1	0	0	0	0	45	1	48
	10	0	0	3	2	2	0	0	6	0	35	48
	Total	46	46	46	47	49	44	47	63	46	44	

Table 5.13: Intra-Model Identification Performance of Entropy-MFCCs (iPhone 5S).

ACC=81.9% $r_{i,1,1}^{(i5S)}$		***Predicted***										
		1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***Total***
Actual	***1***	37	2	1	1	3	1	0	0	3	0	48
	2	1	39	1	2	1	2	0	0	1	1	48
	3	1	0	41	0	2	1	0	0	3	0	48
	4	1	1	0	40	3	1	0	0	1	1	48
	5	5	1	0	4	33	0	0	2	3	0	48
	6	1	2	0	0	0	42	0	1	2	0	48
	7	0	0	1	0	1	1	44	0	1	0	48
	8	0	0	0	1	2	0	0	42	3	0	48
	9	1	1	1	1	1	1	0	0	42	0	48
	10	0	0	0	1	1	1	0	0	1	42	48
	Total	47	46	44	50	47	50	44	45	60	44	

Table 5.14: Inter-Model Identification Performance of Entropy-MFCCs (Apple iPhone Series)

ACC=92.38% $r_{i,1,1}^{(m)}$		***Predicted***				
		i4	***i4S***	***i5***	***i5S***	***Total***
Actual	***i4***	363	11	24	2	400
	i4S	7	368	21	4	400
	i5	12	9	375	4	400
	i5S	7	1	20	372	400
	Total	389	389	440	382	

This experiment demonstrates two significant viewpoints: First, calls transmitted from individual devices of the same model are correctly assigned to the respective mobile device model with small false identification rate, and second, calls transmitted from

mobile devices with different models, but the same series are assigned to the corresponding mobile device model. However, for the inter-mobile device model identification, the false identification rate is slightly higher.

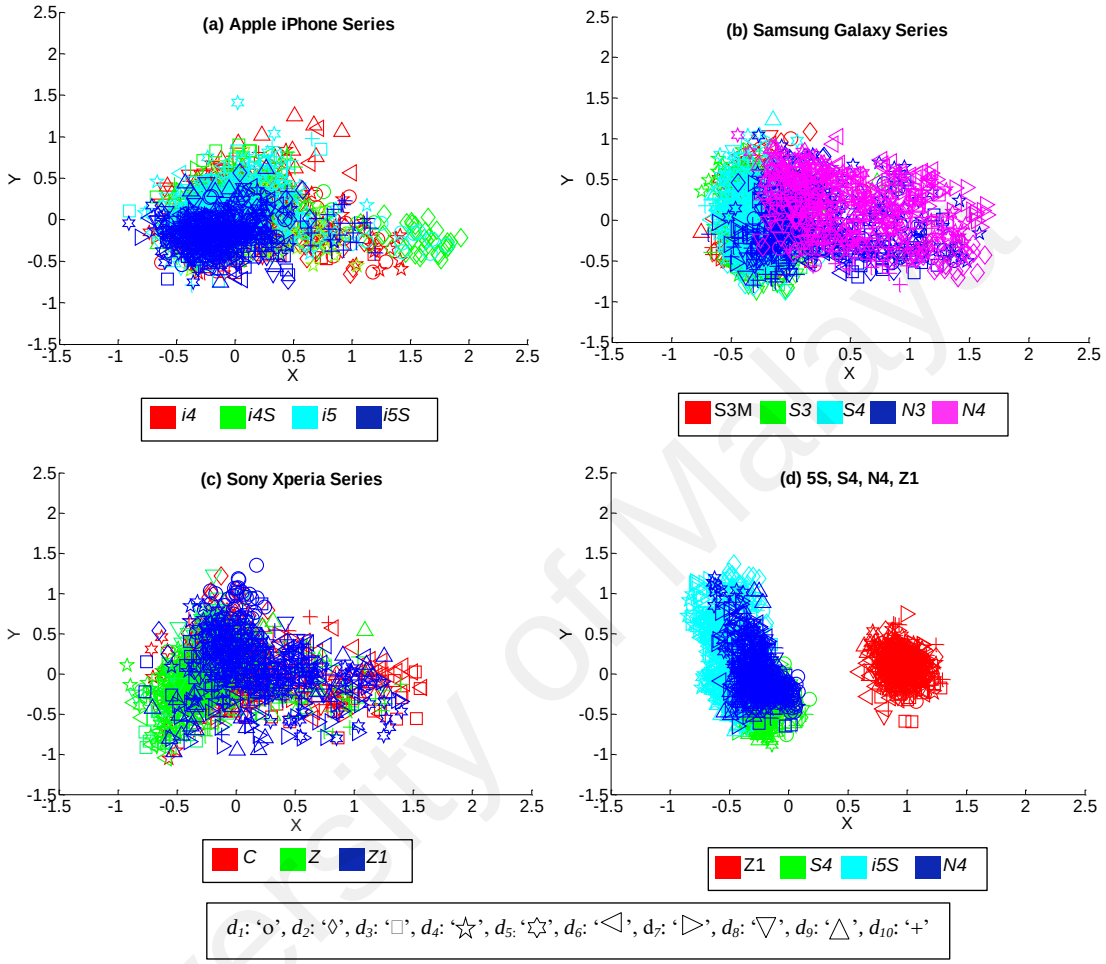


Figure 5.13: Visualization of the Inter- and Intra-Model Similarity of the Entropy-MFCC Features

To provide a visualization of intra- and inter-mobile device model similarity among all 120 utilized devices with respect to the entropy-MFCC feature set, this approach computed for each device the centroid over all corresponding feature values and then applied multi-dimensional scaling to map the centroids to 2D. As demonstrated in Figure 5.13, the intra- and inter-mobile device model similarity are visualized for (a) Apple iPhone series, (b) Samsung Galaxy series, (c) Sony Xperia series, and (d) the group of four different mobile device models. It is evident from this visualization that mobile devices of the same model are clustered together, which prove the hypothesis that the

entropy-MFCC feature set is capable of discriminating among mobile device models. Based on these visualizations the following points could be learned:

In Figure 5.13a-c, the discrimination between devices of the same series such as Apple iPhone 4, 4S, 5 and 5S is difficult, but the discrimination between different units of mobile devices of the same model is more difficult. On the other hand, a clear discrimination between mobile devices of different series such as Apple iPhone 5S, Samsung Galaxy S4, Samsung Galaxy Note 4 and Sony Xperia Z1 is possible. In the majority of cases, it is hard to discriminate among mobile devices of the same model due to the large intra-model similarity, whereas the inter-model similarity is comparably smaller. This observation proves the suitability of the entropy-MFCC feature set for source mobile device model identification.

5.3.4 Intra and Inter-Model Similarity based on ZMBics

The experiment utilized the same experimental setup and dataset as for the previous section for intra- and inter-mobile device identification by using SVM classifier based on ZMBic features. The classification results for intra-mobile device model identification in case of Apple iPhone 4, 4S, 5 and 5S are summarized in Table 5.15-Table 5.18, whereas for inter-mobile device model identification, the result is shown in Table 5.19.

Table 5.15: Intra-Model Identification Performance of ZMBics (iPhone 4).

ACC=95.6% $r_{i,1,1}^{(i4)}$		***Predicted***										
		1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***Total***
Actual	***1***	46	0	0	1	0	1	0	0	0	0	48
	2	0	46	0	0	0	1	0	0	0	1	48
	3	1	0	47	0	0	0	0	0	0	0	48
	4	0	0	0	46	0	1	0	0	0	1	48
	5	1	0	0	1	46	0	0	0	0	0	48
	6	0	1	0	1	0	45	0	0	0	1	48
	7	0	0	0	1	0	0	46	0	1	0	48
	8	0	0	0	1	0	1	0	46	0	0	48
	9	0	0	0	2	0	1	0	0	45	0	48
	10	0	0	0	2	0	2	0	0	0	44	48
	Total	48	47	47	55	46	52	46	46	46	47	

Table 5.16: Intra-Model Identification Performance of ZMBics (iPhone 4S).

ACC=94.4% $r_{i,1,1}^{(i4S)}$		***Predicted***										
		1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***Total***
Actual	***1***	45	0	1	0	1	0	1	0	0	0	48
	2	0	46	0	1	1	0	0	0	0	0	48
	3	1	0	46	1	0	0	0	0	0	0	48
	4	0	0	0	46	0	0	1	0	0	1	48
	5	1	0	0	0	45	0	1	0	1	0	48
	6	0	0	0	0	0	48	0	0	0	0	48
	7	1	0	0	0	1	0	46	0	0	0	48
	8	0	0	0	1	0	0	0	47	0	0	48
	9	0	0	0	1	2	0	0	0	45	0	48
	10	0	1	0	1	0	0	1	0	0	45	48
	Total	48	47	47	51	50	48	50	47	46	46	

Table 5.17: Intra-Model Identification Performance of ZMBics (iPhone 5).

ACC=96.9% $r_{i,1,1}^{(i5)}$		***Predicted***										
		1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***Total***
Actual	***1***	45	2	0	0	1	0	0	0	0	0	48
	2	1	45	0	0	1	0	1	0	0	0	48
	3	1	0	47	0	0	0	0	0	0	0	48
	4	0	1	0	46	1	0	0	0	0	0	48
	5	1	0	0	1	46	0	0	0	0	0	48
	6	1	1	0	0	0	46	0	0	0	0	48
	7	0	1	0	0	0	0	47	0	0	0	48
	8	1	1	0	0	0	0	0	46	0	0	48
	9	0	0	0	0	0	0	0	0	48	0	48
	10	1	1	0	0	0	0	0	0	0	46	48
	Total	51	52	47	47	49	46	48	46	48	46	

Table 5.18: Intra-Model Identification Performance of ZMBics (iPhone 5S).

ACC=94.8% $r_{i,1,1}^{(i5S)}$		***Predicted***										
		1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***Total***
Actual	***1***	46	0	1	0	0	0	0	0	0	1	48
	2	0	47	0	0	1	0	0	0	0	0	48
	3	0	0	46	1	0	0	0	0	0	1	48
	4	0	0	1	45	0	0	1	0	0	1	48
	5	0	0	0	0	46	0	1	0	0	1	48
	6	0	0	1	1	0	46	0	0	0	0	48
	7	0	0	1	0	1	0	46	0	0	0	48
	8	0	0	1	0	0	0	0	47	0	0	48
	9	0	0	0	0	0	0	0	0	47	1	48
	10	0	0	1	1	0	0	0	0	1	45	48
	Total	46	47	52	48	48	46	48	47	48	50	

Table 5.19: Inter-Model Identification Performance of ZMBics (Apple iPhone).

ACC=94.4% $r_{i,1,1}^{(m)}$		***Predicted***				
		i4	***i4S***	***i5***	***i5S***	***Total***
Actual	***i4***	347	2	6	5	360
	i4S	8	338	6	8	360
	i5	4	9	334	13	360
	i5S	5	7	8	340	360
	Total	364	356	354	366	

To provide a visualization of intra- and inter-mobile device model similarity among all 120 utilized devices with respect to ZMBic feature set, this approach computed for each device the centroid over all corresponding feature values and then applied multi-dimensional scaling to map the centroids to 2D. As demonstrated in Figure 5.14, the intra-

and inter-mobile device model similarity are visualized for (a) Apple iPhone series, (b) Samsung Galaxy series, (c) Sony Xperia series, and (d) the group of four different mobile device models. Although the feature values corresponding to different mobile device models were represented in different colors, it is clearly can be seen that the feature values belong to the same mobile device unit representing by different symbols are clustered together. This observation is consistent with results obtained in running intra- and inter-mobile device model identification experiment.

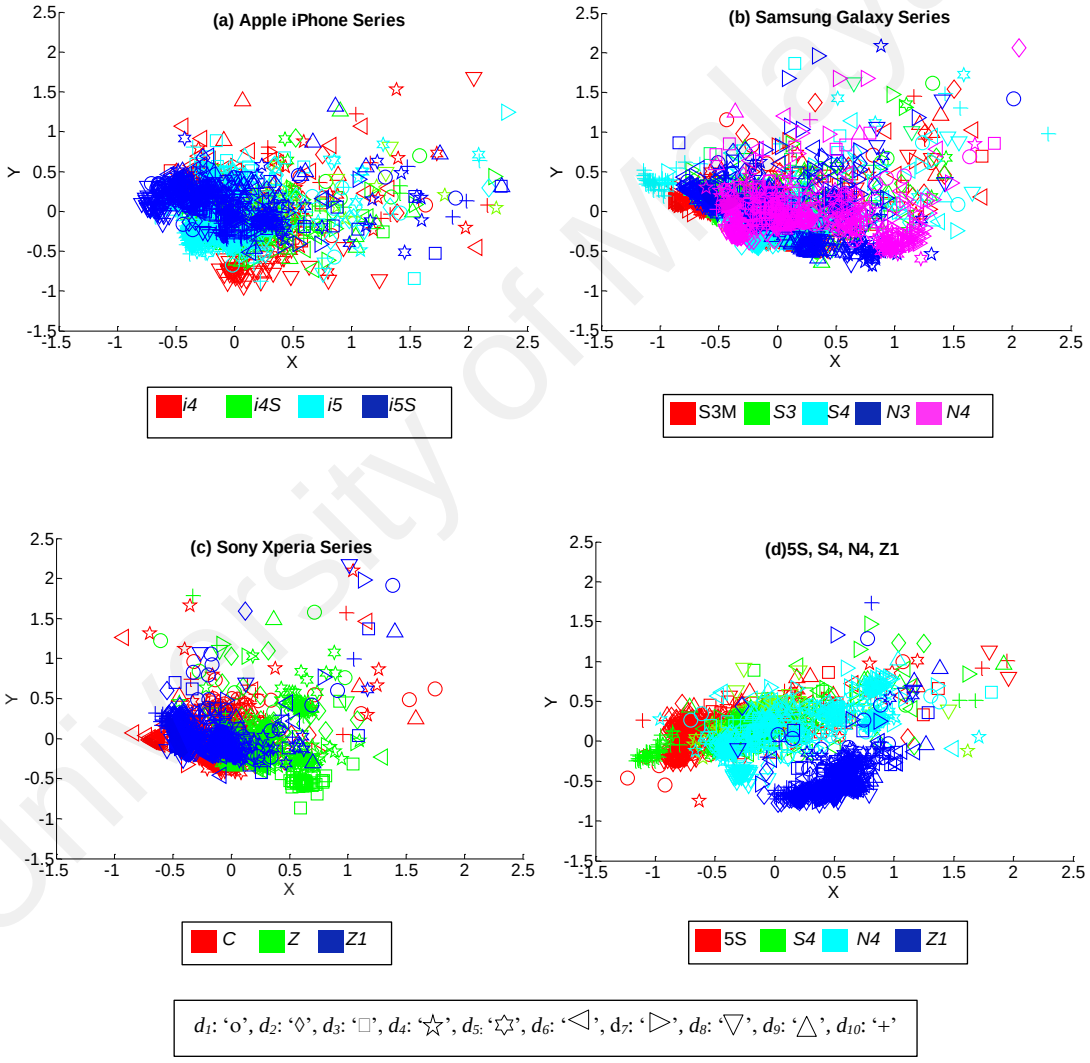


Figure 5.14: Visualization of the Inter- and Intra-Model Similarity of the ZMBic$_M$ Features

5.3.5 Conclusion

Overall, the experiments in this section demonstrated two significant viewpoints: First, for both entropy-MFCCs and ZMBic feature sets calls transmitted from individual

devices of the same model were perfectly assigned to the respective mobile device units with small false identification rates; on the other hand, ZMBic feature set with smaller intra-model similarity achieved higher identification accuracy rates for individual source mobile device identification. Second, entropy-MFCC feature set revealed higher identification rates for inter-mobile device model identification with higher intra-model similarities and smaller inter-model similarities between features.

5.4 Performance Evaluation-Phase III: Mobile Device Model Identification in Closed set using Entropy-MFCC

The first phase highlighted the feasibility of entropy-MFCC features from near-silent signals, particularly for individual source mobile device identification using recorded VoIP calls. However, the Phase I has not considered the influence of the changes in filterbank spacing of MFCCs and the non-extensive property of Tsallis entropy to optimize this feature set for source mobile device identification. The performance evaluation in Phase II suggested the suitability of the optimized entropy-MFCCs for source mobile device model identification. Hence, the Phase III extends the evaluation study with respect to the optimized entropy-MFCC feature set along with critical evaluation for source mobile device model identification and also aims to satisfy following objectives:

(a) To investigate the influence of applying a different filterbank spacing in the MFCC feature extraction process as it allows to rearrange the MFCC filterbanks in order to capture mobile device response function from near-silent signals compared to selected state-of-the-art feature sets.

(b) To investigate the effect of applying different classification algorithms in the source mobile device model identification process by taking advantage of a set of classifiers implemented in Weka.

(c) To investigate the effect of applying the different number of data instances, devices and models for source mobile device model identification, as the classifier performance depends on the number of selected training and testing data instances within mobile devices, models or brands, the number of devices per mobile device model, and the number of mobile device models and brands.

(d) To investigate the effect of different influences on the recording process such as environments, speakers, type of communication and stationary for source mobile device model identification.

(e) To investigate the effect of applying selected post-processing operations on the call recording signal as the classifier performance may reduce when the training set differs from the test set due to such operations.

Hence, five main experiments were conducted based on mobile device model identification corresponding to the call recording samples of DS3, as described in Section 4.1.1. This dataset consists of all 120 mobile devices in 12 different models, whereby there are 10 different mobile device units from each model. The experiment generated a total of 120 data instances from each call recording of length approximately 180 seconds (as entitled in Section 4.1.2), whereby different subsets of the call recordings were utilized along with these devices corresponding to the aim of experiment ($\mathcal{R}_{train,\ test} = \{r_{i,j,k}^{(m)} \mid d_i \in \mathcal{D}_{train,test}^{(m)} \wedge s_j \in \mathcal{S}_{train,test} \wedge e_k \in \mathcal{E}_{train,test}\}$). Experiments' results in this section are based on 10-fold cross validation; meanwhile, the data instances corresponding to the training and test subsets were selected accordingly. The evaluation was represented as closed set because the test dataset was processed by one of the known models utilized in building the training model.

Furthermore, the experiments utilized 49 zeroth order MFCCs, whereby the Mel-cepstrum output consists of one frame per row, and each frame includes 49 coefficients. Later, the algorithm generates a total of 49 entropy-MFCC features by computing the Shannon entropy of the MFCCs and 49 entropy-MFCC features by computing the Tsallis entropy of the MFCCs. Eventually, the algorithm merged both feature sets and generated a total of 98 entropy-MFCC features. The following sections include the description of the experiments, results, and discussion.

5.4.1 Benchmarking Feature sets

The experiments evaluate the feasibility of the entropy-MFCC feature set for source mobile device model identification through evaluation of the proposed optimization techniques and comparison against state-of-the-art feature sets. This process justifies the choices made to handle MFCCs and entropy of MFCCs extraction process during optimizing the entropy-MFCC feature set for source mobile device model identification. The first experiment focused on the Mel filterbank spacing in computing the log energy, zeroth, first and second order MFCCs. The second experiment employed the Tsallis entropy (Ji-Kai et al., 2012; Pardede & Shinoda, 2012; Sparavigna, 2015) in addition with well-known Shannon entropy and evaluated its performance for different parameter settings. The third experiment utilized mean of MFCCs, LFCCs, and BFCCs. In addition, this experiment evaluates the performance of MFCCs based GSV (D. Garcia-Romero & Epsy-Wilson, 2010; Hanilci & Kinnunen, 2014) to allow comparison. Therefore, the experiments utilized the data instances corresponding to call recording subset $\mathcal{R}_{train,\,test} = \{r_{i,1,2}^{(m)} \mid d_i \in \mathcal{D}_{train,test}^{(m)} \wedge s_1 \in \mathcal{S}_{train,test} \wedge e_2 \in \mathcal{E}_{train,test}\}$, in which the mobile devices were located in environment B and the calls were recorded with Dell stationary. The experiments in this section make use of the LIBSVM implementation of the SVM classifier with an RBF by using MATLAB interface, as discussed in Section 4.2. The algorithm implemented the multi-class classification option of the LIBSVM to discriminate between different mobile device models. Thus, the basic parameters of an SVM with an RBF kernel was optimized using the grid-search with γ in the range of $2^{3,\,2,\ldots,-15}$ and C in the range of $2^{-5,-4,\ldots,15}$. Consequently, the best value for γ was set to $2^{-0.9}$ and the best value for C was set to 2^{9}.

5.4.1.1 Performance comparison in applying different control parameters during feature extraction

This experiment first optimized the entropy-MFCC feature set based on the control parameters of the feature extraction methods that discussed in Section 4.1.3. The control parameters include the number of cepstral coefficients (n), number of filters in filterbank (M), length of frames in samples (N), the minimum and maximum frequencies corresponding to the low end of the lowest filter and high end of the highest filter (f_{min} and f_{max}) and entropic index (q). The experiment set the default values for the length of frames in samples N=256, f_{min}= 0 kHz and f_{max}=16 kHz, and q = 2 and then evaluated the performance of source mobile device model identification module for the increasing number of cepstral coefficients n = {12, 24, 36, 48}. For each trial, after selecting the number of cepstral coefficients, the possible number of filters in the filterbank should be larger than n and could be selected from M = {13, 19, 25, 31, 37, 43, 49} series.

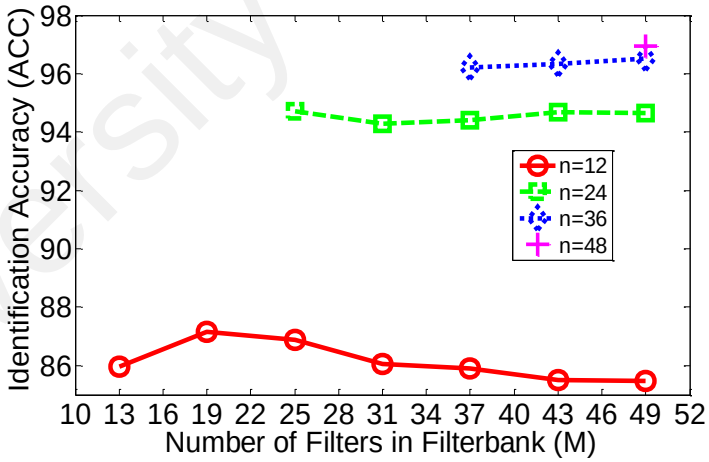


Figure 5.15: Identification Accuracies for Different Number of Cepstral Coefficients and Filters

In Figure 5.15, it is clearly can be seen that increasing the number of cepstral coefficients considerably improves the identification accuracy. For n = 12, the identification accuracy at first increased to 87.17% for M = 25 and then considerably reduced to 85.48% for M = 49. Moreover, for M=25, by increasing the number of zeroth order cepstral coefficients to n=24, the identification accuracy significantly increased to

94.71%. Eventually, the identification accuracy improved up to 96.52% for n = 36 and M = 49 and optimized at 96.94% for n = 48 and M =49.

Furthermore, in order to optimize the f_{min} and f_{max} values for the MFCC extraction, the identification accuracy was determined for a variety of f_{min} and f_{max} values, as plotted in Figure 5.16. It is evident that the best identification accuracy was obtained for f_{min}= 0 kHz and f_{max}=8 kHz. After optimizing the control parameters in MFCC extraction (N=256, f_{min}= 0 kHz, f_{max}=8 kHz, n = 48, and M = 49), the entropic index parameter q was tuned for optimal identification accuracy. From Figure 5.17, the identification accuracy remains at its highest range (>97%) while q is in the interval [0.1-0.25] and [1.25-1.75], whereby the optimal result was achieved at 97.3% for q = 0.1.

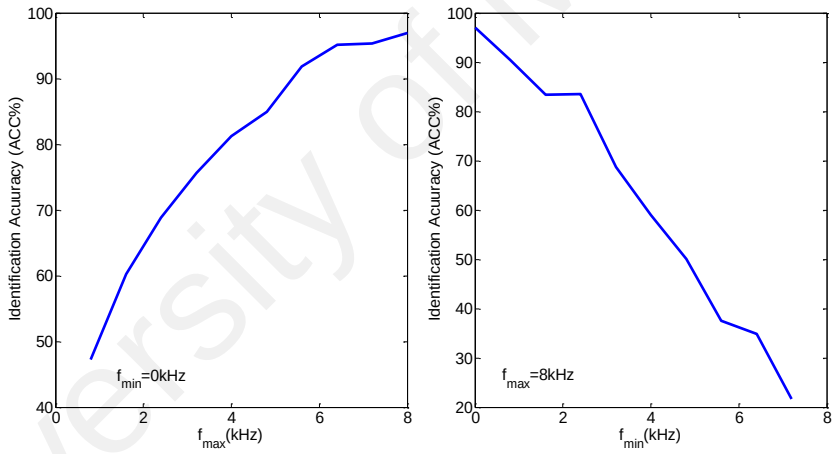


Figure 5.16: Identification Accuracy for Different F_{min} and F_{max} Frequency Values

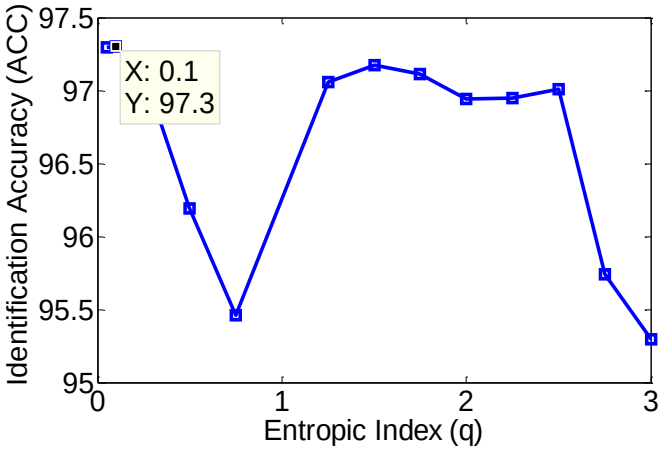


Figure 5.17: Identification Accuracies for Different Entropic Index Parameters in Tsallis Entropy

5.4.1.2 Performance comparison in classifying mobile device models based on state-of-the-art feature sets

As discussed in Section 5.3, the optimal feature set for source mobile device model identification should be able to discriminate among different mobile device models and not among different devices. Based on this strategy, this subsection will first analyze the performance of the entropy-MFCC feature sets with optimized parameters estimated in Section 5.4.1.1 (N=256, f_{min}= 0 kHz, f_{max}=8 kHz, n = 48, M = 49 and q=0.1) by using the LIBSVM classifier with RBF kernel, and optimized C and γ parameters. In the literature research during cepstral analysis, it is possible to utilize Delta and Delta Delta MFCC coefficients; alternatively, it is common to apply normalization techniques such as statistical moments and GSVs on MFCCs in order to reduce the sequences of MFCC vectors to the single feature vector (Alam et al., 2011; D. Garcia-Romero & Epsy-Wilson, 2010; Hanilci & Kinnunen, 2014).

Table 5.20: Identification Accuracies for Optimized Entropy-MFCC Feature Set

Features	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
[entropy-MFCC]$_{Shannon}$	0.013	0.113	8.36%	40.88%	0.916	92.34%
[entropy-MFCC]$_{Tsallis}$	0.006	0.076	3.74%	27.36%	0.963	**96.57%**
[entropy-MFCC]$_{Shannon}$ + [entropy-MFCC]$_{Tsallis}$	0.003	0.058	2.23%	21.13%	0.978	**97.96%**
[entropy-(MFCC+Delta)]$_{Shannon}$ + [entropy-(MFCC+Delta)]$_{Tsallis}$	0.014	0.118	9.04%	42.52%	0.910	91.71%
[entropy-(MFCC+Delta Delta)]$_{Shannon}$ + [entropy-(MFCC+Delta Delta)]$_{Tsallis}$	0.026	0.163	17.28%	58.79%	0.827	84.16%

Table 5.20 compares the identification accuracies for optimized entropy-MFCCs using Shannon against Tsallis entropy as well as its combination. Meanwhile, the identification accuracies were evaluated with respect to zeroth order, first order, and second order MFCCS. It is clearly can be seen, that the optimized Tsallis entropy outperforms Shannon entropy by computing the 49 coefficients of zeroth order MFCCs, whereby the best identification accuracy was determined by using the combined set of [entropy-

MFCC]$_{Shannon}$ + [entropy-MFCC]$_{Tsallis}$. However, the use of Delta and Delta Delta coefficients of MFCCs considerably reduced the identification accuracy of the combined set. Similarly, Table 5.21 reveals the identification accuracies for nine different feature sets discussed in Section 4.1.3 based on the 49 cepstral coefficients generated using the cepstrum-based feature extraction algorithm proposed in this study. Although for this observation LFCCs and BFCCs sequences also consist of 49 coefficients, it can be seen that MFCC always outperforms LFCC and BFCC feature set.

Table 5.21: Identification Accuracies for Nine Different Feature Sets by using the 49 Cepstral Coefficients

Features	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
[entropy-MFCC]$_{Shannon}$ + [entropy-MFCC]$_{Tsallis}$	0.003	0.058	2.23%	21.13%	0.978	**97.96%**
[entropy-LFCC]$_{Shannon}$ + [entropy-LFCC]$_{Tsallis}$	0.005	0.071	3.27%	25.55%	0.967	97%
[entropy-BFCC]$_{Shannon}$ + [entropy-BFCC]$_{Tsallis}$	0.007	0.084	4.595%	30.32%	0.954	95.79%
mean-MFCC	0.005	0.071	3.29%	25.68%	0.967	**96.98%**
mean-LFCC	0.006	0.074	3.59%	26.78%	0.964	96.71%
mean-BFCC	0.006	0.08	4.165%	28.86%	0.958	96.18%
GSV-MFCC	0.026	0.161	16.85%	58.05%	0.832	**85.55%**
GSV-LFCC	0.025	0.156	16.01%	56.58%	0.840	84.33%
GSV-BFCC	0.028	0.166	18.03%	60.06%	0.820	83.47%

On the other hand, Table 5.22 presents the performance metrics for source mobile device model identification when utilized the computation of entropy, mean and GSVs of the basic MFCCs, LFCCs and BFCCs and its dynamic coefficients in order to allow comparison with the state-of-the-art reviews approaches discussed in Section 2.4. It is evident that the performance for all feature sets based on 13 default coefficients is always significantly lower than the feature sets computed by using the optimized cepstral coefficients. Despite the better performance of the optimized entropy-MFCCs, the 13-Dimensional entropy-MFCCs achieved lower performance in compare to 13-Dimensional mean-MFCCs, mean-BFCCs, and mean-LFCCs. This shows that the 13-Dimensional entropy-MFCCs are less robust against increasing the number of mobile

devices to 120 compared to the experiments in Section 5.2 with only 10 to 21 mobile devices.

Table 5.22: Identification Accuracies for Nine Different Feature Sets by Using 13 Default Cepstral Coefficients

Features	*MAE*	*RMSE*	*RAE*	*RRSE*	*Kappa Statistic*	*ACC*
entropy-MFCC	0.035	0.187	22.79%	67.52%	0.772	**79.1%**
entropy-[MFCC + Delta]	0.044	0.21	28.70%	75.77%	0.713	73.68%
entropy-[MFCC + Delta Delta]	0.042	0.205	27.62%	74.32%	0.724	74.68%
entropy-LFCC	0.055	0.234	35.93%	84.77%	0.641	**67.07%**
entropy-[LFCC + Delta]	0.064	0.254	42.13%	91.8%	0.579	61.38%
entropy-[LFCC + Delta Delta]	0.061	0.247	40.04%	89.49%	0.600	63.3%
entropy-BFCC	0.039	0.198	25.53%	71.46%	0.745	**76.6%**
entropy-[BFCC + Delta]	0.049	0.222	32.12%	80.15%	0.679	70.56%
entropy-[BFCC + Delta Delta]	0.047	0.217	30.78%	78.46%	0.692	71.79%
mean-MFCC	0.028	0.167	18.31%	60.52%	0.817	**83.21%**
mean-[MFCC + Delta]	0.041	0.202	26.68%	73.05%	0.733	75.55%
mean-[MFCC + Delta Delta]	0.055	0.234	35.96%	84.80%	0.640	67.04%
mean-LFCC	0.033	0.182	21.69%	65.87%	0.783	**80.11%**
mean-[LFCC + Delta]	0.045	0.211	29.27%	76.50%	0.707	73.17%
mean-[LFCC + Delta Delta]	0.06	0.245	39.27%	88.62%	0.607	64.01%
mean-BFCC	0.031	0.176	20.27%	63.66%	0.797	**81.42%**
mean-[BFCC + Delta]	0.044	0.210	28.89%	76.02%	0.711	73.52%
mean-[BFCC + Delta Delta]	0.058	0.240	37.69%	86.82%	0.623	65.45%
GSV-MFCC	0.042	0.205	27.60%	74.29%	0.724	**74.71%**
GSV-[MFCC + Delta]	0.045	0.211	29.22%	76.44%	0.708	73.22%
GSV-[MFCC + Delta Delta]	0.047	0.217	30.71%	78.37%	0.693	71.85%
GSV-LFCC	0.046	0.214	29.86%	77.28%	0.701	**72.63%**
GSV-[LFCC + Delta]	0.046	0.215	30.11%	77.60%	0.699	72.40%
GSV-[LFCC + Delta Delta]	0.050	0.223	32.50%	80.63%	0.675	70.21%
GSV-BFCC	0.046	0.215	30.38%	77.95%	0.696	**72.15%**
GSV-[BFCC + Delta]	0.048	0.220	31.63%	79.53%	0.684	71.01%
GSV-[LFCC + Delta Delta]	0.226	0.051	33.50%	81.86%	0.665	69.29%

Figure 5.18 compares the overall ROC curves of the LIBSVM classifier among all feature sets and label class for source mobile device model identification. In overall, the performance of the feature sets in ROC curve is consistent with the results achieved from Table 5.21 and Table 5.22. The ROC area for the optimized 98-Dimensional entropy-MFCC features was close to one, but the value was smaller for other feature sets. This finding indicates that for optimized entropy-MFCC features, the false-positive rate is close to zero, and the true positive rate is close to one. Moreover, the ROC area for the 13 zero order default entropy-MFCC is slightly less than mean-MFCC. This is because the Mel filterbanks in MFCCs were designed to capture the perception of the audio signal,

and they are inefficient for characterizing the mobile device response function from near-silent signals.

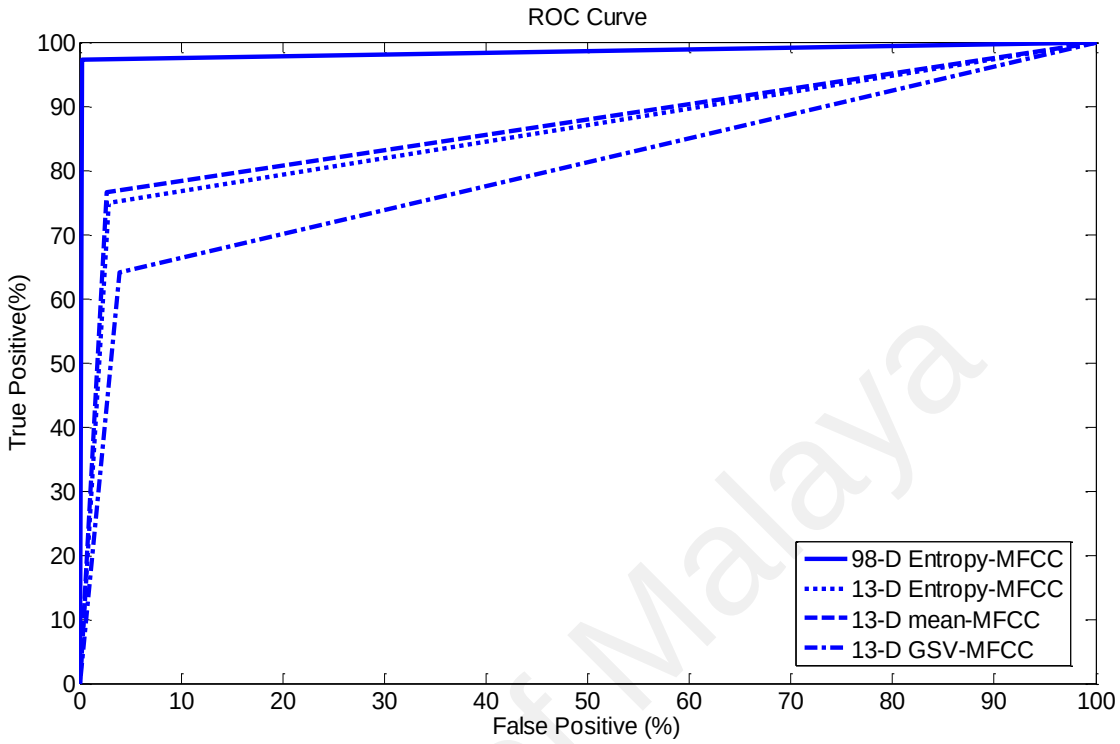


Figure 5.18: Overall ROC Curve of LIBSVM Classifier Using Different Feature Sets on the Class of Labels

5.4.2 Benchmarking Classifiers

This section provides a summarizing review of the detection performance of existing classification algorithms on the source mobile device model identification approach proposed in this study. The evaluation results depend on the performance of the classification algorithms that implemented in the data mining tool WEKA (v.3.6.1). In overall, the classification approaches are known as supervised and unsupervised learning techniques, whereby in Section 5.2.2 the experiment results revealed the evidence that both approaches can be applied in source mobile device model identification, but by using a different extent of success.

5.4.2.1 Performance comparison in classifying mobile device models based on supervised learning techniques

The large number of 74 supervised learning algorithms were implemented in WEKA (v.3.6.10). Hence, this section aims to perform the validation experiment by using all 54 applicable classifiers, in order to determine the most suitable subset of the classifiers for source mobile device model identification. The best performing classifier subset could be used as a point of reference for the source mobile device model identification evaluations within this study. The experiment was conducted based on the subset of the call recording dataset from DS3, in which the mobile devices were located in environment B, and the calls were recorded with Dell stationary. The experimental setup utilized 10-fold cross validation and a total of 14,400 feature vectors (corresponding to 120 mobile devices times 120 data instances generated from each call recording file), whereby a dimensionality of the feature vector of $[\text{entropy-MFCC}]_{\text{Shannon}}$ + $[\text{entropy-MFCC}]_{\text{Tsallis}}$ is 98. All classifiers were utilized with default parameters and settings unless stated otherwise.

The experiment results based on performance metrics shown in Table 5.23 reveal strong variation in the detection performance of the aforementioned classifiers. Hence, the classification algorithms are rearranged in this table with respect to the identification accuracy rates and error metrics, from high performing to the low performing classifiers. This allows determining the five best performing classifiers with $99.63\% < ACC < 95.74\%$. Surprisingly, the *IB1*, *IBk*, and *KStar* classifiers recognized as Lazy classifiers achieved significantly highest accuracy rate of 99.63% and the lowest error metrics. These classifiers hold the training instances and perform no action until classification time. *IB1* is the basic instance learner that finds the training instance closest in Euclidean distance to the given test instance and predicts the same class as the training

instance. *IBk* is the *k*-nearest-neighbor classifier. *KStar* is the nearest neighbor algorithm with a generalized distance function by using transformations.

Table 5.23: Comparison of the Performance Metrics Achieved with the Entropy-MFCC Feature Set

Classifier	*MAE*	*RMSE*	*RAE*	*RRSE*	*Kappa Statistic*	*ACC*
IB1	0.001	0.025	0.41%	9%	0.996	**99.63%**
IBk	0.001	0.025	0.51%	9%	0.996	**99.63%**
MultiScheme	0.001	0.025	0.41%	9%	0.996	**99.63%**
StackingC	0.002	0.025	1.09%	9%	0.996	**99.63%**
Vote	0.001	0.025	0.41%	9%	0.996	**99.63%**
KStar	0.002	0.046	1.51%	16.54%	0.985	**98.60%**
Grading	0.003	0.056	2.08%	20.41%	0.979	**98.09%**
LIBSVM	0.003	0.058	2.23%	21.13%	0.978	**97.95%**
MultiClassClassifier	0.15	0.272	98.43%	98.43%	0.953	**95.74%**
Stacking	0.007	0.084	4.66%	30.53%	0.953	**95.72%**
MultilayerPerceptron	0.034	0.162	22.14%	58.48%	0.805	82.08%
RotationForest	0.082	0.179	53.78%	64.85%	0.793	81.02%
End	0.077	0.175	50.46%	63.28%	0.783	80.11%
RandomComittee	0.087	0.188	56.78%	68.16%	0.752	77.24%
RandomSubSpace	0.101	0.202	66.01%	73.18%	0.72	74.31%
SMO	0.14	0.258	91.78%	93.25%	0.711	73.48%
Bagging	0.088	0.192	57.74%	69.37%	0.7	72.49%
Dagging	0.141	0.259	92.27%	93.55%	0.679	70.58%
RandomForest	0.095	0.202	62.41%	73.13%	0.677	70.42%
Logistic	0.069	0.189	45.23%	68.49%	0.657	68.60%
ClassificationViaRegression	0.078	0.196	51.27%	71.02%	0.652	68.09%
SimpleLogistic	0.074	0.19	48.47%	68.76%	0.651	67.97%
CVPrameterSelection	0.06	0.246	39.45%	88.82%	0.606	63.84%
PART	0.068	0.254	44.75%	91.76%	0.561	59.77%
J48graft	0.07	0.252	45.62%	91.07%	0.556	59.34%
AttributeSelectedClassifier	0.071	0.251	46.52%	90.77%	0.55	58.74%
J48	0.072	0.254	47.30%	91.91%	0.54	57.83%
SimpleCart	0.08	0.241	52.54%	87.15%	0.538	57.69%
BayesNet	0.077	0.241	50.16%	87.09%	0.519	55.95%
JRip	0.088	0.232	57.60%	84.01%	0.508	54.86%
REPTree	0.091	0.236	59.63%	85.21%	0.504	54.52%
NaïveBayes	0.082	0.253	53.42%	91.58%	0.481	52.39%
NaïveBayesSimple	0.082	0.253	53.43%	91.58%	0.481	52.39%
NaïveBayesUpdatable	0.082	0.253	53.42%	91.58%	0.481	52.39%
LogitBoost	0.11	0.23	71.79%	83.28%	0.462	50.67%
NaïveBayesMultinomial	0.147	0.266	95.93%	96.36%	0.44	48.69%
RandomTree	0.088	0.296	57.38%	107.13%	0.426	47.40%
OrdinalclassClassifier	0.096	0.259	62.80%	93.56%	0.397	44.70%
RacedIncrementLogitBoost	0.107	0.245	69.82%	88.56%	0.394	44.40%
FilteredClassifier	0.099	0.284	64.94%	102.64%	0.38	43.13%
ComplementNaïveBayes	0.094	0.311	63.09%	112.33%	0.369	42.17%
LADTree	0.13	0.252	85.19%	91.30%	0.318	37.50%
DecisionTable	0.137	0.257	89.38%	92.93%	0.3	35.80%
VFI	0.152	0.275	99.14%	99.52%	0.294	35.29%
NaïveBayesMultinomialUpdatable	0.147	0.268	96.38%	97.13%	0.26	32.14%
ClassificationViaClustering	0.117	0.343	77.04%	124.13%	0.232	29.52%
LWL	0.147	0.271	96.51%	98.01%	0.118	18.98%
HyperPipes	0.153	0.276	99.92%	99.92%	0.098	17.35%
OneR	0.14	0.374	91.60%	135.35%	0.084	16.03%
AdaBoostM1	0.148	0.272	97.10%	98.56%	0.068	14.54%
MultiBoostAB	0.148	0.272	97.10%	98.56%	0.068	14.54%
DecisionStump	0.148	0.272	97.10%	98.56%	0.068	14.54%
ConjunctiveRule	0.148	0.272	96.97%	98.50%	0.065	14.25%
DMNBtext	0.153	0.276	99.88%	99.89%	0.037	11.69%
ZeroR	0.153	0.276	100%	100%	0	8.33%

The performance of the LIBSVM classifier with RBF kernel and parameters C=500 and γ=0.5 compares well with the Lazy classifiers. Moreover, MultiClassClassifier with slightly lower identification accuracy rates runs multiclass problems with two-class classifiers by using one-versus-all classifications. However, for this classifier, the value of *RRSE* is dramatically increased to 98.43%. The identification accuracy rates of the Multilayer Perceptron, Rotation Forest, and End classifiers were dropped to 80%, 81%, and 82%, respectively, but for these classifiers, the error metric parameters were considerably lower than the MultiClassClassifier.

5.4.2.2 Performance comparison in classifying mobile device models based on unsupervised learning techniques

In Subsection 5.2.2.2 (c), DBSCAN and EM clustering algorithms grouped the majority of instances to their correct clusters. Based on this observation the study assumed that the entropy-MFCC feature set is capable of implementing source mobile device model identification through unsupervised learning techniques. Here, an alternative evaluation on unsupervised learning techniques is conducted, to validate this naïve assumption. Once again, the experiment was conducted based on the subset of the call recording dataset from DS3, in which the mobile devices were located in environment B, and the calls were recorded with Dell stationary. Table 5.24 entitles the performance comparison results determined in the use of all eight clustering algorithms in WEKA using the experimental setup. All clusterers were utilized with default parameters and settings unless stated otherwise. The cluster mode was selected using all dataset as the training data instances. For OPTICS clusterer, despite optimizing its two control parameters known as minimum neighbor distance of ε and minimum cluster size (min_{points}), zero cluster was generated. On the other hand, the DBSCAN clusterer was optimized by setting the control parameters (ε =1.205 and min_{points}=10) and generated a true number of clusters (12 clusters) including 1034 incorrectly clustered instances based

on the available mobile device models in the database. However, 180 instances out of 14400 instances were unclustered. The Cobweb clusterer generated 12 clusters with respect to each class and correctly assigned 8% of the training dataset to each cluster. The EM and *MakeDensityBasedClusterer* algorithm allow to insert the number of clusters beforehand and then determines incorrectly clustered instances, LL, and MDL metrics. Thus, for EM and *MakeDensityBasedClusterer* algorithm 3001 and 1231 instances were incorrectly clustered, respectively. Unfortunately, for both algorithms large MDL score indicates week clustering implementation for source mobile device model identification scenario. In overall, the DBSCAN assigned more instances to its correct cluster, which makes it the better choice.

Table 5.24: Comparison of the Performance Metrics Achieved with the Entropy-MFCC Feature

Clusterer	***ICI***	***UCI***	***LL***	***MDL***	***GC***
Cobweb	0	0	-	-	12
DBSCAN	**1034**	**180**	***NA***	***NA***	**12**
EM	3001	0	89.86	313.96	*NA*
FarthestFirst	10529	0	*NA*	*NA*	*NA*
FilteredClusterer	1701	0	*NA*	*NA*	*NA*
MakeDensityBasedClusterer	1231	0	85.21	318.6	*NA*
OPTICS	0	14400	*NA*	*NA*	0
SimpleKMeans	1701	0	*NA*	*NA*	*NA*

5.4.3 Robustness against Different Dataset

During the forensic investigation, sometimes the available resources are limited in terms of time and money. As a result, it is essential to discover the minimum length required for the call recording files and the number of devices to be considered to train a feature-based mobile device model structure for a specific model reliably. This experiment investigates the effect of applying the different number of data instances, devices and models for source mobile device model identification, as the classifier performance depends on the number of selected training and testing data instances within mobile devices, models or brands, the number of devices per mobile device model, and

the number of mobile device models and brands. The experiment was conducted based on the subset of the call recording dataset from DS3, in which the mobile devices were located in environment B, and the calls were recorded with Dell stationary.

5.4.3.1 Number of data instances

The experiment achieved the detection performances for increasing the number of data instances computed based on 98-D entropy-MFCC feature set through10-fold cross-validation, as shown in Figure 5.19. In order to compare the classification accuracy and robustness of the entropy-MFCC feature set along with the adopted classifier, the experiment was repeated for three different classifiers, including IB1, LIBSVM and rotation forest. These specific classifiers were selected because first the identification accuracies of these classifiers in Table 5.23 are between 80 to 100% and second these classifiers represent three important class of classifiers, including the nearest neighbor-based, kernel-based and decision trees based classifiers. It is clearly can be seen that the IB1 classifier always achieve the highest identification accuracy and the lowest error rates, and it shows minimal variations by increasing the number of data instances, whereas for LIBSVM classifier the identification accuracy follows the downward trend by increasing the number of data instances; meanwhile, the percentage of error rates for LIBSVM classifier considerably increased. Rotation forest classifier always determined the lower identification accuracy and higher error rates in compare to IB1 and LIBSVM classifier, whereas the performance significantly reduced by increasing the number of data instances. This suggests that the proposed source mobile device model identification module based on entropy-MFCC and high performing classifier is robust against increasing the number of data instances.

5.4.3.2 Training and Testing Percentage Split

For evaluation based on the percentage split, separating the dataset into training and test instances is an important part of evaluating a source mobile device model

identification module. Table 5.25 shows the performance comparison with respect to the different percentage of training and test data instances. For all three selected classifiers, the highest identification accuracy and the lowest error rates were achieved by using the 80% of the dataset for training and the remaining 20% for testing. In the overall for both cross-validation and percentage split experiments, it is plausible that increasing the size of the training dataset increases the training time; however, for all classifiers, this resulted in significant improvement in detection performance.

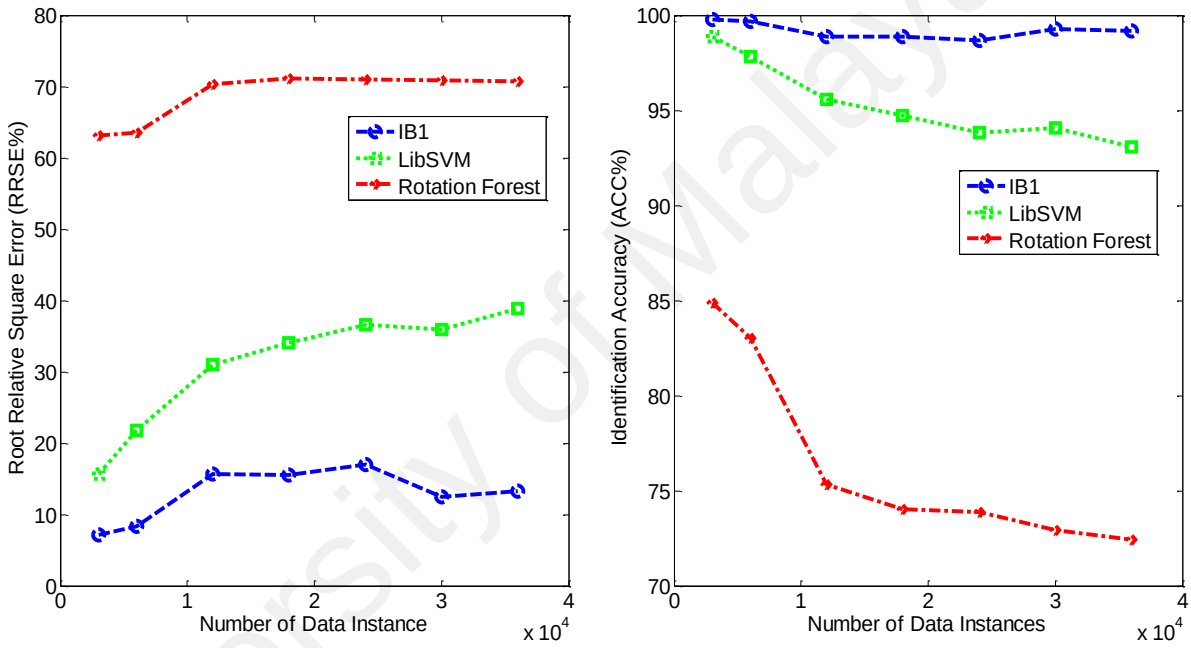


Figure 5.19: Detection Performance Variation of the Entropy-MFCCs with Increasing the Number of Data Instances

Table 5.25: Performance Comparison of Entropy-MFCC Features Based on Different Percentage Split with Respect to Training and Testing Dataset

Percentage Split (50-50)						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.007	0.08	4.23%	29.09%	0.958	96.12%
LIBSVM	0.01	0.102	6.83%	36.95%	0.932	93.74%
Rotation Forest	0.091	0.194	59.29%	70%	0.729	75.16%
Percentage Split (70-30)						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.003	0.050	1.65%	18.18%	0.984	98.49%
LIBSVM	0.006	0.078	4%	28.17%	0.960	96.36%
Rotation Forest	0.085	0.184	55.57%	66.72%	0.772	79.12%
Percentage Split (80-20)						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.002	0.044	1.24%	15.75%	0.988	98.86%
LIBSVM	0.005	0.072	3.43%	26.19%	0.966	96.86%
Rotation Forest	0.083	0.181	54.53%	65.6%	0.784	80.19%

5.4.3.3 Number of devices

The overall detection performance of source mobile device model identification for varying number of devices used with respect to each model are summarized in Table 5.26. The results of 10-fold cross-validation assessment through all three classifiers indicate that the performance reduces by increasing the number of devices utilized for each model. As discussed in Section 5.3.3, this is due to the existence of undesirable distances among entropy-MFCCs corresponding to mobile devices of the same model.

Table 5.26: Performance Comparison of Entropy-MFCC Features Based on Different Number of Available Devices for Each Model

2 device/model						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.001	0.029	0.55%	10.45%	0.995	99.5%
LIBSVM	0.001	0.033	0.73%	12.06%	0.993	99.33%
RotationForest	0.067	0.152	43.54%	54.87%	0.899	90.71%
4 device/model						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.0344	0.001	0.77%	12.43%	0.992	99.29%
LIBSVM	0.0028	0.053	1.86%	19.31%	0.981	98.29%
RotationForest	0.077	0.172	50.53%	62.29%	0.827	84.15%
6 device/model						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.001	0.036	0.86%	13.14%	0.991	99.21%
LIBSVM	0.004	0.065	2.74%	23.42%	0.973	97.49%
RotationForest	0.083	0.18	53.99%	65.05%	0.801	81.75%
8 device/model						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.001	0.037	0.89%	13.31%	0.991	99.19%
LIBSVM	0.006	0.076	3.73%	27.3%	0.963	96.58%
RotationForest	0.089	0.19	58.11%	67.77%	0.753	77.34%
10 device/model						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.002	0.044	1.24%	15.73%	0.988	98.87%
LIBSVM	0.007	0.086	4.8%	30.98%	0.952	95.6%
RotationForest	0.092	0.194	60%	70.3%	0.731	75.32%

5.4.3.4 Number of models

Comparing the results listed in Table 5.27 clearly shows a reduction in identification accuracy and an increase in error rates, each time a new set of mobile device models of the same series was utilized. Although sometimes considering the larger set of classes for

multi-class classification problems reduces the classification accuracy, it is unclear if the variation of the detection performances was due to increase the number of mobile device models or the possible hardware similarities between the utilized modes. The first group of four mobile device models in Table 5.27 (Group 1) is from the same series and manufacturer. Thus, it is expected that such mobile device models have the similar chipset, CPU, Wi-Fi and Cellular technology (Refer to Appendix B4). However, the mobile device models were perfectly detected with high identification accuracy of 99.28% and small error rates by using IB1 classifier.

Table 5.27: Performance Comparison of Entropy-MFCC Features Based on Different Number of Mobile Device Models

Group1: 4 models (i4, 4S, 5, 5S)						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.004	0.060	0.97%	13.9%	0.990	99.28%
LIBSVM	0.007	0.083	5.53%	33.27%	0.945	95.85%
RotationForest	0.214	0.290	56.95%	67.05%	0.766	82.48%
Group 2*: *4 models (i5, S3, S3M, Z)						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.004	0.061	1%	14.14%	0.99	99.25%
LIBSVM	0.012	0.111	3.27%	25.56%	0.967	97.55%
RotationForest	0.174	0.255	46.47%	58.81%	0.827	87%
Group 3*: *4 models (i5S, S4, N3, Z1)						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.001	0.032	0.27%	7.3 %	0.997	99.8 %
LIBSVM	0.005	0.068	1.23%	15.71%	0.988	99.08%
RotationForest	0.126	0.204	33.58%	47.19%	0.905	92.88%
Group 4*: *4 models (i4, S3, N4, C)						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.001	0.03	0.23%	6.83%	0.998	99.83%
LIBSVM	0.008	0.087	2%	20%	0.98	98.5%
RotationForest	0.147	0.23	39.22%	53.19%	0.866	89.98%
Group 5*: *7 models (i4, 4S, 5, 5S , S3M, S3, S4)						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.003	0.059	1.4%	16.73%	0.986	98.8%
LIBSVM	0.013	0.115	5.4%	32.86%	0.946	95.37%
RotationForest	0.148	0.247	60.35%	70.65 %	0.72	76%
Group 6*: *9 models (i4, 4S, 5, 5S , S3M, S3, S4, N3, N4)						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.002	0.045	1.03%	14.32%	0.99	99.1%
LIBSVM	0.009	0.094	4.49%	29.96%	0.955	96.01%
RotationForest	0.111	0.212	56.11%	67.29%	0.761	78.76%
Group 7*: *12 models (i4, 4S, 5, 5S , S3M, S3, S4, N3, N4, C, Z, Z1)						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.001	0.023	0.35%	8.33%	0.997	99.68%
LIBSVM	0.004	0.06	2.33	21.59%	0.977	97.86%
RotationForest	0.08	0.175	52.1%	63.31%	0.731	82.68%

Meanwhile, two different combinations for the group of four mobile device models from the same production year were selected for evaluation (Group 2 and 3). The results depict that the detection performance has increased due to fewer hardware similarities. This is because the majority of presented models were from different manufacturers. Further, it is evident that the identification accuracy of the Group 2 is lower than Group 3 and 4 because in the first group Samsung Galaxy S3 and S3 Mini are from the same series, whereas in the second group Samsung Galaxy S4 and Note 3 are from different series. In Group 4, the mobile devices are from different production year and different series, whereby its result is consistent with Group 3.

5.4.4 Evaluation of Different Influences on the Recording Process

As previously discussed in Section 2.4.3.1, there are a number of issues that make source communication device identification a challenging task. The majority of these issues arise from controlling the influences on the transmitting and receiving ends. The entropy-MFCC feature set was optimized in a way to detect the communication device response function despite the existence of convolutional transfer functions such as the speech signal, recording environment, communication type and recording stationary. The experiments in this section investigate the effect of different influences on the recording process such as environments, speakers, type of communication and stationary for source mobile device model identification.

5.4.4.1 Influences of the Speech

As discussed in Section 4.1.1.3, the DS3 dataset collected the speech recordings from two-sided VoIP and cellular conversations between a male who positioned at the stationary location and the female who picked up the mobile device at each location. The speech utterances were selected from English practice conversations and randomly picked from 20-25 two-sided dialog conversations. This study extracted the entropy-MFCC

feature set from the near-silent segments of the call recordings in order to eliminate the convolution produced by different speaker's transfer function, loudness and speech contents (Eq. 3.7). Table 5.28 reveals the result for source mobile device model identification by extracting the optimized entropy-MFCC feature set ([entropy-MFCC]$_{Shannon}$ + [entropy-MFCC]$_{Tsallis}$) from original speech recording signal. Further, results were compared against selected feature sets from source acquisition device identification literature, extracted from the speech recording signal in order to allow comparison. It is clearly can be seen that the performance of the entropy-MFCC feature set in regard to Shannon, Tsallis, and combined entropy is significantly lower than when was extracted from near-silent segments (Refer to Table 5.20). This suggests that the speech signal contaminates the feature values by influences of the speech other than the transmitting mobile device response function. Further, the experiment utilized both mean-MFCC and GSV-MFCC extracted based on the 13 default zero order MFCC coefficients from speech recordings, as commonly used in the literature (D. Garcia-Romero & Espy-Wilson, 2010; Hanilçi et al., 2012; Hanilci & Kinnunen, 2014). It is evident that the performance of both feature sets is considerably lower in compare to when extracted from near-silent segments, as shown in Table 5.22.

Table 5.28: Identification Accuracies for Selected Feature Sets over the Influence of the Speech by Using LIBSVM Classifier

Features	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
[entropy-MFCC]$_{Shannon}$ (13 default coefficients)	0.064	0.253	41.99%	91.64%	0.58	61.51%
[entropy-MFCC]$_{Tsallis}$ (13 default coefficients)	0.068	0.26	44.3%%	94.13%	0.557	59.39%
[entropy-MFCC]$_{Shannon}$ + [entropy-MFCC]$_{Tsallis}$ (n=48, M=49)	0.024	0.16	15.9%	56.39%	0.84	85.42%
mean-MFCC (13 default coefficients)	0.053	0.23	34.67%	83.2%	0.653	68.22%
mean-MFCC (n=48, M=49)	0.009	0.094	5.79%	34.04%	0.942	94.69%
GSV-MFCC (13 default coefficients)	0.025	0.16	16.65%	57.71%	0.833	84.74%
GSV-MFCC (n=48, M=49)	0.058	0.242	38.19%	87.4%	0.618	64.99%

Moreover, as indicated in Table 5.28, mean-MFCC feature set based on the optimized cepstral filterbanks surprisingly achieved the best performance for extracting the mobile device response function from the speech signal. However, due to the contamination of the speech signal, the performance of this feature set is also slightly lower than when was extracted from near-silent segments (Refer to Table 5.21).

5.4.4.2 Influences of the Mobile Device Environment

As the speech signal is propagated through the air medium, it is usually distorted by various environmental factors prior to arriving at the microphone. Section 3.1.3 discussed the influences of the recording environment on the call recording signal and presented these influences in terms of (Eq. 3.8). The experiment in this section was conducted to investigate the influences of the recording environment on the performance of the source mobile device model identification module. As discussed in Section 4.1.1.3, the DS3 dataset collected call recordings from mobile devices located in two indoor and two outdoor locations. Because all four environments were located inside the university campus during the timeline of one year, there was no control over the environmental sources, and the call recordings may have been collected during crowded or empty sessions. Hence, in order to train the classifier based on different environmental influences, at first, the experiment was conducted with call recordings corresponding to environment B and then each time the call recordings from the environments C, D, and E was added to the dataset. Table 5.29 reveals the results for source mobile device model identification based on call recordings from different environmental scenarios. It is clearly can be seen that combining the call recordings corresponding to calls that received from different environments and recorded with Dell stationary degrades the performance of the entropy-MFCC feature set in source mobile device model identification. This is because different environments have different response function that leaves different

influences on the call recording signal. As a result, the classifier may train the data instances based on the environment rather than the mobile device response function.

Table 5.29: Influences of Different Environments on Performance of the Entropy-MFCC Features for Source Mobile Device Model Identification

Environment B, C						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.011	0.105	7.21%	37.97%	0.928	93.39%
LIBSVM	0.02	0.142	13.17%	51.32%	0.868	87.93%
Rotation Forest	0.106	0.218	69.43%	78.71%	0.599	63.22%
Environment B, C, D						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.025	0.157	16.13%	56.81%	0.839	85.21%
LIBSVM	0.032	0.178	20.71%	64.36%	0.793	81.01%
Rotation Forest	0.114	0.23	74.9%	83.52%	0.5	54.16%
Environment B, C, D, E						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.033	0.181	21.38%	65.39%	0.786	80.4%
LIBSVM	0.039	0.196	25.26%	71.07%	0.747	76.85%
Rotation Forest	0.119	0.237	77.56%	85.75%	0.459	50.39%
Environment C, D, E						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.025	0.157	16.1%	56.75%	0.839	85.24%
LIBSVM	0.033	0.18	21.31%	65.28%	0.787	80.47%
Rotation Forest	0.115	0.231	75.16%	83.52%	0.508	54.88%
Environment D, E						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.015	0.123	9.95%	44.61%	0.901	90.88%
LIBSVM	0.024	0.156	15.98%	56.54%	0.84	85.35%
Rotation Forest	0.11	0.224	71.97%	80.85%	0.558	59.48%

5.4.4.3 Influences of the VoIP and Cellular Communications

The main difference between the VoIP and cellular network is that the first one converts the voice to IP packets and then converts the IP packets to RF signal in order to transmit the signal by using the WLAN or GSM data (E, 3G or 4G), whereas the former transmits the voice by using the RF signals through the cellular network. The generalized call recording process pipeline was illustrated in Figure 3.4. It is apparent that the transmitter, receiver and communication channel blocks might have different processes due to the aforementioned differences between VoIP and cellular wireless communication channels. The proposed source mobile device identification approach in this study is different with GSM mobile device identification approach proposed in (Hasse et al.,

2013). This is because the existing study utilized the spectral analysis techniques for estimating the mobile device response function by using both VoIP and cellular calls, whereas Hasse et al. (2013) employed RF fingerprints that are only applicable for detection of GSM devices. As discussed in Section 4.1.1.3, the experiment utilized the cellular calls received from environment B and recorded by using Nokia Lumia stationary and the VoIP calls received from the same environment and recorded with Dell stationary. Table 5.30 evaluates and compares the performance of the entropy-MFCC feature set for source mobile device model identification based on recorded VoIP calls against cellular calls and their combination. The results indicate that entropy-MFCC is perfectly capable of detecting the source mobile device model of the recorded cellular calls, whereby the identification accuracy and error rates compare well with the results for detecting the source mobile device model of the VoIP calls. In addition, because the recording signals corresponding to both VoIP and Cellular calls contain the mobile device response function, it is possible to combine the data instances from both VoIP and Cellular calls for source mobile device model identification.

Table 5.30: Source Mobile Device Model Identification for VoIP and Cellular Call Recordings by Using Entropy-MFCC Feature Set.

VoIP Calls						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.001	0.023	0.35%	8.33%	0.997	99.68%
LIBSVM	0.004	0.06	2.33	21.59%	0.977	97.86%
RotationForest	0.08	0.175	52.1%	63.31%	0.731	82.68%
Cellular Calls						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.006	0.078	3.98%	28.22%	0.96	96.35%
LIBSVM	0.003	0.057	2.1%	20.49%	0.979	98.08%
RotationForest	0.064	0.152	42.02%	54.84%	0.877	88.68%
VoIP and Cellular Calls						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.004	0.059	2.31%	21.5%	0.977	97.88%
LIBSVM	0.006	0.075	3.71%	27.24%	0.963	96.6%
RotationForest	0.081	0.179	52.98%	64.75%	0.785	80.25%

5.4.4.4 Influences of the Recording Stationary

As illustrated in the generalized call recording process pipeline illustrated in Figure 3.4, the RF front end of the stationary device captures the transmitted RF signal. In the proposed call recording setup in this study, the stationary device process and records the received signal. Hence, it is important to investigate the influences of the stationary device response function on the call recording signal. The experiments conducted in this section at first utilized the VoIP call recording signals recorded by the Dell stationary then added the VoIP calls recorded by the iMac stationary and the cellular calls recorded by the Nokia Lumia stationary to the dataset one at a time.

Table 5.31: Influences of Different Stationaries on Performance of the Entropy-MFCC Features for Source Mobile Device Model Identification

VoIP Calls, Dell Stationary						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.001	0.023	0.35%	8.33%	0.997	99.68%
LIBSVM	0.004	0.06	2.33	21.59%	0.977	97.86%
RotationForest	0.08	0.175	52.1%	63.31%	0.731	82.68%
VoIP Calls, iMac Stationary						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.008	0.091	5.36%	32.75%	0.946	95.08%
LIBSVM	0.015	0.121	9.6%	43.81%	0.904	91.21%
RotationForest	0.072	0.171	47.32%	61.74%	0.791	80.8%
VoIP calls, Dell and iMac Stationary						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.005	0.069	3.09%	24.86%	0.969	97.17%
LIBSVM	0.012	0.11	7.53%	38.79%	0.925	93.1%
RotationForest	0.082	0.183	53.64%	66.18%	0.759	77.91%
VoIP and Cellular calls, Dell, iMac and Nokia Lumia Stationary						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.005	0.072	3.39%	26.05%	0.966	96.89%
LIBSVM	0.009	0.095	5.94%	34.46%	0.941	94.56%
RotationForest	0.082	0.183	53.65%	66.1%	0.76	78.01%

Table 5.31 contains the results based on each call recording subset for source mobile device model identification by using entropy-MFCCs and three different classifiers. It is evident that the detection performance results corresponding to VoIP calls recorded with Dell stationary are in exceptionally good agreement with the results corresponding to VoIP calls recorded with iMac stationary. Although both iMac and Dell stationaries recorded VoIP calls, the differences are as following: (a) different operating systems (Mac

iOS and Windows), (b) different recording software (Pamela for Skype, VodBurner + Callnote), and (c) different audio recording formats ('.mp3' and '.wav'). Moreover, Nokia Lumia stationary utilized Windows Phone GSM call recorder and recorded calls in the stereo channel with '.mp4' format. Despite these differences, the performance of the source mobile device model identification remains robust against combining the data instances from different stationaries. Meanwhile, it is evident that the identification accuracies corresponding to the call recordings recorded with iMac stationary are lower compared to the Dell stationary. This is due to the lossy format of the '.mp3' files recorded with iMac stationary, whereas the call recording files in Dell stationary are in '.wav' format.

5.4.5 Robustness against Selected Post-Processing Operations

The classifier performance may reduce when the training dataset differs from the test dataset due to post-processing operations on the call recording signal. The experiment in this section utilized the VoIP call recording subset from DS3 corresponding to Dell stationary and environment B (recorded with '.wav' format), whereby the training data instances were generated from the original call recordings, and the test dataset was generated from the call recordings after the three different post-processing operations, including the splitting, MP3 conversion and denoising. To perform splitting, the experiment utilized audio editing software *Audacity* for splitting the call recording signal to two equal length files. The audio file of length 90 seconds was saved in '.wav' format and utilized for test dataset. For MP3 conversion, the test files were converted from '.wav' format to MP3 format and saved by using VLC Media Player. The denoising was performed by using the noise reduction effect provided by audio editing software *Audacity* with the noise reduction level of 12 dB. Noise Removal reduces constant background sounds such as hum, whistle, whine or buzz, and moderate amounts of 'hiss'. Table 5.32 summarizes the results for source mobile device model identification for call

recordings undergone further post-processing operations by using entropy-MFCCs. It appears that splitting has minimal influence on the performance of the entropy-MFCC feature set, whereas there is the considerable performance drop due to MP3 conversions due to the existence of lossy compression. Similarly, the denoising process slightly reduces the classification performance. It could be inferred that the post-processing operation introduces undesired signal variations in the call recording signal.

Table 5.32: Influences of Different Post-Processing Operations on Performance of the Entropy-MFCC Features for Source Mobile Device Model Identification

Original file						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0	0.018	0.2%	6.33%	0.998	99.82%
LIBSVM	0.002	0.046	1.38%	16.62%	0.986	98.73%
RotationForest	0.08	0.174	52.62%	63.09%	0.836	85%
Split file						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.001	0.026	0.46%	9.54%	0.996	99.58%
LIBSVM	0.002	0.047	1.44%	16.95%	0.986	98.68%
RotationForest	0.079	0.173	51.87%	62.68%	0.829	84.32%
Compressed file						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.041	0.175	0.26%	46.72%	0.892	91.89%
LIBSVM	0.032	0.178	8.49%	41.19%	0.915	93.64%
RotationForest	0.175	0.262	46.72%	60.43%	0.8	85.02%
Denoising						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
IB1	0.002	0.048	1.51%	17.41%	0.985	98.86
LIBSVM	0.008	0.087	5%	31.62%	0.95	96.25%
RotationForest	0.175	0.262	46.72%	60.44%	0.697	77.25%

5.4.6 Discussion

Prior work has focused on source acquisition device identification based on microphone recording in relation to both speech and non-speech signals. Hanilçi et al. (2012), for example, used cell phone devices as an ordinary tape recorder to collect speech recordings. Although these studies proved that MFCCs extracted from speech recordings are the most effective features to capture device response function, the results lack evaluation on the robustness of MFCCs. This is because MFCC features are contextualized by the speech contents, speaker's characteristics and the environment. In this study, we have optimized the robustness of MFCCs by redefining the Mel filterbank

spacing, increasing the number of zeroth order coefficients and computing the entropy of MFCCs from near-silent segments of the call recording signals for the prototype of the source mobile device model identification. This feature set exhibited lower performance to capture characteristics of the mobile device response function with the influences of the speech signal, environment, and channel distortion. Meanwhile, by eliminating the speech segments, we have proven the feasibility and suitability of entropy-MFCC features in capturing the device specific fingerprints from near-silent segments.

We found that by using all selected classifiers, entropy-MFCC feature set always exhibits high performance against mean-MFCC and GSV-MFCC feature set. Apart from that the combination of entropy with LFCC and BFCC features performed well for source mobile device model identification. These findings proved the significance of entropy on capturing the device specific information from the call recording signal. Furthermore, in terms of classifiers, nearest-neighbor based classifier (IB1) and SVM classifier (LIBSVM) achieved the best performance with respect to the classification accuracy and robustness, whereas LIBSVM revealed considerably larger error rates. In general, some aspects of the proposed method compare well with existing research on acquisition device identification (Buchholz et al., 2009; D. Garcia-Romero & Epsy-Wilson, 2010; Hanilçi et al., 2012; Kraetzer et al., 2007; Kraetzer et al., 2012; Kraetzer et al., 2011; Panagakis & Kotropoulos, 2012b). However, this study adds an advantage to the previous approaches in the following ways: (a) extracts the features from near-silent segments diminishes the need for collecting a set of call recordings with a variety of speakers who are different with the speakers used for collecting the test dataset, (b) rearranges the spacing in Mel filterbanks in order to capture the characteristics of the mobile device rather than perception of the audio signal, (c) utilizes both Shannon and Tsallis entropy to intensify the energy of MFCCs for near-silent frames, (d) allows intra- and inter-mobile device model identification from recorded VoIP and cellular calls.

5.4.7 Conclusion

The proposed scheme specifically identifies the model of the transmitting mobile devices from recorded VoIP and cellular calls for a forensic investigation. The proposed scheme has captured device specific influences and eliminated contaminating influences such as speech and environment through developing a control system model and mathematical modeling of the device response function. This shows entropy-MFCCs capture the mobile device response function from near-silent segments. The experiments in this section provide the framework for future studies to assess the performance for practical audio forensic cases.

5.5 Performance Evaluation-Phase IV: Individual Mobile Device Identification in Closed set using ZMBic

The set of comprehensive experiments in Phase III proved the feasibility of optimization techniques for extracting entropy-MFCC features from near-silent signals, particularly for source mobile device model identification using recorded calls. However, the proposed entropy-MFCC feature set is designed to minimize inter-model similarity and maximize intra-model similarity. Although the findings in Section 5.3.3 indicate that the entropy-MFCC feature set is capable of mapping each call recording to the specific mobile device used, this feature set revealed promising results for source mobile device model identification. This section employed ZMBic feature set for individual source mobile device identification because the intra-model similarity of the ZMBic features is lower than the entropy-MFCC features. This feature set was computed based on higher-order statistics to model the nonlinear characteristics of the mobile device response function as intrinsic mobile device fingerprints. Further, the Phase IV extends the evaluation study with respect to the optimized ZMBic feature set along with critical evaluation for individual source mobile device identification and also aims to satisfy the following objectives:

(a) To investigate the influence of applying different control parameters during the bicoherence extraction along with computing the ZMs of the bicoherence as it allows to optimize the ZMBic feature set in order to capture an intrinsic mobile device fingerprint from near-silent signals in compare to Hu moments of the bicoherence.
(b) To investigate the effect of applying different classification algorithms in the individual mobile device identification process by taking advantage of a set of classifiers implemented in Weka.
(c) To investigate the effect of applying the different number of data instances and devices for individual source mobile device identification, as the classifier

performance depends on the number of selected training and testing data instances within mobile devices and the number of devices per mobile device model, and the number of mobile device models and brands. Increasing the size of the training dataset increases the training time, whereby at some point this causes small or little improvement in classification accuracy. Moreover, sometimes considering the larger set of classes for multi-class classification problems reduces the classification accuracy.

(d) To investigate the effect of different influences on the recording process such as environments, speakers, type of communication and stationary for individual mobile device identification.

(e) To investigate the effect of applying selected post-processing operations on the call recording signal as the classifier performance may reduce when the training set differs from the test set due to such operations.

This section utilized the same setup by repeating the five experiments in Section 5.4, except with the fact that the experiments were conducted with the aim of individual source mobile device identification by using the ZMBic feature set. Furthermore, the experiments utilized the bicoherence magnitude and phase, whereby both the bicoherence magnitude and phase outputs are the square matrix with their size equal to the FFT length. Later, the algorithm generated a total of 28 ZMBic features by computing the ZMs of the bicoherence magnitude and 28 ZMBic features by computing the ZMs of the bicoherence phase. The following sections include the description of the experiments, results, and discussion.

5.5.1 Benchmarking Feature sets

The experiments in this section evaluate the feasibility of ZMBic feature set and the proposed optimization techniques for individual source mobile device identification

through comparison against state-of-the-art feature sets. At the beginning, this process justifies the choices made to compute the bicoherence magnitude and phase, and its corresponding ZMs during optimizing the ZMBic feature set for individual source mobile device identification. At this stage, first, the experiment controlled the parameter settings such as FFT length (n*fft*) and a number of samples per segment (*N*) to compute the bicoherence phase and magnitude. Second, the experiment evaluated and compared the performance of the different ZM polynomials during computing the ZMBic features. Eventually, the experiment compared the optimized ZMBic feature set against the state of the art feature sets for individual source mobile device identification. Furthermore, the experiment utilized the same subset of the DS3 and experimental setup as in Section 5.4.

5.5.1.1 Performance comparison in applying different control parameters during feature extraction

This experiment first optimized the ZMBic feature set based on the control parameters of the bicoherence extraction methods that discussed in Section 4.1.4. The control parameters include the FFT length (*nfft*), the number of samples per segment (*N*), type of time-domain window to be applied to each data segment, and percentage overlap. The dataset was lowered down to the subset of 10 individual mobile devices of Apple iPhone 4, located in environment B, whereby the calls were recorded with Dell stationary. The experiment selected the time-domain Hamming window with no overlap for bicoherence estimation, each time set the *nfft* value to 64, 128, 256, and 512 and increased the *N* value for each *nfft* setting, until $N \leq nfft / 2$. Subsequently, the ZMBic feature set was evaluated for each *nfft* and *N* parameter setting by using the LIBSVM classifier. The plot in Figure 5.20 suggests that for all four *nfft* settings, increasing the *N* value has dramatically reduced the identification accuracy. Hence, the best identification accuracy was achieved with *nfft* = 64, 128 and N=32. Meanwhile, in order to reduce the

computation time due to the unnecessary larger bicoherence spectrum, the optimal choice is *nfft* = 64 and *N*=32.

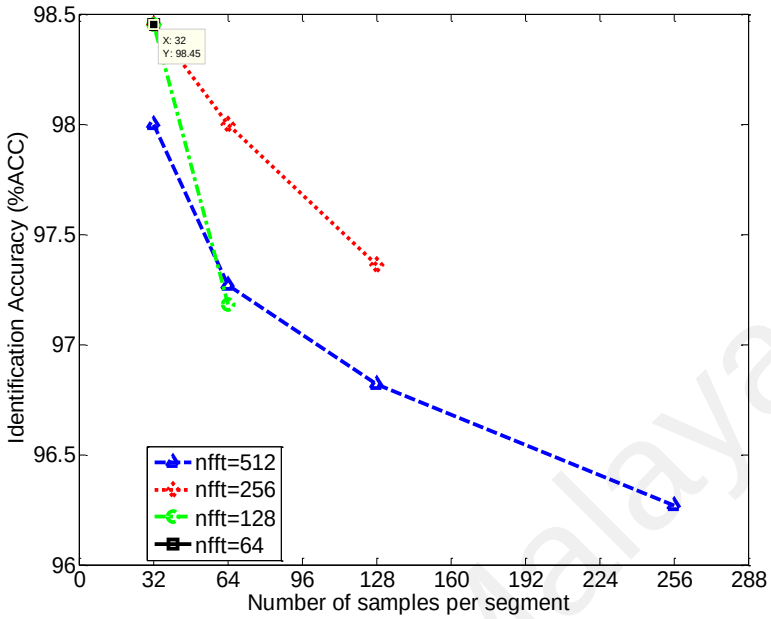


Figure 5.20: Identification Accuracies for Different FFT Length (*nfft*) and Number of Samples per Segment (*N*)

In the next step, the experiment investigated the feasibility of the different ZM polynomials to reduce the dimensionality of the bicoherence spectrum and at the same time extract the device specific information from the bicoherence spectrum. In order to optimize the proposed feature set with respect to the most fitting ZM polynomial, the experiment evaluated and compared the performance of the ZMBic feature set for different ZM polynomials with their corresponding polynomial orders. Because the bicoherence spectrum is the square matrix, the selected ZM polynomials should have been defined for the symmetric shapes such as circular, hexagonal or square orthogonal Zernike polynomials in order to be applicable to the surface data provided by the bicoherence spectrum. Table 5.33 lists the performance of the individual source mobile device identification by using LIBSVM classifier among 10 Apple iPhone 4 devices for different ZM polynomials during ZMBic extraction. The results suggest that the Hexagon rotated 30-degree polynomial is the optimal choice. This is because, this polynomial reduces the dimensionality of the bicoherence spectrum to the 28-Dimensional feature

vector, and at the same time, it achieves the highest identification accuracy and Kappa statistic and the lowest error rates. In Section 4.1.4.2, the 30-degree hexagonal polynomial was selected to compute the ZMs of the bicoherence spectrum.

Table 5.33: Identification Accuracies for Different ZM Polynomials

ZM polynomials	*Polynomial order*	*MAE*	*RMSE*	*RAE*	*RRSE*	*Kappa Statistic*	*ACC*
Fringe (circular)	j=1…37	0.003	0.056	1.72%	18.53%	0.983	98.46%
Square	j=1…45	0.004	0.059	1.92%	19.59%	0.981	98.27%
Hexagon	j=1…45	0.003	0.052	1.52%	17.41%	0.985	98.64 %
Hexagon Rotated 30 Degree	j=1…28	0.003	0.051	1.41%	16.82%	0.986	**98.73%**

5.5.1.2 Performance comparison in classifying mobile device units based on state-of-the-art feature sets

As discussed in Section 5.3, an optimal feature set for individual source mobile device identification should be able to perfectly discriminate among mobile device units and not among mobile device models. Based on this strategy, this subsection will analyze and compare the performance of the $ZMBic_M$, $ZMBic_{Ph}$ and its combination against Hu moments, entropy and mean of the bicoherence magnitude and phase by using the LIBSVM classifier with RBF kernel, and optimized C and γ parameters, in order to justify its selection over other state of the art normalization techniques. This evaluation utilized the call recordings from the small subset of 10 individual Apple iPhone 4 devices corresponding to environment B and Dell stationary. Table 5.34 reveals the best performance for $ZMBic_M$, whereby it is evident that combining the $ZMBic_M$ with the $ZMBic_{Ph}$ feature set has a minimal contribution to improving the identification accuracy and reducing the error rates. Moreover, the results suggest the poor performance of the statistical normalization techniques such as entropy and mean over geometrical moments such as Zernike and Hu moments.

Subsequently, Table 5.35 reveals the overall performance of the ZMBic feature set, in compare to HuMBic feature set, the conventional mean-MFCC, and GSV-MFCC feature

sets as well as the optimized entropy-MFCC, mean-MFCC and GSV-MFCC feature sets for individual source mobile device identification by using the LIBSVM classifier. The experiment utilized call recordings from 120 mobile devices corresponding to environment B and Dell stationary. It appears that the performance of the optimized entropy-MFCC, mean-MFCC and GSV-MFCC feature sets extracted from near silent segments of the call recording signal compares well with $ZMBic_M$ and $ZMBic_M$+ $ZMBic_{Ph}$ feature sets. Malik and Miller (2012) utilized the distance between scale-invariant Hu moments of the bicoherence magnitude spectrum and the cross-correlation between the bicoherence phase spectra for automatic microphone identification by using the hypothesis testing. However, it appears that combining the $HuMBic_M$ with $HuMBic_{Ph}$ improves the overall performance of the feature set for all selected classifiers. Further, the results indicate that the optimized entropy-MFCC feature set through cepstral analysis techniques (Refer to Section 5.4), is perfectly capable of discriminating among mobile devices of the same model. It should, however, be noted that in audio acquisition device identification literature (Garcia-Romero & Espy-Wilson, 2010; Hanilçi et al., 2012; Hanilci & Kinnunen, 2014), the mean-MFCC and GSV-MFCC feature sets were utilized in their conventional forms, in which their performance is considerably lower than the optimized acoustic features in this study.

Table 5.34: Performance Evaluations Based on Different Statistical and Geometrical Moments of the Bicoherence Magnitude and Phase

Features	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
$ZMBic_M$	0.003	0.051	1.41%	16.82%	0.986	**98.73%**
$ZMBic_{Ph}$	0.009	0.092	4.75%	30.81%	0.953	95.73%
$ZMBic_M$+ $ZMBic_{Ph}$	0.003	0.003	1.62%	17.98%	0.984	98.55%
$HuMBic_M$	0.005	0.073	2.93%	24.21%	0.971	**97.36%**
$HuMBic_{Ph}$	0.022	0.147	11.92%	48.83%	0.881	89.27%
$[Entropy\text{-}Bic_M]_{Shannon}$	0.114	0.337	63.23%	112.46%	0.368	43.09%
$[Entropy\text{-}Bic_M]_{Tsallis}$	0.052	0.228	28.99%	76.15%	0.710	73.91%
$[Entropy\text{-}Bic_{Ph}]_{Shannon}$	0.032	0.18	17.88%	59.8%	0.821	83.91%
$[Entropy\text{-}Bic_{Ph}]_{Tsallis}$	0.052	0.228	28.99%	76.15%	0.710	73.91%
$Mean\text{-}Bic_M$	0.104	0.322	57.68%	107.40%	0.423	48.09%
$Mean\text{-}Bic_{Ph}$	0.058	0.241	32.22%	80.28%	0.678	71%

Table 5.35: Performance Evaluations based on ZMBic Feature Set against Selected State-of-the-Art Feature Sets

Features	*MAE*	*RMSE*	*RAE*	*RRSE*	*Kappa Statistic*	*ACC*
$ZMBic_M$	0	0.019	2.1%	20.5%	0.979	97.92%
$ZMBic_{Ph}$	0.003	0.058	20.49%	64%	0.795	79.68%
$ZMBic_M$+ $ZMBic_{Ph}$	0	0.018	2.17%	20.84%	0.978	98.05%
$HuMBic_M$	0.003	0.051	15.9%	56.39%	0.841	84.24%
$HuMBic_{Ph}$	0.008	0.087	45.25%	95.13%	0.548	55.13%
$HuMBic_M$+ $HuMBic_{Ph}$	0	0.025	3.85%	27.75%	0.962	96.18%
$[entropy\text{-}MFCC]_{Shannon}$ + $[entropy\text{-}MFCC]_{Tsallis}$	0	0.016	1.58%	17.78%	0.984	98.43%
mean-MFCC (13 default coefficients)	0.002	0.044	11.45%	45.76%	0.886	88.64%
mean-MFCC (n=48, M=49)	0	0.011	0.68%	11.66%	0.993	99.33%
GSV-MFCC (13 default coefficients)	0.005	0.069	28.82%	75.92%	0.712	71.42%
GSV-MFCC (n=48, M=49)	0	0.0123	0.92%	13.54%	0.991	99.1%

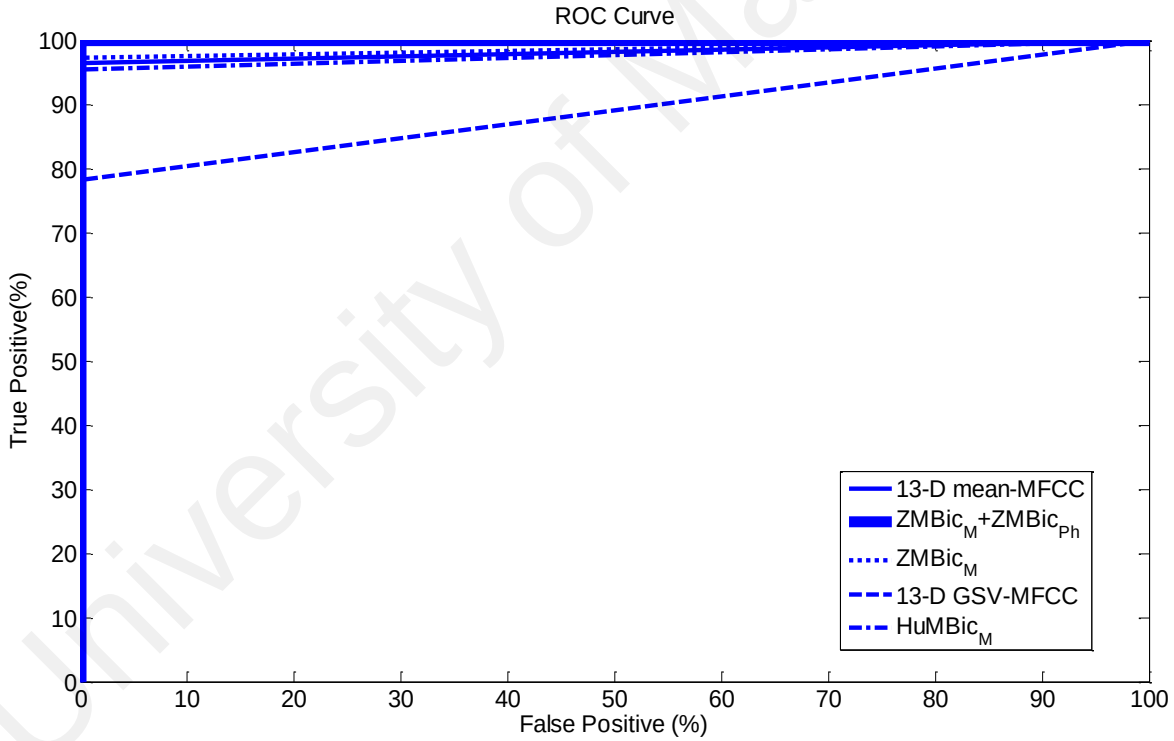


Figure 5.21: Overall ROC Curve of LIBSVM Classifier Using Different Feature Sets on the Class of Labels Based on Individual Mobile Devices

In addition, Figure 5.21 compares the overall ROC curves of the LIBSVM classifier among all feature sets and label class for individual source mobile device identification. The ROC area for the combined $ZMBic_M$ and $ZMBic_{Ph}$ features was close to one, but the value was smaller for the state-of-the-art feature sets such as $HuMBic_M$, mean-MFCC,

and GSV-MFCC feature sets. This finding indicates that for the ZMBic feature set, the false positive rate is close to zero, and the true positive rate is close to one. Moreover, the ROC area for the 48-D entropy-MFCC features for individual source mobile device identification compares well with the combined $ZMBic_M$ and $ZMBic_{Ph}$ feature set.

5.5.2 Benchmarking Classifiers

This section provides a summarizing review of the detection performance of existing classification algorithms on the individual source mobile device identification approach proposed in this study. The evaluation results depend on the performance of the classification algorithms that implemented in the data mining tool WEKA (v.3.6.1).

5.5.2.1 Performance comparison in classifying mobile device models based on supervised learning techniques

The large number of 74 supervised learning algorithms were implemented in WEKA (v.3.6.10). Hence, this section aims to perform the validation experiment by using all 66 applicable classifiers, in order to determine the most suitable subset of the classifiers for individual source mobile device identification. The best performing classifier subset could be used as a point of reference for the individual source mobile device identification evaluations within this study. The experiment was conducted based on the subset of the call recording dataset from DS3, in which 10 Apple iPhone 4 devices were located in environment B, and the calls were recorded with Dell stationary. The experimental setup utilized 10-fold cross validation and a total of 1200 feature vectors (corresponding to 10 mobile devices times 120 data instances generated from each call recording file), whereby the dimensionality of the feature vector of $ZMBic_M$ is 28. All classifiers were utilized with default parameters and settings unless stated otherwise.

The experiment results based on the performance metrics shown in Table 5.36 reveal the strong variation in the detection performance of the aforementioned classifiers. Hence,

the classification algorithms are sorted in this table with respect to the identification accuracy rates and error metrics, from high performing to the low performing classifiers. It is evident that among a total of 66 classifiers tested in this experiment, 12 classifiers achieved the high identification accuracy of between 98-99%, 41 classifiers obtained the identification accuracy of between 92-97.91%, 3 classifiers determined promising performance with identification accuracy of between 70-90%, and the remaining 10 classifiers achieved the low identification accuracy of less than 23%. The MultiClassClassifier achieved the highest identification accuracy of 99% and Kappa Statistics of 0.989; however, it determined significantly high RRSE and RAE of above 97%. Based on this observation, the five best performing classifiers with identification accuracy in the range of 98% < *ACC* < 98.45% and the error rates in the range of 17% < RRSE < 90% were selected. The selected classifiers were MultilayerPerceptron (NN) with ACC of 98.45%, Logistic (linear logistic regression) with ACC of 98.45%, IBk (nearest-neighbor instance based learning) with ACC of 98.45%, Rotation Forest (ensemble of decision trees) with ACC of 98% and sequential minimal optimization (SMO) (trains a support vector classifier) with ACC of 98%.

Because Stacking, Vote, and MultiScheme classifiers utilize the baseline methods (i.e. LIBSVM and IB1) for combining the classifiers and the decisions were made by averaging the probability estimates, they were excluded from the top classifiers list. The experiments in the subsequent sections utilized the selected top classifiers for performance evaluation and comparison.

Table 5.36: Performance Comparison of ZMBic Features Based on Different Classification Algorithms

Classifier	*MAE*	*RMSE*	*RAE*	*RRSE*	*Kappa Statistic*	*ACC*
MultiClassClassifier	0.176	0.293	97.59%	97.59%	0.989	99%
StackingC	0.005	0.048	2.79%	16.07%	0.985	98.64%
Vote	0.006	0.048	3.04%	15.99%	0.984	98.55%
Stacking	0.003	0.056	1.72%	18.53%	0.983	98.46%
MultilayerPerceptron	0.006	0.05	3.08%	16.52%	0.983	98.45%
Logistic	0.003	0.053	1.68%	17.79%	0.983	98.45%
IBk	0.005	0.055	2.70%	18.47%	0.983	98.45%
IB1	0.003	0.056	1.72%	18.53%	0.983	98.45%
MultiScheme	0.007	0.055	3.62%	18.41%	0.981	98.27%
RotationForest	0.01	0.061	5.53%	20.38%	0.978	98%
KStar	0.004	0.063	2.23%	20.89%	0.978	98%
SMO	0.16	0.272	89.08%	90.78%	0.978	98%
LMT	0.006	0.056	3.03%	18.68%	0.977	97.91%
SimpleLogistic	0.006	0.055	3.25%	18.42%	0.976	97.82%
RandomForest	0.011	0.067	6.33%	22.44%	0.972	97.46%
FT	0.006	0.065	3.12%	21.54%	0.972	97.45%
RandomComittee	0.01	0.064	5.25%	21.23%	0.97	97.27%
LIBSVM	0.006	0.076	3.23%	25.43%	0.968	97.09%
CVPrameterSelection	0.006	0.076	3.23%	25.43%	0.968	97.09%
Grading	0.006	0.076	3.23%	25.43%	0.968	97.09%
End	0.013	0.068	7.05%	22.59%	0.967	97%
Bagging	0.014	0.071	7.87%	23.63%	0.967	97%
Dagging	0.161	0.272	89.17%	90.72%	0.967	97%
Decorate	0.015	0.073	8.14%	24.25%	0.964	96.73%
NaïveBayesMultinomial	0.111	0.2	61.49%	66.75%	0.963	96.64%
LogitBoost	0.009	0.072	4.70%	23.85%	0.962	96.55%
RandomSubSpace	0.017	0.073	9.51%	24.28%	0.962	96.55%
ClassificationViaRegression	0.017	0.077	9.58%	25.73%	0.961	96.45%
LWL	0.149	0.263	82.73%	87.73%	0.96	96.36%
NaïveBayesSimple	0.007	0.085	4.04%	28.17%	0.959	96.36%
NaïveBayes	0.007	0.086	4.13%	28.52%	0.959	96.27%
NaïveBayesUpdatable	0.007	0.086	4.13%	28.52%	0.959	96.27%
ComplementNaïveBayes	0.008	0.087	4.24%	29.13%	0.958	96.18%
NNge	0.008	0.087	4.24%	29.13%	0.958	96.18%
AttributeSelectedClassifier	0.009	0.083	4.73%	27.49%	0.957	96.09%
J48	0.009	0.086	5.00%	28.69%	0.954	95.82%
J48graft	0.009	0.089	5.00%	29.78%	0.954	95.82%
RBFNetwork	0.009	0.078	5.10%	26.04%	0.952	95.64%
NBTree	0.01	0.088	5.77%	29.31%	0.952	95.64%
BayesNet	0.009	0.089	4.86%	29.59%	0.952	95.64%
REPTree	0.015	0.093	8.24%	31.07%	0.95	95.46%
RandomTree	0.01	0.097	5.25%	32.41%	0.948	95.27%
LADTree	0.015	0.083	8.47%	27.65%	0.947	95.18%
VFI	0.124	0.214	69.02%	71.35%	0.947	95.18%
DTNB	0.01	0.089	5.73%	29.54%	0.946	95.09%
PART	0.01	0.096	5.76%	31.91%	0.946	95.09%
nestedDichotomies.ND	0.012	0.097	6.69%	32.39%	0.944	95%
BFTree	0.012	0.095	6.48%	31.79%	0.943	94.91%
nestedDichotomies.ClassBalancedND	0.012	0.098	6.55%	32.79%	0.941	94.72%
SimpleCart	0.014	0.1	7.60%	33.38%	0.94	94.64%
Ridor	0.012	0.107	6.36%	35.68%	0.936	94.27%
nestedDichotomies.DataNearBalancedND	0.014	0.106	7.55%	35.42%	0.932	93.91%
OrdinalclassClassifier	0.017	0.101	9.54%	33.69%	0.928	93.55%
FilteredClassifier	0.017	0.116	9.36%	38.79%	0.916	92.45%
DecisionTable	0.049	0.125	27.24%	41.69%	0.887	89.82%
OneR	0.023	0.151	12.63%	50.25%	0.874	88.64%
ClassificationViaClustering	0.055	0.234	30.37%	77.93%	0.697	72.55%
HyperPipes	0.176	0.295	97.74%	98.16%	0.128	21.55%
NaïveBayesMultinomialUpdatable	0.136	0.277	75.31%	92.26%	0.115	20.36%
ConjunctiveRule	0.161	0.284	89.46%	94.64%	0.109	19.82%
DecisionStump	0.161	0.284	89.66%	94.76%	0.108	19.73%
AdaBoostM1	0.161	0.284	89.66%	94.77%	0.108	19.73%
MultiBoostAB	0.161	0.284	89.66%	94.77%	0.108	19.73%
RacedIncrementLogitBoost	0.18	0.3	100%	100%	0	10%
ZeroR	0.18	0.3	100%	100%	0	10%
DMNBtext	0.18	0.3	100.01%	100.02%	-0.005	9.55%

5.5.2.2 Performance comparison in classifying mobile device models based on unsupervised learning techniques

This experiment was conducted based on the call recordings collected from 10 individual Apple iPhone 4 devices from DS3 dataset for source individual mobile device identification, in which the mobile devices were located in environment B, and the calls were recorded with Dell stationary. All clusterers were utilized with default parameters and settings unless stated otherwise. Table 5.37 entitles the detection performance for individual devices determined in the use of all of WEKAs eight clustering algorithms using this experimental setup. The cluster mode was selected using all dataset as the training data instances. For OPTICS clusterer, despite optimizing its two control parameters known as minimum neighbor distance of ε and minimum cluster size (min_{points}), zero cluster was generated. On the other hand, the DBSCAN clusterer after optimizing the control parameters ε and min_{points}, generated a true number of clusters with a total of 10 clusters in relation to the total mobile device models with zero incorrectly clustered instance. However, 26 instances out of 1200 instances were unclustered. The Cobweb clusterer perfectly distributed 8% of the training dataset to each cluster. For EM and *MakeDensityBasedClusterer* algorithm 239 and 199 instances were incorrectly clustered, respectively. Furthermore, for both algorithms, smaller MDL scores indicate strong clustering implementation for individual source mobile device identification scenario. In overall, the DBSCAN assigned more instances to its correct cluster, which makes it the better choice.

5.5.3 Robustness against Different Dataset

The experiments in this section reevaluated the experiments in Section 5.5.3, in order to investigate the effect of applying the different number of data instances, devices and models for individual source mobile device identification. The experiment was conducted

based on the subset of the call recording dataset from DS3, in which the mobile devices were located in environment B, and the calls were recorded with Dell stationary.

Table 5.37: Performance Evaluation based on Different Clustering Algorithms

Clusterer	*ICI*	*UCI*	*LL*	*MDL*	*GC*
Cobweb	0	0	*NA*	*NA*	10
DBSCAN	0	26	*NA*	*NA*	10
EM	239	0	71.05	14.11	*NA*
FarthestFirst	839	0	*NA*	*NA*	*NA*
FilteredClusterer	207	0	*NA*	*NA*	*NA*
HierarchicalClusterer	745	0	*NA*	*NA*	*NA*
MakeDensityBasedClusterer	199	0	48.28	36.88	*NA*
OPTICS	0	1200	*NA*	*NA*	0
SimpleKMeans	207	0	*NA*	*NA*	*NA*

5.5.3.1 Number of data instances

The experiment evaluated the performances for increasing the number of data instances, computed by using 28-D $ZMBic_M$ feature set and evaluated through10-fold cross validation. Figure 5.22 summarizes the results, in which for comparison the experiment repeated the evaluation for five best performing classifiers, including Multilayer Perceptron, Logistic, IBk, Rotation Forest and SMO, as suggested in Section 5.5.2. It is clearly can be seen that the Multilayer Perceptron classifier in the majority of cases achieve the highest identification accuracy and the lowest error rates, whereby it improves slowly by increasing the number of data instances. For SMO classifier, the identification accuracy follows the upward trend by increasing the number of data instances; meanwhile, the percentage of error rates for SMO classifier is considerably large and slightly reduced by increasing the data instances. Rotation forest classifier frequently determined the lower identification accuracy and similar error rates in compare to Logistic and IBk classifiers, whereas for all classifiers the overall performance considerably increased by increasing the number of data instances. This suggests that the proposed individual source mobile device identification module based on ZMBic and high performing classifier is robust against increasing the number of data instances.

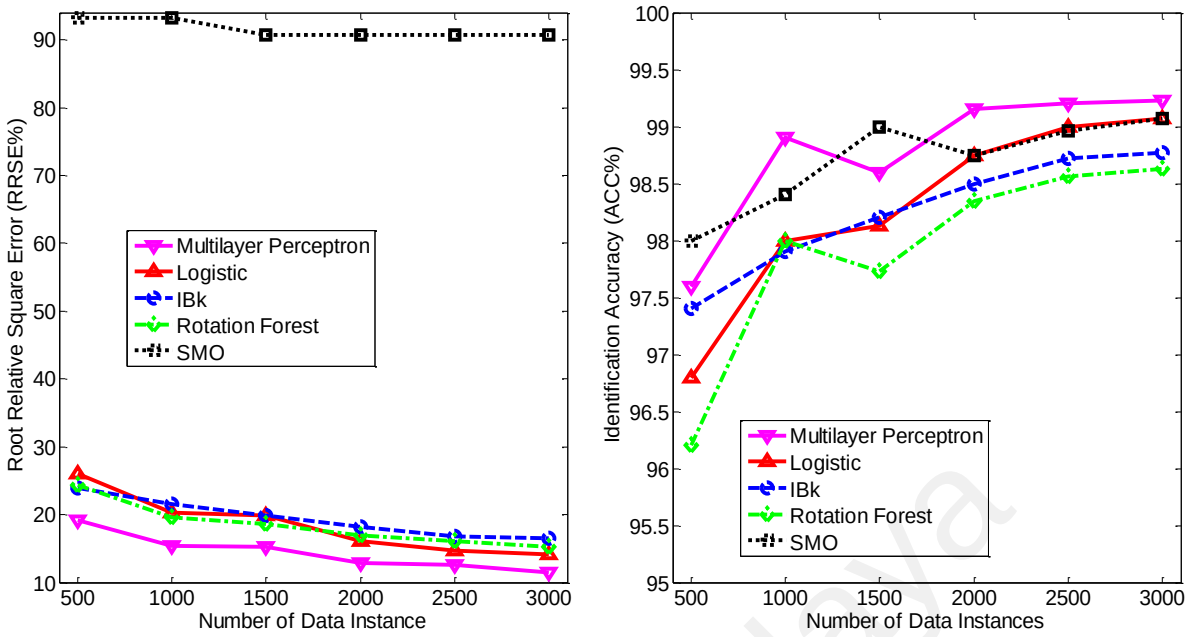


Figure 5.22: Performance Evaluation of the ZMBic$_M$ for Increasing Number of Data Instances

5.5.3.2 Training and Testing Percentage Split

The experiment evaluated the individual mobile device identification performance for the different percentage of training and test data instances. Table 5.38 shows the results, in which it is clearly can be seen that by using the 80% of the dataset for training and the remaining 20% for testing, the majority of classifiers achieved their best performances with highest identification accuracy and the lowest error rates. However, the classifiers' performance considerably dropped by splitting the dataset to half, corresponding to training and testing dataset. It could be inferred therefore that for both cross-validation and percentage split experiments, increasing the size of the training dataset increases the training time; however, for all classifiers, this resulted in significant improvement in detection performances.

5.5.4 Evaluation of Different Influences on the Recording Process

In Section 3.2.3, the study suggested modeling mobile device response function through analysis of near-silent segments of the call recording signal using higher-order cumulant spectra. The ZMBic feature set employs HOSA to model the mobile device

response function in which there are some advantages. The HOSA techniques: (a) suppresses the additive Gaussian noise, (b) reconstructs the true phase and magnitude response of signals, and (c) detects the mobile device nonlinear response in call recording signal. Thus, it is expected that the influences such as speech, environments, communication channel and the stationary device are minimized. The experiments in this section were conducted in order to prove the robustness of the ZMBic feature set against such influences for individual source mobile device identification.

Table 5.38: Performance Evaluation Based on Different Percentage Split with Respect to Training and Testing Dataset

Percentage Split (70-30)						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
MultilayerPerceptron	0.01	0.07	5.3%	23.22%	0.966	96.97%
Logistic	0.005	0.067	2.83 %	22.43%	0.973	**97.58%**
IBk	0.008	0.074	4.27%	24.48%	0.973	97.27%
RotationForest	0.016	0.08	8.98%	26.61%	0.96	96.36%
SMO	0.161	0.273	89.1%	90.79%	0.973	**97.58%**
Percentage Split (50-50)						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
MultilayerPerceptron	0.011	0.075	5.97%	24.86%	0.96	96.36%
Logistic	0.008	0.081	4.23%	27.04%	0.958	96.18%
IBk	0.01	0.08	5.35%	26.75%	0.985	96.73%
RotationForest	0.017	0.083	9.37%	27.69%	0.947	95.27%
SMO	0.161	0.273	89.13%	90.86%	0.962	96.55%
Percentage Split (80-20)						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
MultilayerPerceptron	0.007	0.064	4%	21.39%	0.975	**97.73%**
Logistic	0.006	0.07	3.04%	23.39%	0.97	97.27%
IBk	0.006	0.06	3.12%	20%	0.98	**98.18%**
RotationForest	0.012	0.069	6.65%	23.11%	0.97	**97.27%**
SMO	0.16	0.272	89.06%	90.73%	0.97	97.27%

5.5.4.1 Number of devices

The experiment repeated the individual source mobile device identification based on ZMBic features for increasing number of classes regarding individual mobile devices for each model. The overall detection performance for each mobile device model and class population is detailed in Table 5.39. The results of 10-fold cross-validation assessment through NN classifier (Multilayer Perceptron classifier implemented in data mining tool WEKA) indicate that for the majority of models the classifier's performance only slightly reduces by increasing the number of devices. As discussed in Section 5.3.3, this is due to

the existence of significant distances among ZMBic features corresponding to the mobile devices of the same model.

Table 5.39: Performance Evaluation for Increasing Number of Classes Based on Individual Devices per Model

4 Mobile Devices						
Model	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistics***	***ACC***
i4	0.009	0.065	2.38%	14.93%	0.988	99.09%
i4S	0.01	0.076	2.59%	17.43%	0.982	98.64%
i5	0.009	0.079	2.45%	18.31%	0.982	98.64%
i5S	0.009	0.066	2.33%	15.23%	0.985	98.86%
S3M	0.01	0.078	2.63%	18.05%	0.982	98.64%
S3	0.01	0.08	2.78%	18.44%	0.976	98.18%
S4	0.012	0.091	3.19%	29.94%	0.973	97.96%
N3	0.013	0.093	3.38%	21.49%	0.973	97.96%
N4	0.012	0.086	3.16%	19.83%	0.976	98.18%
C	0.011	0.078	2.95%	18.07%	0.979	98.41%
Z	0.008	0.064	2.19%	14.81%	0.988	99.09%
Z1	0.009	0.071	2.44%	16.48%	0.985	98.86%
6 Mobile Devices						
Model	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistics***	***ACC***
i4	0.007	0.058	2.55%	15.54%	0.988	98.94%
i4S	0.007	0.062	2.67%	16.75%	0.984	98.64%
i5	0.008	0.064	2.69%	17.07%	0.982	98.49%
i5S	0.007	0.062	2.67%	16.65%	0.982	98.49%
S3M	0.009	0.072	3.2%	19.18%	0.98	98.33%
S3	0.007	0.057	2.43%	15.41%	0.986	98.79%
S4	0.008	0.063	2.78%	16.78%	0.982	98.49%
N3	0.008	0.066	2.8%	17.69%	0.984	98.64%
N4	0.008	0.066	2.94%	17.7%	0.978	98.18%
C	0.007	0.055	2.4%	14.69%	0.984	98.64%
Z	0.008	0.066	2.92%	17.82%	0.982	98.49%
Z1	0.009	0.075	3.38%	20.02%	0.976	98.03%
8 Mobile Devices						
Model	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistics***	***ACC***
i4	0.006	0.057	2.91%	17.1%	0.982	98.41%
i4S	0.008	0.064	3.43%	19.22%	0.975	97.84%
i5	0.006	0.058	2.87%	17.61%	0.982	98.41%
i5S	0.006	0.057	2.89%	17.11%	0.981	98.3%
S3M	0.008	0.068	3.56%	20.62%	0.973	97.61%
S3	0.006	0.05	2.49%	15.16%	0.987	98.86%
S4	0.006	0.05	2.58%	15.24%	0.986	98.75%
N3	0.006	0.056	2.74%	16.86%	0.984	98.64%
N4	0.006	0.054	2.75%	16.26%	0.983	98.52%
C	0.007	0.057	3.06%	17.25%	0.979	98.18%
Z	0.006	0.05	2.61%	15.11%	0.986	98.75%
Z1	0.008	0.066	3.5%	19.8%	0.978	98.07%
10 Mobile Devices						
Model	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistics***	***ACC***
i4	0.006	0.05	3.08%	16.52%	0.983	98.46%
i4S	0.006	0.054	3.24%	17.88%	0.978	98%
i5	0.006	0.056	3.3%	18.67%	0.979	98.09%
i5S	0.006	0.061	3.53%	20.26%	0.976	97.82%
S3M	0.007	0.066	4.08%	21.95%	0.971	97.36%
S3	0.005	0.046	2.68%	15.4%	0.986	98.73%
S4	0.004	0.041	2.35%	13.63%	0.988	98.91%
N3	0.006	0.053	3.09%	17.78%	0.982	98.36%
N4	0.006	0.051	3.07%	16.89%	0.981	98.27%
C	0.005	0.046	2.66%	15.47%	0.985	98.64%
Z	0.006	0.049	3.03%	16.39%	0.983	98.46%
Z1	0.006	0.056	3.51%	19.22%	0.976	97.82%

5.5.4.2 Influences of the Speech

The experiment utilized call recordings from 120 mobile devices corresponding to environment B and Dell stationary. Subsequently, the original speech recording signal is used to determine the performance of the individual source mobile device identification based on ZMBic features and compared its performance against state-of-the-art feature sets from source acquisition device identification literature, as shown in Table 5.40. It is clearly can be seen that the $ZMBic_M$ and $ZMBic_{Ph}$ feature sets in addition to its combined set achieved the best performance for extracting the mobile device response function from the speech signal. This suggests that the ZMBic feature set is robust against the influences of the speech, and it's perfectly capable of capturing characteristics of nonlinearities induced by the mobile device nonlinear system. Moreover, by extracting the features from near-silent segments the framework eliminates parts of the signal which also contains mobile device specific information. As a result, the overall performance of the bispectrum-based features is slightly larger than when was extracted from near-silent segments (Refer to Table 5.35). Further, the experiment utilized cepstrum-based features such as mean-MFCC and GSV-MFCC extracted based on the 13 default zero order MFCC coefficients from speech recordings, as commonly used in the literature (D. Garcia-Romero & Espy-Wilson, 2010; Hanilçi et al., 2012; Hanilci & Kinnunen, 2014) for individual source mobile device identification. Meanwhile, the optimized cepstrum-based feature sets from Section 5.5, such as $[\text{entropy-MFCC}]_{\text{Shannon}}$ + $[\text{entropy-MFCC}]_{\text{Tsallis}}$, mean-MFCC, and GSV-MFCC were also evaluated for individual source mobile device identification. It is evident that the performances of the both conventional and optimized mean-MFCC and GSV-MFCC feature sets were considerably reduced in compare to when were extracted from near-silent segments. This performance drop is due to the contamination of speech on mobile device frequency response when modeled as a linear system.

Table 5.40: Performance of the ZMBic Feature Set against Selected State-of-the-Art Feature Sets Extracted from Speech Recordings

Features	*MAE*	*RMSE*	*RAE*	*RRSE*	*Kappa Statistic*	*ACC*
HuMBic$_{M}$	0.001	0.028	4.57%	30.24%	0.954	95.47%
ZMBic$_{M}$	0	0.011	0.77%	12.42%	0.992	99.24%
ZMBic$_{Ph}$	0.003	0.055	18%	60%	0.82	82.15%
ZMBic$_{M}$+ ZMBic$_{Ph}$	0	0.01	0.63%	11.26%	0.994	**99.37%**
[entropy-MFCC]$_{Shannon}$ + [entropy-MFCC]$_{Tsallis}$	0.002	0.04	9.82%	44.33%	0.902	90.26%
mean-MFCC (13 default coefficients)	0.003	0.056	19.25%	62.05%	0.808	80.91%
mean-MFCC (n=48, M=49)	0	0.012	0.89%	13.31%	0.991	99.12%
GSV-MFCC (13 default coefficients)	0.006	0.075	33.74%	82.14%	0.663	66.55%
GSV-MFCC (n=48, M=49)	0.002	0.04	9.63%	43.88%	0.904	90.46%

5.5.4.3 Influences of the Mobile Device Environment

The experiment in this section was conducted to investigate the influences of the recording environment on the performance of the individual source mobile device identification module. Here the study repeated the experiment in Section 5.4.4.2, in order to train the classifier based on different environmental influences. At first, the experiment was conducted with call recordings corresponding to environment B and then each time the call recordings from the environments C, D, and E were added to the dataset. Table 5.41 reveals the results for individual source mobile device identification based on the call recordings from different environmental scenarios. It is evident that by increasing the inhomogeneity of the data instances with respect to the mobile device environment, the reduction in performance of the ZMBic feature set has increased.

Moreover, it is clearly can be seen that for all classifiers the overall performance reduction due to combining the data instances corresponding to calls that received from different environments and recorded with Dell stationary is less than the performance drop for entropy-MFCCs during source mobile device model identification (Refer to Table 5.29). This is because the ZMBic feature set is more robust against different

environments, whereby these environments have different response function that leaves different influences on the call recording signal.

Table 5.41: Influences of Different Environments on Performance of the $ZMBic_M$ Feature Set for Identifying Individual Apple iPhone 4 Devices

Environment B, C						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
MultilayerPerceptron	0.006	0.06	3.32%	20.14%	0.976	97.82%
Logistic	0.004	0.064	2.31%	21.24%	0.977	97.91%
IBk	0.005	0.062	2.66%	20.79%	0.978	98.05%
RotationForest	0.011	0.063	5.95%	21.1%	0.975	97.77%
SMO	0.16	0.272	89.13%	90.79%	0.974	97.68%
Environment B, C, D						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
MultilayerPerceptron	0.007	0.066	3.83%	22%	0.969	97.24%
IBk	0.005	0.063	2.55%	21.5%	0.978	98%
RotationForest	0.012	0.066	6.65%	22%	0.971	97.36%
SMO	0.161	0.273	89.17%	90.82%	0.968	97.1%
Environment B, C, D, E						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
MultilayerPerceptron	0.021	0.125	11.86%	41.53%	0.889	90%
IBk	0.005	0.068	2.82%	22.63%	0.974	97.69%
RotationForest	0.013	0.069	7.12%	22.9%	0.969	97.24%
SMO	0.161	0.273	89.23%	90.82%	0.961	96.45%
Environment C, D, E						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
MultilayerPerceptron	0.021	0.12	11.47%	40%	0.894	90.45%
IBk	0.004	0.062	2.46%	20.53%	0.979	98.1%
RotationForest	0.012	0.067	6.54%	22.23%	0.968	97.1%
SMO	0.161	0.272	89.19%	90.81%	0.97	97.26%
Environment D, E						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
MultilayerPerceptron	0.007	0.062	3.72%	20.56%	0.973	97.55%
Logistic	0.005	0.067	2.59%	22.32%	0.974	97.65%
IBk	0.004	0.056	2.32%	18.81%	0.982	98.4%
RotationForest	0.013	0.068	6.95%	22.66%	0.969	97.2%
SMO	0.16	0.272	89.14%	90.81%	0.971	97.4%

5.5.4.4 Influences of the VoIP and Cellular Communications

The experiment evaluates and compares the performance of the $ZMBic_M$ feature set for individual source mobile device identification based on recorded VoIP calls against cellular calls and their combination. Table 5.42 reveals the results, which indicate that the $ZMBic_M$ feature set is perfectly capable of detecting the specific mobile device transmitted the recorded cellular call, whereby the identification accuracy and error rates compare well with the results for detecting the individual source mobile device of the

VoIP calls. In addition, because the recording signals corresponding to both VoIP and Cellular calls contain the mobile device response function, it is also possible to combine the data instances from both VoIP and Cellular calls for individual source mobile device identification.

Table 5.42: Individual Source Mobile Device Identification for VoIP and Cellular Call Recordings by Using ZMBic Feature Set.

VoIP Calls						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
MultilayerPerceptron	0.006	0.05	3.08%	16.52%	0.983	98.46%
Logistic	0.003	0.053	1.68%	17.8%	0.983	98.46%
IBk	0.005	0.06	2.7%	18.47%	0.983	98.46%
RotationForest	0.01	0.061	5.53%	20.38%	0.978	98%
SMO	0.16	0.272	89.08%	90.78%	0.978	98%
Cellular Calls						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
MultilayerPerceptron	0.004	0.036	2.22%	11.98%	0.991	99.22%
Logistic	0.001	0.035	0.8%	11.5%	0.993	99.33%
IBk	0.004	0.047	2.44%	15.67%	0.988	98.9%
RotationForest	0.005	0.044	2.86%	14.52%	0.986	98.78%
SMO	0.16	0.272	89%	90.68%	0.99	99.11%
VoIP and Cellular Calls						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
MultilayerPerceptron	0.006	0.055	3.17%	18.47%	0.978	98.5%
Logistic	0.003	0.055	1.75%	18.39%	0.982	98.4%
IBk	0.004	0.053	2.1%	17.6%	0.984	98.6%
RotationForest	0.009	0.058	5.14%	19.37%	0.978	98%
SMO	0.16	0.272	89.1%	90.72%	0.978	98.05%

5.5.4.5 Influences of the Recording Stationary

The experiments conducted in this section at first utilized the VoIP call recording signals transmitted from 10 individual Apple iPhone 4 devices and recorded by the Dell stationary, then added the VoIP calls transmitted from the same devices and recorded by the iMac stationary, and then finally added the corresponding cellular calls recorded by the Nokia Lumia stationary to the dataset one at a time. Table 5.43 contains the results based on each stationary subset and their combination for individual source mobile device identification by using the ZMBic feature set and five different classifiers. It is evident that the detection performance results corresponding to VoIP calls recorded with Dell stationary are in exceptionally good agreement with the results corresponding to VoIP

calls recorded with iMac stationary. Section 5.4.4.4 discussed the differences between Dell and iMac stationary and their VoIP recording software as well as Nokia Lumia stationary and its GSM call recorder application. Despite these differences in stationaries and recording conditions, the performance of the individual source mobile device identification remains robust against combining the data instances from different stationaries.

Table 5.43: Influences of Different Stationaries on Performance of the ZMBic Feature Set for Individual Source Mobile Device Identification

VoIP Calls, Dell Stationary						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
MultilayerPerceptron	0.006	0.05	3.08%	16.52%	0.983	98.46%
Logistic	0.003	0.053	1.68%	17.8%	0.983	98.46%
IBk	0.005	0.06	2.7%	18.47%	0.983	98.46%
RotationForest	0.01	0.061	5.53%	20.38%	0.978	98%
SMO	0.16	0.272	89.08%	90.78%	0.978	98%
VoIP Calls, iMac Stationary						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
MultilayerPerceptron	0.005	0.047	2.9%	15.53%	0.99	98.67%
Logistic	0.002	0.04	0.94%	13.21%	0.99	99.11%
IBk	0.004	0.042	2.2%	14%	0.99	99.11%
RotationForest	0.006	0.045	3.28%	15.08%	0.988	98.89%
SMO	0.16S	0.272	88.95%	90.72%	0.99	99.11%
VoIP calls, Dell and iMac Stationary						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
MultilayerPerceptron	0.005	0.05	2.83%	16.8%	0.983	98.5%
Logistic	0.003	0.056	1.86%	18.66%	0.982	98.35%
IBk	0.004	0.054	2.16%	17.91%	0.984	98.55%
RotationForest	0.011	0.061	5.87%	20.44%	0.976	97.85%
SMO	0.16	0.272	89.04%	90.66%	0.981	98.3%
VoIP and Cellular calls, Dell, iMac and Nokia Lumia Stationary						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
MultilayerPerceptron	0.012	0.091	6.9%	30.17%	0.94	94.62%
Logistic	0.005	0.068	2.66%	22.78%	0.974	97.62%
IBk	0.004	0.054	2.02%	18.12%	0.984	98.52%
RotationForest	0.01	0.061	5.7%	20.36%	0.976	97.79%
SMO	0.16	0.272	89.05%	90.63%	0.979	98.1%

5.5.5 Robustness against Selected Post-Processing Operations

The experiment in this section utilized the subset of DS3 dataset including the VoIP call recording signals corresponding to 10 individual Apple iPhone 4 devices in environment B, recorded by using Dell stationary ('.wav' format), whereby the data instances were generated from the call recordings after the three different post-processing

operations, including the splitting, MP3 conversion, and denoising. The experiment and procedures that described in Section 5.4.5 for all three post-processing operations were repeated here, in order to evaluate the robustness of the ZMBic feature set against selected post-processing operations. Table 5.44 summarizes the results for individual source mobile device identification for call recordings undergone further post-processing operations by using the ZMBic feature set. It appears that the ZMBic feature set is moderately robust against post-processing operations such as splitting, and denoising. Nevertheless, the feature set is less robust against MP3 conversion due to the effects of compression.

Table 5.44: Influences of Different Post-Processing Operations on Performance of the ZMBic Features for Individual Source Mobile Device Identification

Original File						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
MultilayerPerceptron	0.006	0.05	3.08%	16.52%	0.983	98.46%
Logistic	0.003	0.053	1.68%	17.8%	0.983	98.46%
IBk	0.005	0.06	2.7%	18.47%	0.983S	98.46%
RotationForest	0.01	0.061	5.53%	20.38%	0.978	98%
SMO	0.16	0.272	89.08%	90.78%	0.978	98%
Split File						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
MultilayerPerceptron	0.008	0.063	4.27%	21.05%	0.973	97.6%
Logistic	0.002	0.041	0.996%	13.52%	0.991	99.2%
IBk	0.006	0.049	3.48%	16.3%	0.987	98.8%
RotationForest	0.015	0.075	8%	24.88%	0.958	96.2%
SMO	0.16	0.274	89.1%	90.78%	0.98	98.2%
Compressed File						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
MultilayerPerceptron	0.007	0.061	4.84%	22.38%	0.971	97.4%
Logistic	0.003	0.054	2.13%	19.71%	0.98	98.2%
IBk	0.016	0.111	10.87%	40.64%	0.916	92.4%
RotationForest	0.011	0.066	7.44%	23.97%	0.958	96.2%
SMO	0.139	0.26	92.45%	93.3%	0.973	97.6%
Denoising						
Classifier	***MAE***	***RMSE***	***RAE***	***RRSE***	***Kappa Statistic***	***ACC***
MultilayerPerceptron	0.005	0.048	2.83%	16.12%	0.984	98.55%
Logistic	0.002	0.041	1.02%	13.56%	0.99	99.1%
IBk	0.008	0.076	4.2%	25.32%	0.968	97.1%
RotationForest	0.013	0.066	7%	25.32%	0.975	97.73%
SMO	0.16	0.272	89.01%	90.79%	0.982	98.36%

5.5.6 Discussion

The higher-order spectral analysis produced an excellent performance for identifying the non-linear systems operating under a random input. This study modeled the acquisition section of the mobile device's impulse response with nonlinear function. The higher-order spectral analysis was used to model this nonlinearity using the ZMs of the bicoherence magnitude and phase spectrum. Malik and Miller (2012) achieved promising results based on the distance between scale-invariant Hu moments of the bicoherence magnitude spectrum and the cross-correlation between the bicoherence phase spectra for automatic microphone identification. Although this study proved that the significant distances and correlations exist between the scale-invariant Hu moments of the bicoherence magnitude and phase respectively that allow individual source microphone identification, the study focused on the evaluation of the effectiveness of the proposed system using ambient noise recording only on the small group of microphones. In this section, the experimental setup evaluated the feasibility of the proposed features against selected feature extraction methods from literature, including the statistical measures of MFCCs, and scale invariant Hu moments of the bicoherence through different types of classifiers and clustering techniques that were implemented in data mining tool WEKA. The analysis was performed based on individual source mobile device identification among 120 mobile devices in 12 models for speech and non-speech segments under different environmental influences, communication networks, and stationaries.

The experiments in this section optimized the lightness and performance of the bicoherence by defining the small *nfft* value, selecting the type and order of the Zernike polynomials and computing the ZMBic from near-silent segments of the call recording signal for the prototype of the individual source mobile device identification. Surprisingly, this feature set exhibited slightly higher performance to capture characteristics of the mobile device response function, when extracted from the original

speech recording signal. This is because by framing the speech signal into short audio frames a stochastic nonstationary speech signal was considered as stationary for the duration of the frame. Alternatively, the good performance of the ZMBic features from the original speech recording signal is because, for the collected call recording dataset in this study, the variations due to the different speakers and speech context were minimal. Moreover, because the performance of the ZMBic features from the near-silent segments of the speech recording signal compares well with when extracted from the original speech recording signal, it is a good practice to eliminate the need for collecting the speech- and speaker-independent dataset. The results indicated that by using all selected classifiers, ZMBic feature set always exhibits high performance against conventional mean-MFCC and GSV-MFCC feature sets. Apart from that the combination of the sale invariant Hu moments of the bicoherence magnitude and phase features performed well for individual source mobile device identification. These findings also proved the significance of spectral analysis techniques, in optimizing the acoustic features such as entropy-MFCC and ZMBic feature set on capturing the device specific information from the call recording signal.

Furthermore, in terms of classifiers, NN (MultilayerPerceptron), linear logistic regression (Logistic), nearest-neighbor instance based learning (IBk), rotation forest (RotationForest), and SVM classifier (SMO) achieved the best performance with respect to the classification accuracy and robustness. In general, some aspects of the proposed method compare well with existing research on microphone identification. However, this study adds an advantage to the previous approach in the following ways: (a) apply HOSA for detecting the transmitting mobile device nonlinear system from the call recording signal, (b) extracts the features from near-silent segments of the real-time call recording conversations to diminish the need for collecting a set of call recordings with a variety of speakers who are different with the speakers used for collecting the test dataset, (c)

optimizes the bicoherence spectrum by selecting the most suitable control parameters, (d) applies the orthonormal 30-degree hexagonal Zernike polynomials on bicoherence spectrum that has the following advantages over scale-invariant Hu moments: provides simple rotation invariance, higher accuracy for detailed shapes, less information redundancy and better image reconstruction, (e) allows individual communicating mobile device identification from recorded VoIP and cellular calls.

5.5.7 Conclusion

The problem of optimizing acoustic features which use HOSA techniques in order to detect the communicating mobile device response function despite the existence of undesired convolutional transfer functions was studied. This section developed different experimental frameworks for evaluation of the ZMBic feature set along with state-of-the-art feature sets from literature for individual source mobile device identification. The proposed scheme specifically identifies the individual transmitting mobile devices from recorded VoIP and cellular calls for a forensic investigation. The ZMBic feature set has discriminated the frequency response of the mobile device nonlinear system against contaminating influences such as speech and environment. It could be inferred therefore that the results in this section provide the framework for future studies to assess the performance for practical audio forensic cases.

5.6 Performance Evaluation-Phase V: Source Mobile Device Model Identification in Open Sets

In Sections 5.2-5.5, the experiments entitled evaluation of feature-based mobile device identification through closed sets, where all questioned mobile device sources were considered known. Subsequently, in this section, the study considers the problem of unknown mobile devices and possible false detection in so-called open sets. The proposed open set algorithm handles the unknown mobile devices via one-against-all LIBSVM classifier and minimizing the training data error associated with it. As detailed in Section 4.1.5.3, the proposed open set source mobile device model identification framework in this study matches a call recording to its source model by using the entropy-MFCC feature set, whereby this work has access to a limited set of mobile device models for training, and a call recording can be processed and transmitted from any mobile device model, including mobile device models to which this work never had access.

5.6.1 Experiment and Procedure Description

In the closed set evaluation, source mobile device model identification is considered as an *N*-class (*N* is the number of known models) classification problem and solved with multi-class classifiers. The trained multi-class classifiers predict a class label for a testing dataset. However, when the call recordings received from unknown sources are presented in the evaluation, they will be falsely predicted to the false classes among known models. In order to overcome this problem, the open set classification algorithm employed in this study consists of the following steps:

(a) Call recordings corresponding to a total of 7 out of 10 mobile device units of the same model located at different environments were utilized in order to create training data instances for each mobile device model.

(b) Subsequently, the algorithm built a 12-class SVM classifier using LIBSVM library with RBF kernel, and optimized C and γ parameters from the training set of instances corresponding to call recordings received from all 12 mobile device models.

(c) The call recordings corresponding to all 3 remaining mobile device units from each model were assigned to the test set. The algorithms were tested against each call recording file one at a time in order to predict the source mobile device model. Because in the open set scenario the true class label of the mobile device is unknown, the data instances generated from the corresponding file were tested against all possible class labels. The algorithm determined the maximum prediction accuracy and the class label by which the maximum prediction accuracy was achieved.

(d) If the maximum prediction accuracy is above the specific threshold, then the call recording file is received from the predicted source mobile device model, otherwise, the algorithm determined that the source mobile device model was unknown.

(e) Next, the experiment reevaluated the proposed open set scheme by repeating the steps (c) and (d), using the call recordings from DS4 dataset. This dataset consists of call recordings from both cellular and VoIP communications, which were recorded on the real-time basis without controlling the speech, environment, and stationary device parameters. More details on the specifications of this dataset were provided in Section 4.1.1.

5.6.2 Results

The experiment at first utilized DS3 dataset which consists of all VoIP calls recorded with Dell stationary, all VoIP calls recorded with iMac stationary and all GSM calls recorded with Nokia Lumia stationary. Table 5.45 shows the results of the source mobile device model identification by using the entropy-MFCC feature set with respect to each call recording subset. Meanwhile, the call recordings from environment B, C, D and E were recorded with the same stationary as the query dataset were perfectly detected with

identification accuracy of above 90%. The performance is consistent with results obtained with 10-fold cross validation. Nevertheless, the identification accuracy has slightly reduced due to the use of different devices for training and test set. Second, the experiment utilized DS4 without controlling the speech, environment, and stationary device parameters.

Table 5.45: Identification Accuracies for One-Versus-All SVM Classifier for Identifying the Source Model of the Mobile Devices in DS3

Model	*VoIP Calls /Dell stationary ACC%*				*VoIP Calls / iMac stationary ACC%*				*Cellular Calls / Nokia Lumia stationary ACC%*			
($\mathcal{M}$)	d_8	d_9	d_{10}	$d_{8...10}$	d_8	d_9	d_{10}	$d_{8...10}$	d_8	d_9	d_{10}	$d_{8...10}$
i4	91	92.5	90.24	90.31	90.33	91.2	90.48	90.54	93.22	90.68	90.76	92.32
i4S	90.57	90.24	96.59	90	89.57	90.84	94.39	92.11	96.8	94.18	95.56	93.22
i5	91	92.2	94	90.34	96	93.6	93	95.18	93.04	94.8	95.83	92.82
i5S	91.18	92.88	91.88	91.58	94.28	92.7	96.08	93.02	96.07	98.25	94.24	92.82
S3M	96.56	91.36	94.48	92	90.1	91.36	93.48	91.34	91.87	90.37	92.05	90.11
S3	92.73	93.72	93.16	92.16	99	97.88	90.23	91.13	98.77	97.62	96.2	94.97
S4	94.32	90.71	98.69	90.25	97.43	92.92	93.38	91.13	98.51	93.67	98.78	97.06
N3	92.25	90	92.33	92.71	96.52	91.58	97.65	90.68	96.2	94.97	95.63	94.22
N4	95.94	94.78	96.36	95.11	96.3	96.98	94.5	92.48	93.37	90.94	90.13	91.78
C	98.37	95.48	90.77	93.27	98.37	95.48	92.25	95.63	97.26	90.9	94.88	95.06
Z	90.31	95.88	98.36	94.39	99.23	95.4	90.45	96.75	97.13	93.01	94.29	91.49
Z1	90.58	93.33	91.56	90.21	90.68	90.2	93.15	92.18	92.33	90.1	93.3	90.44

Table 5.46 demonstrates the communication type of the mobile devices and the accuracies for identifying the model of their communicating mobile devices by using the proposed open set evaluation. Although the mobile device response function for such a query call recording was contaminated with different influences, the results are quantitatively similar to those of earlier evaluations. Moreover, the experiment evaluated the proposed scheme for a set of call recordings from DS2 (Refer to Appendix B3), that their model is not known to the trained classifier by using DS3. For the call recordings with the unknown model, the proposed open set scheme achieved the maximum accuracy of less than 8.33% in which the algorithm returns the source model of the query call recording as unknown and will assign it to the outlier class.

Table 5.46: Identification Accuracies for One-versus-All SVM Classifier for Identifying the Source Model of the Mobile Devices in DS4

Mobile Device Model $\mathcal{M}$	***Communication Type***						
	d_1	d_2	d_3	d_4	d_5	d_6	d_7
i4	Cellular	Cellular	Cellular				
i4S	Cellular	Cellular	Cellular				
i5	Cellular	Cellular	Cellular	VoIP	VoIP	VoIP	Cellular
i5S	Cellular	Cellular	Cellular	Cellular	VoIP	Cellular	VoIP
Mobile Device Model $\mathcal{M}$	***%ACC***						
	d_1	d_2	d_3	d_4	d_5	d_6	d_7
i4	90.33	91.65	89.6				
i4S	89.72	90.6	90.36				
i5	90.99	90.51	89.08	89.25	90.6	89.98	90.88
i5S	90.14	88.13	91.3	89.45	91.83	91.85	90.22

5.6.3 Discussion

The results presented for the open set source mobile device model identification in the previous section are encouraging as the query call recordings with known model were correctly assigned to their class labels, and the call recordings with the unknown model were assigned to the outlier class. It appears from this evidence that although the influences of speech, environment, and the stationary device could be controlled or eliminated, the type of communication, as well as the quality of call and network affects the performance of the entropy-MFCC feature set for source mobile device model identification.

Specifically, the performance is influenced by two main factors, firstly, the type and quality of network connection during VoIP calls, (i.e. the variation of the Wi-Fi speed and coverage, high to low cellular speeds (2G-4G)), which reduces the similarity distances between the feature values corresponding to call recordings from the same mobile device and secondly, random effects of the wireless communication channel due to different locations and variable radio parameters. A high influence of both factors causes overfitting in which the algorithm works well with the data instances that utilized during the training, but not on the data instances that generated from the new call recordings. As an implication from this, the framework should consider the limitation as

one of the factors in order to facilitate automatic source mobile device model identification in real-world forensic investigations.

The first factor is difficult to control; the framework only controlled the network architecture influences by using the same Wi-Fi connection provided on the university campus for all VoIP communications. Moreover, still, there is no control over the Wi-Fi strength and coverage. Further, the Wi-Fi will be automatically shifted to cellular data when Wi-Fi connectivity is poor. The quality of this connection also depends on to the cellular data speed.

The second factor, however, has been considered in the design of the framework as following:

- The framework assumed that the GSM cellular communication always utilizes the same speech coder for communication.
- The framework controlled the influences via different obstacles, reflectors, and diffractors in its propagation path using the same four locations for positioning the mobile devices with respect to the stationary device.

Furthermore, in order to compare its performance with other studies, it is compared with representative instance OCC approach in Vu et al. (2012) for identifying microphones from noise recordings. This study took advantage of the OCC approach to focus on the detection of target class instances by using tight decision boundary determined from training data. Hence, the microphone identification experiments achieved the overall recall of above 0.8 and low precision rates across the OCC models. It should be noted that the OCC models were required to be trained only one time and independently. As a result, when a new microphone is considered only the new OCC model was trained on audio samples of that microphone without retraining the current

OCC models. In this study, despite the most challenging scenarios, the proposed scheme determined high recall and precision rates of above 0.99, which is a promising result in comparison to Vu et al.'s study. This result strongly suggests that the proposed framework has shown an appropriate robustness in terms of the speech signal, noisy environments and communication channel; moreover, because of the performance results, it is appropriate for applying it to the automatic source mobile device identification. Unfortunately, this is hardly a full evaluation of the performance and further computational effort is required, yet it presents a basis for comparison with state-of-the-art studies.

5.6.4 Conclusion and Limitations

The fifth phase of the evaluation study investigated the performance of the proposed framework for open set source mobile device model identification. The performance results investigated the two most challenging influences contaminating the mobile device response function known as the type and quality of the network or cellular connection during the call, as well as the random effects of the wireless communication channel. Both influences generate high variance in the dataset and cause an overfitting problem. The overfitting problem reduces the performance of the source mobile device model identification process, whenever the mobile devices of the known model to the classifier were not used during the training of the classier. In conclusion, the investigation was done by considering different call recording scenarios as follows:

(a) The performance of the source mobile device model identification process for VoIP and Cellular calls was measured using the identification accuracy in the detection process. It appears that the performance results are significant with a reasonable false detection corresponding to DS3 dataset. Moreover, the detection performance is influenced by the same factors, mainly because the same control conditions were

applied during collecting the calls transmitted from training and test subset mobile devices. Further, the performance of the source mobile device model identification process which was trained with VoIP and Cellular call recordings from DS3 and tested with call recordings from DS4 and DS2 dataset revealed high identification accuracy that is slightly lower than the first evaluation. The performance reduction was due to the different control conditions were applied during collecting the DS4 dataset.

(b) In comparison with the performance results of Vu et al. (2012), the performance of the proposed framework has shown a reasonable result in the open set source mobile device identification process. In terms of the robustness evaluation, the proposed method maintained good performance after evaluating the framework for the call recordings collected under uncontrolled conditions.

In addition, the biggest advantage of the investigation in this study is to analyze the overall performance, with different control conditions in the data collection process, in order to evaluate the feasibility of the proposed open set framework. In addition, with the main aspects above, the experiment also clearly demonstrate that the performance of the open set source mobile device identification process is reasonable and can be seen as a significant result since it can be achieved with high identification accuracy and robustness.

In conducting the experiment, this study has found some limitations and below is some of these along with suggestions on how to reduce them.

(a) *Connection Speed and Quality*: Although the results presented in this study were considered a reasonable result, they were also influenced by the different Wi-Fi strength and coverage as well as the random effects of the wireless communication channel. Further, the Wi-Fi will be automatically shifted to cellular data when Wi-Fi connectivity is poor. The quality of this connection also varies with the cellular data

speed. This limitation increases the variance of the feature values with respect to different call recordings corresponding to the same mobile device unit. This causes an overfitting problem. Therefore, in order to reduce the limitation and produce the open set evaluation in practical forensic investigations despite these influences, it is suggested to consider methods to overcome an overfitting problem such as regularization, which forces the magnitude of the parameters to be smaller.

(b) *Computation Time*: The performance evaluation of the open set source mobile device model identification framework required a considerable amount of time in terms of the training and testing process. The proposed approach at first built the multi-class SVM classifier with respect to all call recordings from four different environments, which were recorded with the one stationary at a time. The DS3 dataset consists of a total of 28 call recordings for each mobile device model, whereby for each call recording signal a total of 100 data instances were generated. Hence, the classifier was built based on the feature values corresponding to a total of 33600 data instances. The first issue is that the data preparation, feature extraction and training processes for such a large dataset require large computation time. Moreover, the second issue is that the proposed open set scheme computed the prediction accuracy of the test data instances generated from the query call recording against the pre-trained model for all possible class labels. This action also requires substantial processing time, which encourages to improve the efficiency of the proposed open set framework in the future.

5.7 Summary of the Results for Source Mobile Device Identification

This chapter has discussed the evaluation study of the spectral analysis techniques in optimizing the acoustic features for intra- and inter-source mobile device model identification modules used in the proposed framework. The promising results from the

initial phase to the final phase have presented a combination of different aspects of evaluation, and they highlighted their specific findings and conclusions.

The key objective of describing the evaluation at different levels of studies is to investigate the specific objective at each level. The findings have demonstrated strong evidence to support the capability of the proposed framework to function efficiently with respect to its operational characteristics. In addition, the comparison of the methods in the evaluation studies also drives the framework and its feasibility to enable the spectral analysis techniques in the feature extraction process in the intra- and inter-mobile device model identification.

CHAPTER 6: PROTOTYPE DESIGN AND IMPLEMENTATION

Overall, in this study, the proposed framework has been validated and evaluated through rigorous experiments. The next step of this study is to design and implement a prototype system that is capable of handling its key operations and able to demonstrate how these can be implemented in practice. This chapter discusses the prototype implementation of the proposed framework, and particularly the feature-based CDIM. The main functions of the CDIM modules have been built by using a MATLAB GUI interface that can be used to visualize, process and analyze the call recording signal transmitted from any communication device. Different types of modeling languages, like use case diagrams and state diagrams, are implemented to demonstrate the visual understating of the prototype. Lastly, the call recording scenarios that are used in this chapter are based on DS3 and DS4 dataset.

6.1 Implementation Overview

The proposed framework consists of four main parts, as shown in Figure 6.1. The three main parts, the data preparation, data visualization as well as the feature extraction modules, have been completely implemented, whereas the test and analysis modules have been adopted LIBSVM library. There are two reasons behind implementing existing machine learning tools such as LIBSVM, instead of implementing new classifiers from scratch. The first reason is that implementing these modules has been out of the scope of the proposed research, and the second reason was that utilized classifier LIBSVM is the open-source machine learning tool that widely used among state-of-the-art works with high classification accuracy, ideal computational efficiency, and robustness. The design of the modules has been developed using the MATLAB GUI, which allows the modules to have interactive and friendly interfaces. The same database was used with feature

extraction, analysis, and test systems. The descriptions of the implemented modules are as follows:

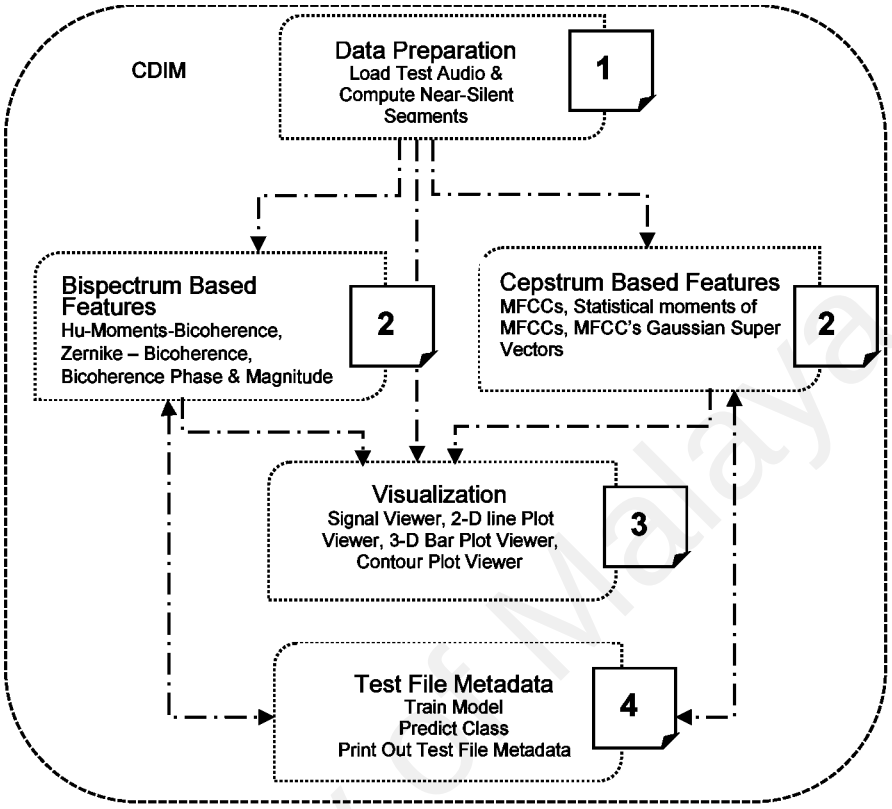


Figure 6.1: Modules Implementation

(a) *Data Preparation*: This module consists of two main sub-modules: *Sample and Enframe Signal*, and *Near-Silent Detection* modules. The *Sample and Enframe Signal* module allows to import the query call recording file, samples and enframes the signal and then calls the *Data Visualization* module to visualize the audio spectrum. The *Near-Silent Detection* module computes the near silent segments of the signal and then again activates the *Data Visualization* module to visualize the near-silent audio spectrum.

(b) *Data Visualization*: It utilizes four main plot viewer modules: *Signal Viewer*, *2-D Line Plot Viewer*, *3-D Line Plot Viewer* and *Contour Plot Viewe*r modules. The *Signal Viewer* module visualizes the call recording signal spectrum and its near silent segments. The *2-D Line Plot Viewer* module shows the feature values against their indexes, whereas the *3-D Bar Plot Viewer* module shows the absolute value of the

covariance elements of the feature sets. Further, the *Contour Plot Viewer* module demonstrates the bicoherence magnitude and phase spectrum of the signal.

(c) *Feature Extraction Systems*: The two main modules of the prototype are: *Cepstrum-based Feature Extraction* and *Bicoherence-based Feature Extraction*. More details on the functionality of these modules can be found in Sections 4.1.3 and 4.1.4, correspondingly. The modules run independently as two different approaches, and they are used to optimize acoustic features for source mobile device identification. In the same way with the experiment in the previous chapter, their implementation is based on MATLAB. During *Data Visualization*, these modules connect to the query call recording dataset uploaded by the *Data Preparation*, whereas the *Train Model* and *Predict Class* allows importing the train and test dataset path directory.

(d) *Test File Metadata*: The other two modules that play the critical role in running the prototype are*: Train Model* and *Predict Class.* Although the *Predict Class* module always requires the model that is built by the *Train Model* module in order to make the decision, both modules are able to act independently. The *Predict Class* uses the test files path directory given by the user, calls the *feature extraction* module and makes the decision by using the pre-trained model in the database. It is also possible to rebuild the model if there is any update in the database. In the end, *Predict Class* predicts the model of the mobile device utilized for making the query call and prints out the test file metadata.

6.2 Prototype Functionalities

The gain insight into the main functionalities of the proposed framework, modeling languages, such as *use case diagrams* and *state diagrams*, are presented.

6.2.1 Use Case Diagram

Use case modeling has been commonly adopted to plot a graphical functional explanation of the interaction between external entities and systems, in addition to their cooperation. The diagrams are utilized to determine the characteristics of the developed systems, without the necessity to mention how those characteristics are implemented. Figure 6.2 demonstrates the system level and explains the relationship between external systems and the system itself.

The role of user, as presented in Figure 6.2, is given below:

(a) Users are able to manage the MATLAB GUI modules that represent the Data Preparation module. This includes the ability to upload the call recording audio file, play and visualize the call recording signal, in addition to play and visualize the near-silent signal.

(b) Users are able to manage the MATLAB GUI modules that represent the Feature Extraction module. This includes the ability to select the type of the feature set, the control parameters for the feature set and visualize the statistical properties of the feature sets.

(c) Users are able to manage the MATLAB GUI modules that represent the Test File Metadata module. This includes the ability to import the path directory of the test file, update the call recording database and rebuilt the model.

The use case diagram has obtained a short-term summary of the modules' functionality. However, it lacks clarification on how those modules are operated. Hence, the state diagrams are utilized in the subsequent section.

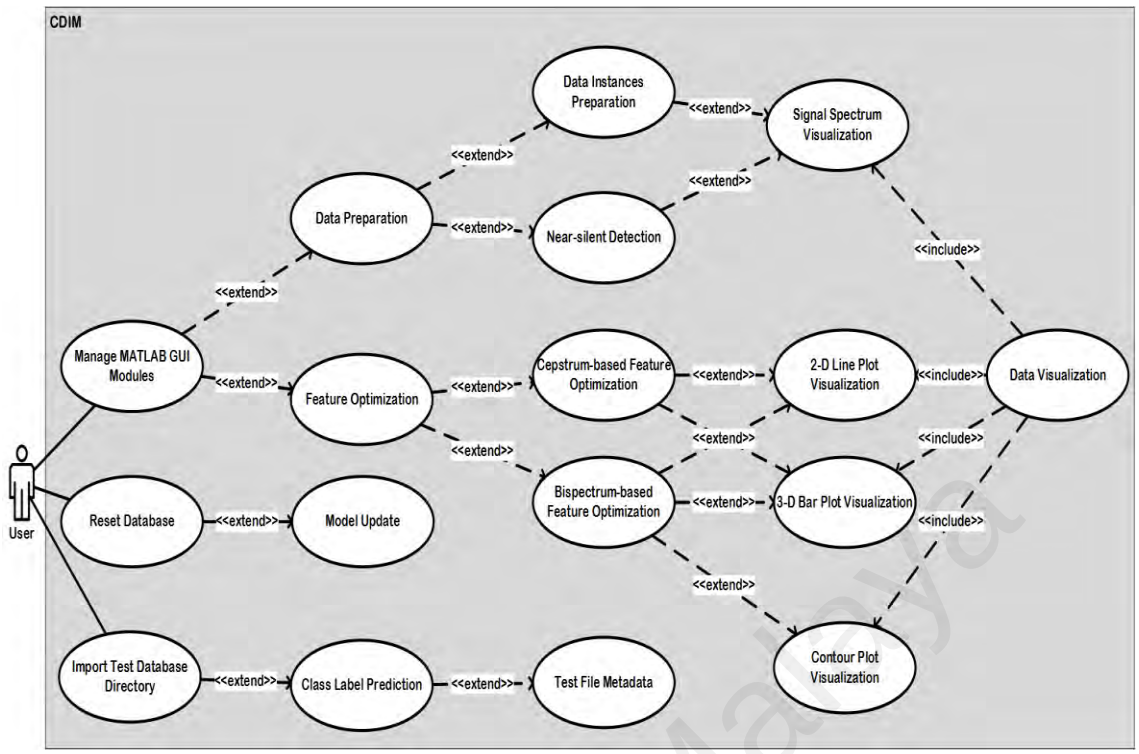


Figure 6.2: CDIM Use Case Diagram

6.2.2 State Diagrams

A state diagram designates all the possible states of an object as an event occurs, and is used to show the characteristics of an object by using many use cases of a system, in addition, to focusing on the flow of control one state to another. Figure 6.3 reveals all the possible states in the proposed framework, and it summarizes the characteristics of the running system. As prime-state, there are four main states and short-term summary of them are discussed as below:

(a) *Data Preparation*: This will be counted as the initial state (T1) if the user uploads the test file through the *Data Preparation* module. There are three sub-states in this specific state, as shown in Figure 6.4: *Plot Signal Spectrum*, *Detect Near-silent Segments*, and *Plot Near-silent Signal Spectrum*. By the end of this state, the uploaded file is sampled and enframed, and then the shortened audio frames are plotted and played back (T2.2). The user manually invokes the *Detect Near-Silent Segments* state,

by the end of this state, it plays and plots the near-silent segments of the shortened near-silent audio frames (T2.3).

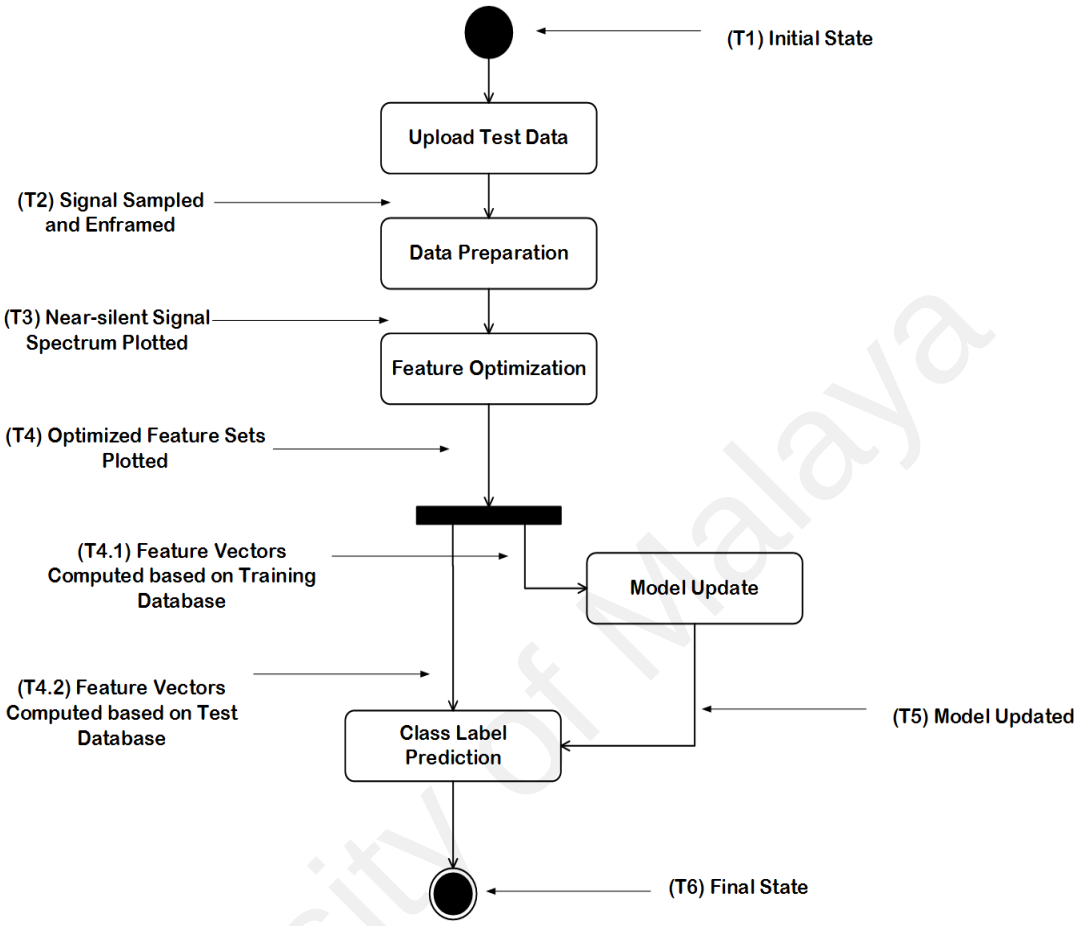


Figure 6.3: Prime-State Diagram

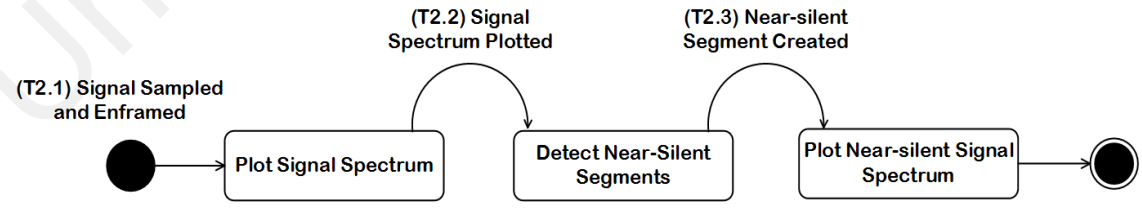


Figure 6.4: Data Preparation State

(b) *Feature Optimization*: This state is also manageable by the user and runs specific feature extraction algorithms in order to optimize acoustic features for source mobile device model identification. Hence, it allows assigning different parameters and select

plot types for a variety of cepstrum-based and bispectrum-based feature sets. Figure 6.5 shows that this state consists of two independent sub-states, correspondingly. By the end of each independent sub-state, the selected feature set is visualized in one selected plot (T3.1.3 and T3.2.3).

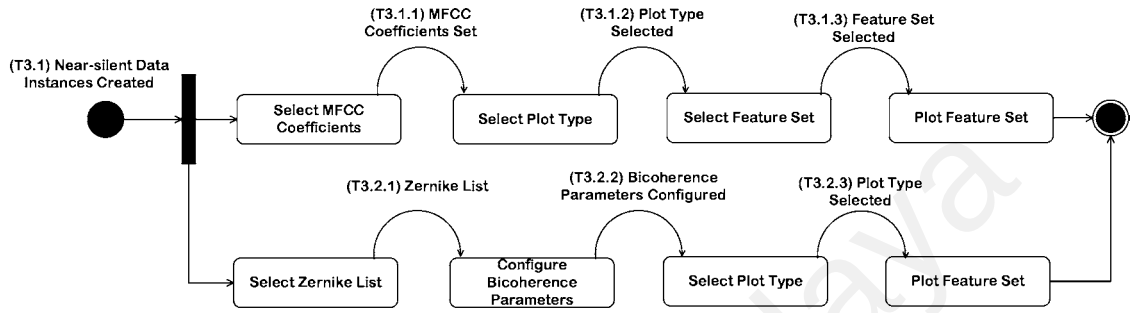


Figure 6.5: Feature Optimization State

(c) *Model Update*: This state initiates by the user if new data are added to the training database or if there is any change in parameter configuration of the LIBSVM classifier. As shown in Figure 6.6, this state is built up from three main sub-states, which run in succession: *Create Label Vector*, *Build Model*, and *Restore Model.* The output of this state will be used in the next state *Class Label Prediction.* The *Create Label Vector* state creates the label vector corresponding to all data instances in the training database and their value assigned with respect to the mobile device models in the database. The *Build Model* state builds the training model based on the LIBSVM classifier by using the feature vectors and label vectors computed from the training database. Subsequently, *Restore Model* state replaces the new model with the old model.

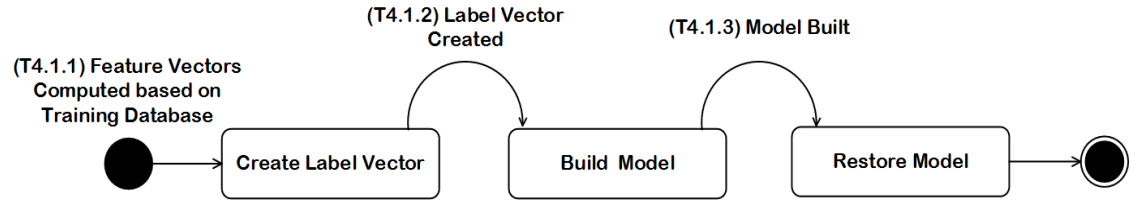


Figure 6.6: Model Update State

(d) *Class Label Prediction*: This particular state begins if the user imports the test database class path directory through the *Test File Metadata* module. Here, it stays in the same state as long as the user manually invokes the *Predict Class* state (T2). This state consists of four sub-states as illustrated in Figure 6.7: *Create One-class Label Vector, Compute Prediction Accuracy, Set Maximum Prediction Accuracy, Update Test File Metadata.* The progression from one state to another is dependent upon the output from the state before them. The class label vector is created for all possible class labels in the training database (T4.2.3); moreover, the prediction accuracy is computed for all possible class labels (T4.2.4); meanwhile the prediction accuracy updates the maximum prediction accuracy if it is larger (T4.2.5). In the end of the state maximum prediction accuracy and its corresponding class label is determined (T4.2.6) and utilized to update test file metadata.

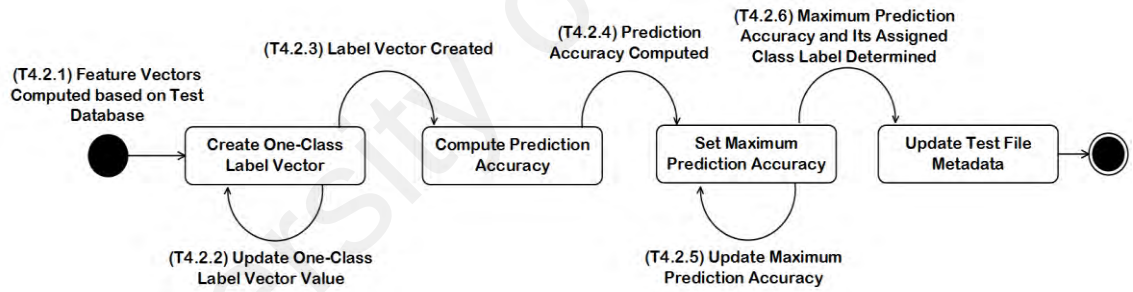


Figure 6.7: Class Label Prediction State

6.2.3 MATLAB GUI Modules

The MATLAB GUI modules facilitate a graphical user-friendly interface that allows potential users to view and configure the modules. The MATLAB GUI modules offer a MATLAB based signal processing and machine learning solutions that obtain strong understandings about the simplicity of the feature extraction and analysis process. Simplicity, easy-to-use, modifiable and adaptable modules that help optimizing feature extraction results allow potential users to investigate spectral analysis on call recording signal. Moreover, the modules offer different types of visualization and provide a wider

perspective of the existing condition regarding statistical properties of the feature sets and the test file metadata. Their functionality can be summarized into three main functions:

(a) *Data Preparation*: This category makes available the spectrum analysis of the shortened audio frames of the original call recording and its near-silent segments. As highlighted in the use case diagram in Section 6.2.1, examples of outputs that can be presented include:

   a. *Data Preparation Status*: The status bar allows the user to understand that the required module is running as long as the status is - *Processing*.

   b. *Spectrum Viewer Graph*: This helps to visualize the data preparation processes and compare the original call recording signal against its near-silent segments. It also allows optimizing the near-silent detection algorithm through the configuration of the near-silent threshold.

   c. *Audio Player*: This comes along with the *Spectrum Viewer graph,* again it allows to observe the data preparation process and optimize the near-silent detection algorithm parameters.

(b) *Cepstrum-based Feature Optimization*: This category allows visualization of the feature values output from the cepstrum-based feature optimization processes. The cepstrum-based feature optimization processes are conducted as follows:

   a. *Select MFCC Coefficient*: This option allows to select the Mel-cepstrum coefficients based on 12 zeroth order MFCCs, 13 zeroth order MFCCs including its zeroth coefficient, 24 zeroth order plus Delta coefficients, 36 zeroth order, plus Delta and Delta Delta coefficients, 39 zeroth order, plus Delta and Delta Delta coefficients, including the log energy, and 49 zeroth order MFCCs designed through optimizing the Mel filterbank spacing.

b. *Select Plot Type*: This option allows to analyze the statistical properties of the selected feature set through the 2-D line plot as well as the absolute value of the covariance elements of each feature set through the 3D-Bar plot.

c. *Select Feature Set*: This option allows to select the most suitable cepstrum-based feature set among the 12 different feature sets. The feature sets were implemented by applying linear, bark- and Mel-filterbanks, in addition to different normalization techniques applied on the LFCC, BFCC and MFCC sequence vectors in order to perform dimensionality reduction and extract the mobile device specific information.

(c) *Bispectrum-based Feature Optimization*: This category investigates the feature values output from the bispectrum-based feature optimization processes. The bispectrum-based feature optimization processes are conducted as follows:

a. *Select Zernike List*: This option allows to set the ZMs polynomials with different orders in order to perform dimensionality reduction and extract the mobile device specific information from the bicoherence spectrum.

b. *Select Bicoherence Parameters*: This option allows to compute and visualize the contour plot of the bicoherence magnitude and phase spectrum for different values of the *nfft* and number of segments per sample $N_{segsamp}$.

c. *Select Plot Type*: This option allows to study and compare statistical properties of the ZMs of the bicoherence magnitude and phase against Hu moments of the bicoherence magnitude through 2-D line plot; in addition, it compares the absolute value of the covariance elements of each feature set through the 3D-Bar plot.

(d) *Test File Metadata Identification*: This category predicts the mobile device model by using the feature vectors corresponding to the test file from the feature extraction process as well as the built model from on the training process. The model identification processes are performed as follows:

a. *Train Model*: This option allows to rebuild the SVM classifier after updating the training database or modifying the *svmtrain* options.
b. *Predict Model*: This interface facilitates the option to set the test database directory otherwise it utilizes the test file loaded in the *data preparation* interface. This interface uses the feature values corresponding to the test file and the built model to predict the mobile device model. The test file metadata appears in the text box to announce the model of the mobile device utilized for making the call.

This section has provided the prototype functionalities. Moreover, to provide the operation and visualization of the MATLAB GUI modules, print screens for the prototype can be seen in the subsequent section.

6.3 Demonstrating CDIM Prototype

The main functionalities of the proposed framework and its MATLAB GUI modules are presented in Section 6.2. This section presents some examples of how the audio signal can be processed to extract acoustic features, and how these features can be analyzed and interpreted according to their statistical properties. It also presents the details of the training and prediction results using the relevant functions to show the test file metadata, including the source mobile device model of the recorded call and the identification accuracy. The VoIP and cellular call recordings from 7 out of 10 individual mobile devices per model in DS3 are stored in a training database. The DS3 database consists of different class directories, which could be selected prior to building the model. The test database directory refers to the call recordings from unknown mobile device sources that their model may or may not be among the mobile device models in the training database.

The prototype consists of two distinct modules: the back end applications and MATLAB GUI modules. To demo the prototype, there are four sections and the details of their descriptions are as follows:

6.3.1 The Back-End Applications

In order for the back end applications of the prototype to operate appropriately, MATLAB interface of the LIBSVM package needs to be installed first by using the supported MATLAB compiler. Hence, the MATLAB assigned the Microsoft Visual C++ 2013 Professional (C) as the compiler. The remaining of the MATLAB functions is written in MATLAB language and saved in the class path directory of the MATLAB GUI.

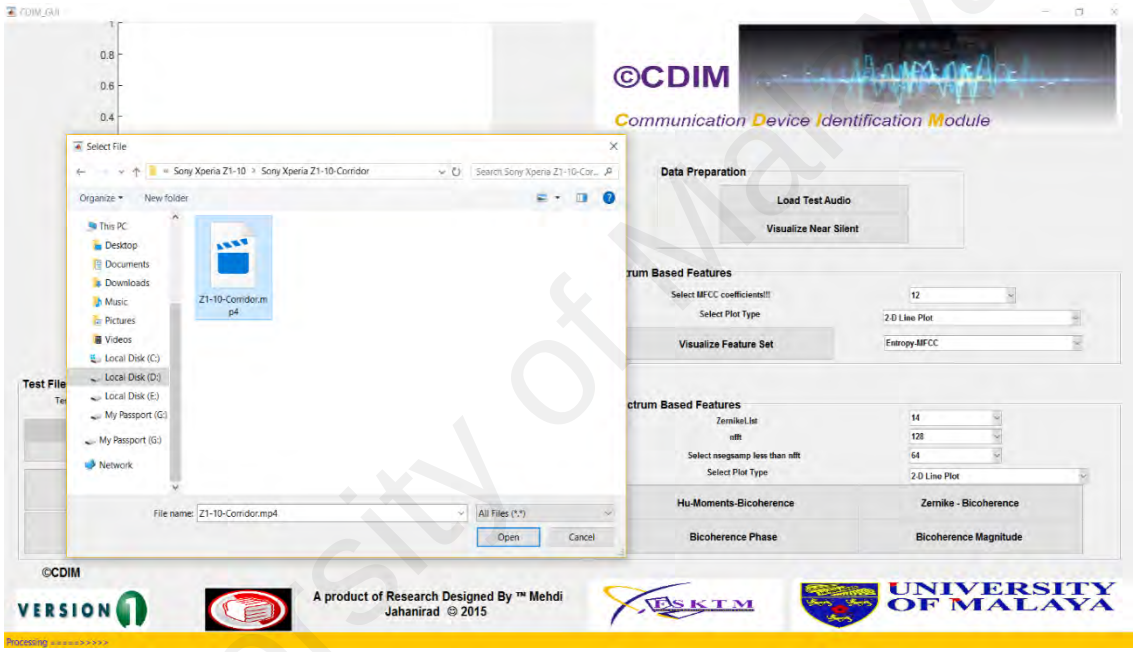


Figure 6.8: Loading the Test Audio File

6.3.2 Data Preparation

To start the visualization of the sample audio recording, the user shall go to *Data Preparation* part in CDIM module. This part contains two sections, *Load Test Audio* and *Visualize Near Silent*. Figure 6.8 illustrates the steps of loading the test audio file, in which by pressing this button the 15 seconds of sample audio recording automatically plays. The next button in *Data Preparation* module is *Visualize Near Silent* button. When the user presses this button, the near silent segments in original sample audio recording will be extracted, and the program automatically draws the spliced silent parts in the upper left side plot. By pressing this button same as the previous button, the program

automatically starts to play the 15 seconds of the silent parts from sample audio recording. Figure 6.9 illustrates the plot of the spliced near silent segments from original sample audio recording.

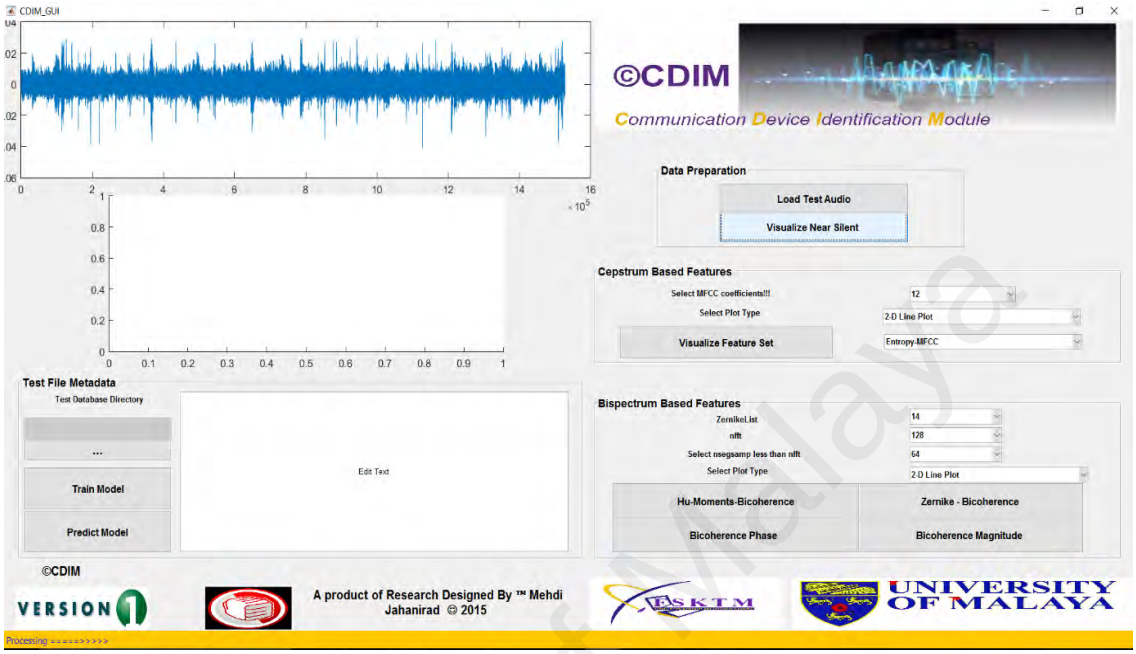


Figure 6.9: Visualizing the Near Silent Segments Spectrum

6.3.3 Cepstrum-based Feature Optimization

In *Cepstrum Based Features* part in CDIM module, there are three important sections, which are ***Select MFCC coefficients***, ***Select Plot Type*** and ***Visualize Feature Set***. In the first drop-down menu, the user can specify the number of MFCC coefficients (12, 13, 24, 36, 39). Figure 6.10 illustrates the drop-down menu which allows selecting a number of MFCC coefficients. Furthermore, in *Select Plot Type* section in *Cepstrum Based Features* part, the user can determine the type of plot, which is the *2-D Line Plot* and *3-D Bar Plot*. Figure 6.11 illustrates the drop-down menu which contains the *2-D Line Plot* and *3-D Bar Plot*. Finally, the *Visualize Feature Set* section in *Cepstrum Based Features* module, shows the benchmarking of the different feature sets that all have already been implemented and compared with the main feature set (Entropy-MFCC) among cepstrum based features.

Figure 6.10: The Drop-down Menu of the MFCC Coefficients

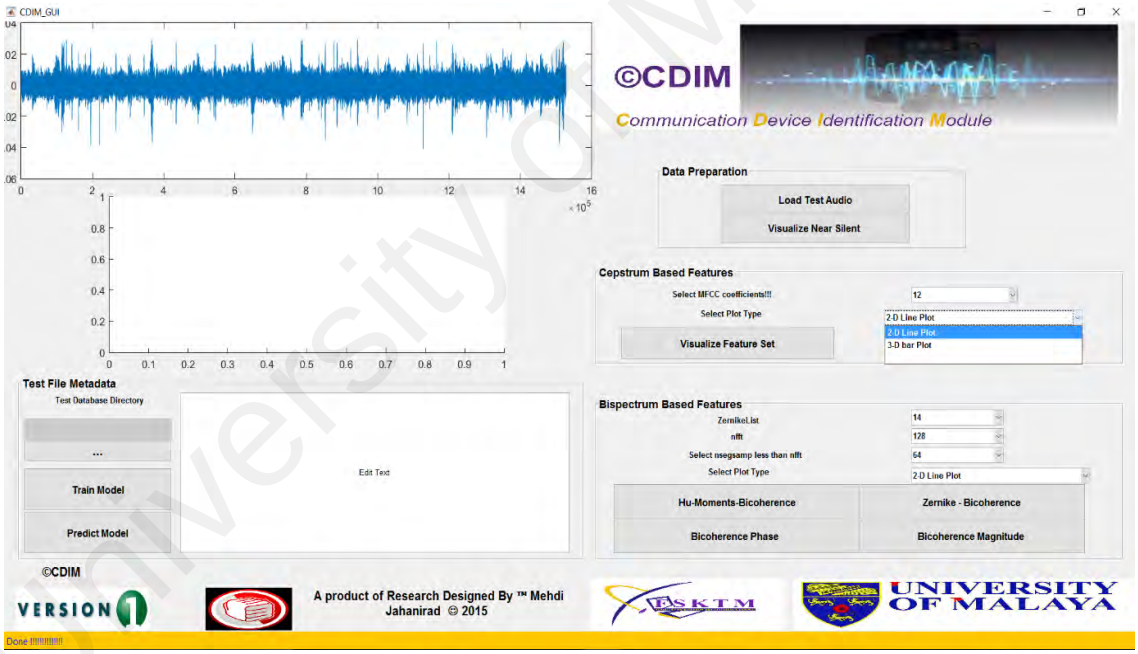


Figure 6.11: The Drop-down Menu of the 2-D Line Plot and 3-D Bar Plot

Figure 6.12: The Drop-down Menu of the Different Cepstrum Based Features

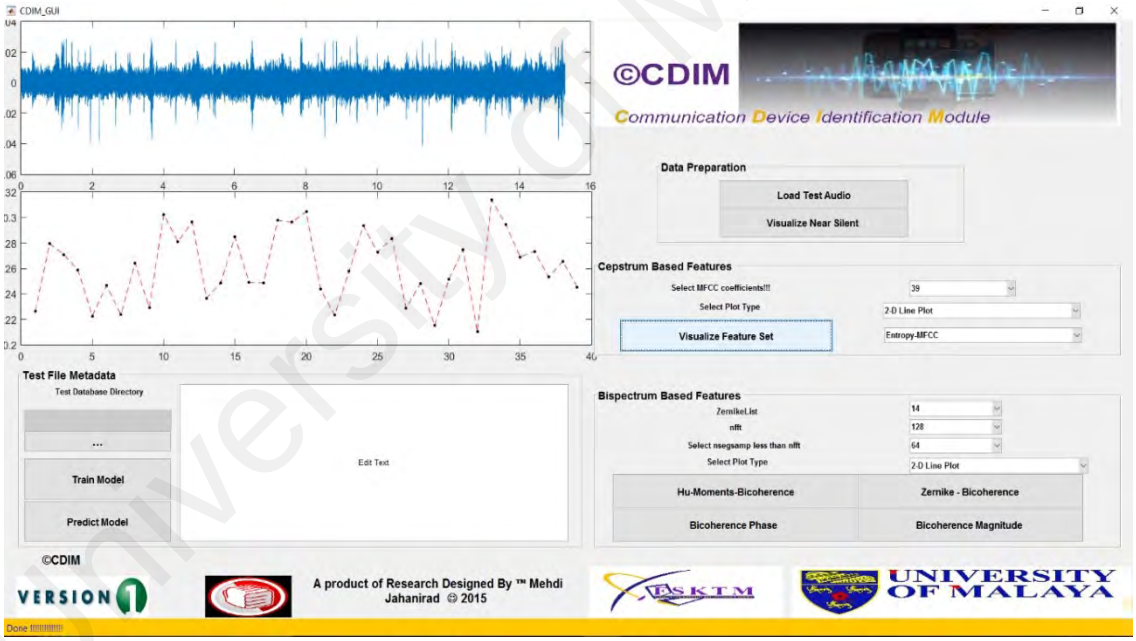


Figure 6.13: Visualization of the Selected Feature Set Using the 2-D Line Plot

Figure 6.12 illustrates the drop-down menu of different cepstrum based features (*Entropy-MFCC*, *Entropy-LFCC*, *Entropy-BFCC*, *GSV-MFCC*, *GSV-LFCC*, *GSV-BFCC*, *Mean-MFCC*, *Stdev-MFCC*, *Kurt-MFCC*, *Skew-MFCC*, *VAR-MFCC*, *MFCC*). After selecting which feature sets shall be drawn, the user has to press the *Visualize*

Feature Set button to draw the result in the plot (the middle plot on the left side of CDIM). Figure 6.13 and Figure 6.14 illustrate the 2-D Line and 3-D Bar plot, respectively.

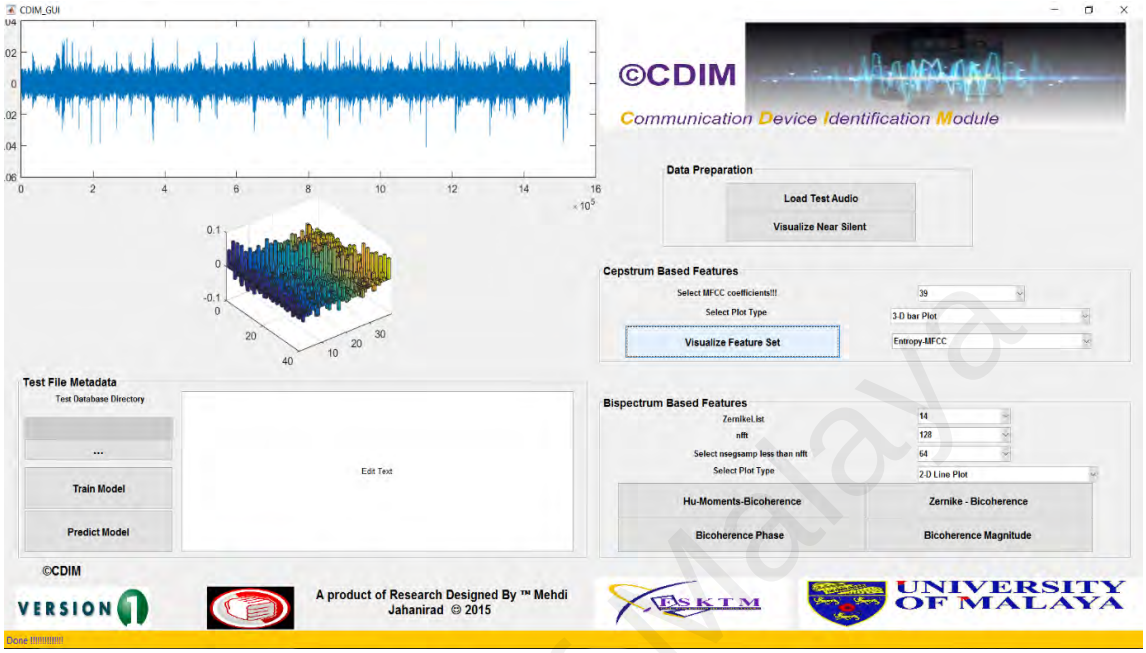


Figure 6.14: Visualization of the Selected Feature Set Using the 3-D Bar Plot

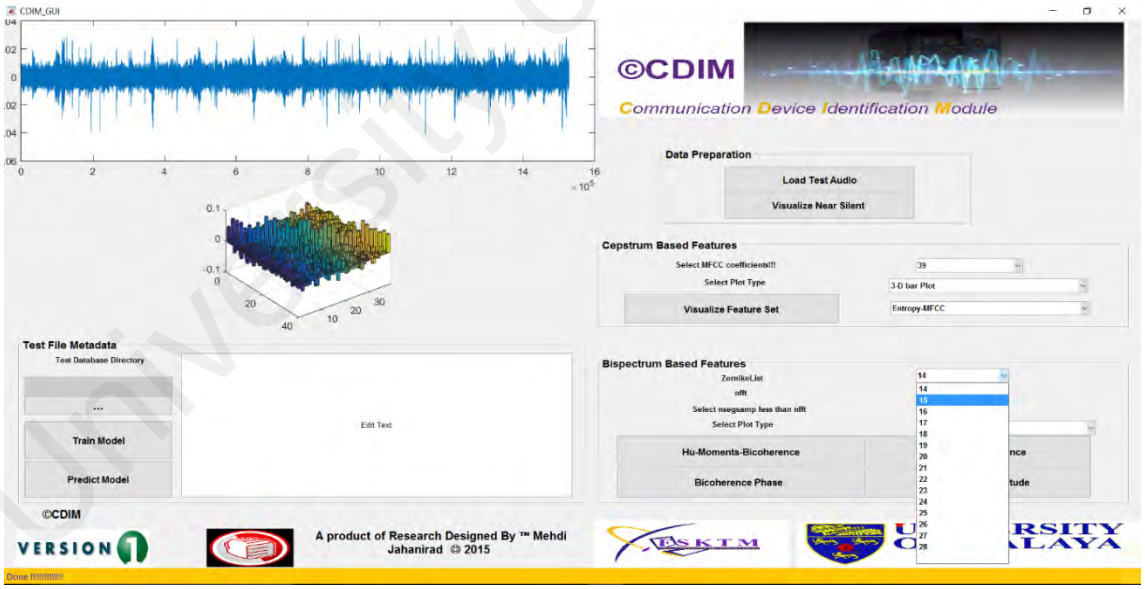


Figure 6.15: The Drop-down Menu of the Zernike Polynomial Type and Order

6.3.4 Bispectrum based Feature Optimization

In *Bispectrum Based Features* module in CDIM, there are four important sections, which are ***ZernikeList***, ***nfft, Select nsegsamp less than nfft*** and ***Select Plot Type***. In ZernikeList which is a drop-down menu, the user can specify the Zernike polynomial type

and order. Figure 6.15 illustrates the drop-down menu that can determine the Zernike polynomial type and order. Moreover, in *nfft* which is the drop-down menu, the user can specify any value for the *nfft* length to estimate bicoherence spectrum. Figure 6.16 illustrates the drop-down menu that can determine the value of *nfft* length to estimate bicoherence spectrum. In addition, *Select nsegsamp less than nfft* is another drop-down menu in this part that allows a user to select the value of *nsegsamp* to estimate bicoherence spectrum.

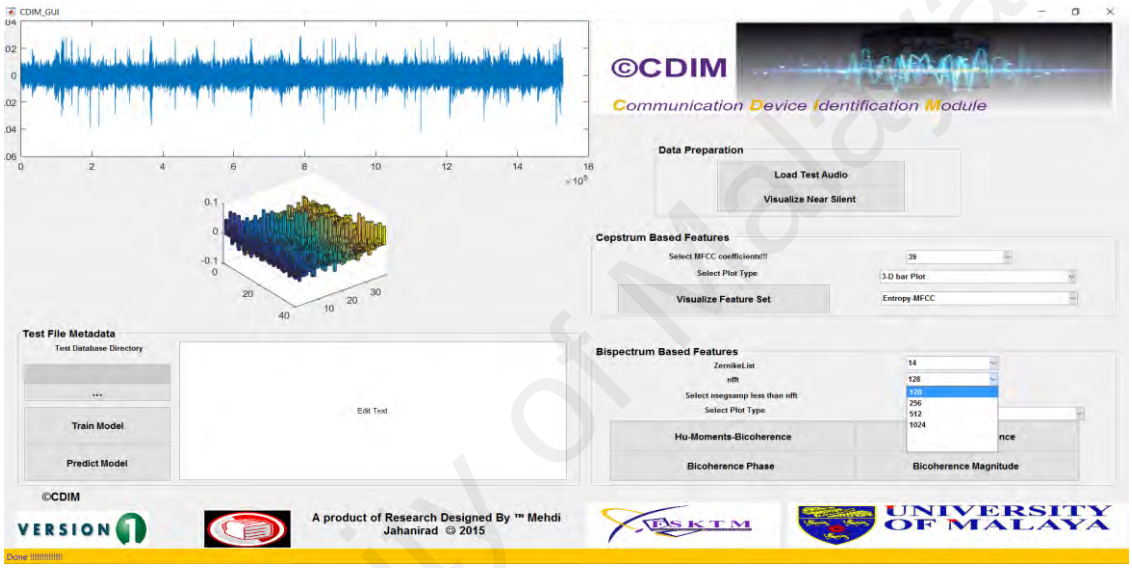


Figure 6.16: The Drop-down Menu to Set *nfft* for computing Bicoherence

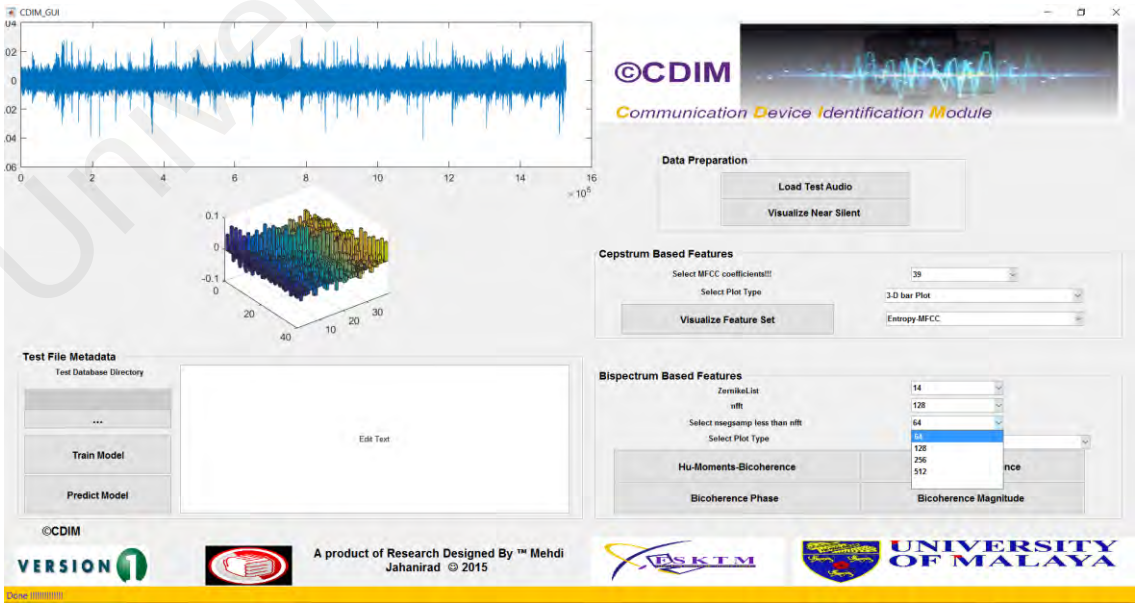


Figure 6.17: The Drop-down Menu to set *nsegsamp* for Computing Bicoherence

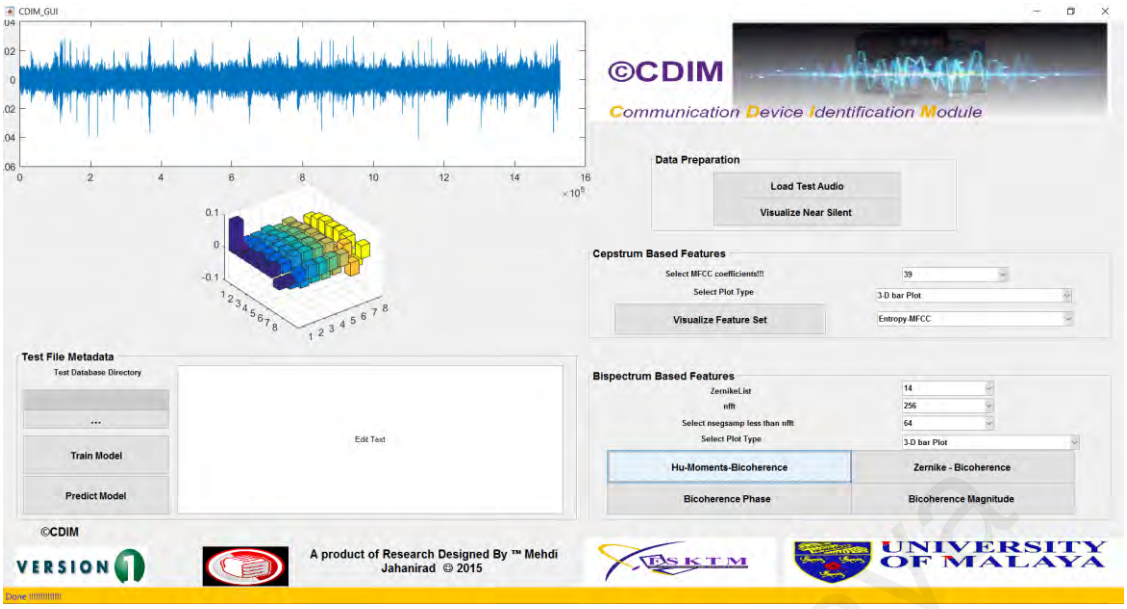


Figure 6.18: Visualizing the Hu-Moments-Bicoherence Using the 3-D Bar Plot

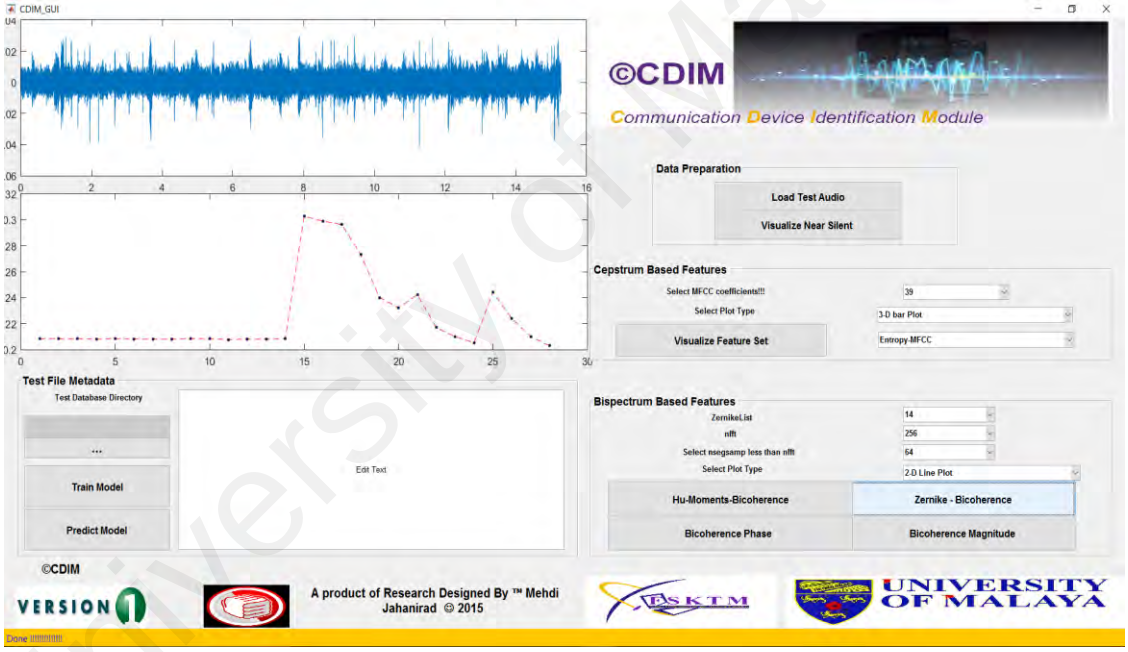


Figure 6.19 : Visualizing the Zernike – Bicoherence using the 2-D Line Plot

Figure 6.17 illustrates drop-down menu that allows a user to select the value of nsegsamp to estimate bicoherence spectrum. After selecting the required parameters as well as the plot type, the user has to press any of four buttons, which are ***Hu-Moments-Bicoherence***, ***Zernike – Bicoherence***, ***Bicoherence Phase*** and ***Bicoherence Magnitude*** in order to draw the result in the plot. It should be noted that for the ***Hu-Moments-Bicoherence*** and ***Zernike – Bicoherence*** user has the option to draw 3-D Bar Plot or

2-D Line Plot, as shown in Figure 6.18 and Figure 6.19, respectively. Using a different approach, for the ***Bicoherence Magnitude*** and ***Bicoherence Phase***, the program draws the contour plot only. Figure 6.20 and Figure 6.21 visualize the contour plot of the Bicoherence magnitude and phase, respectively.

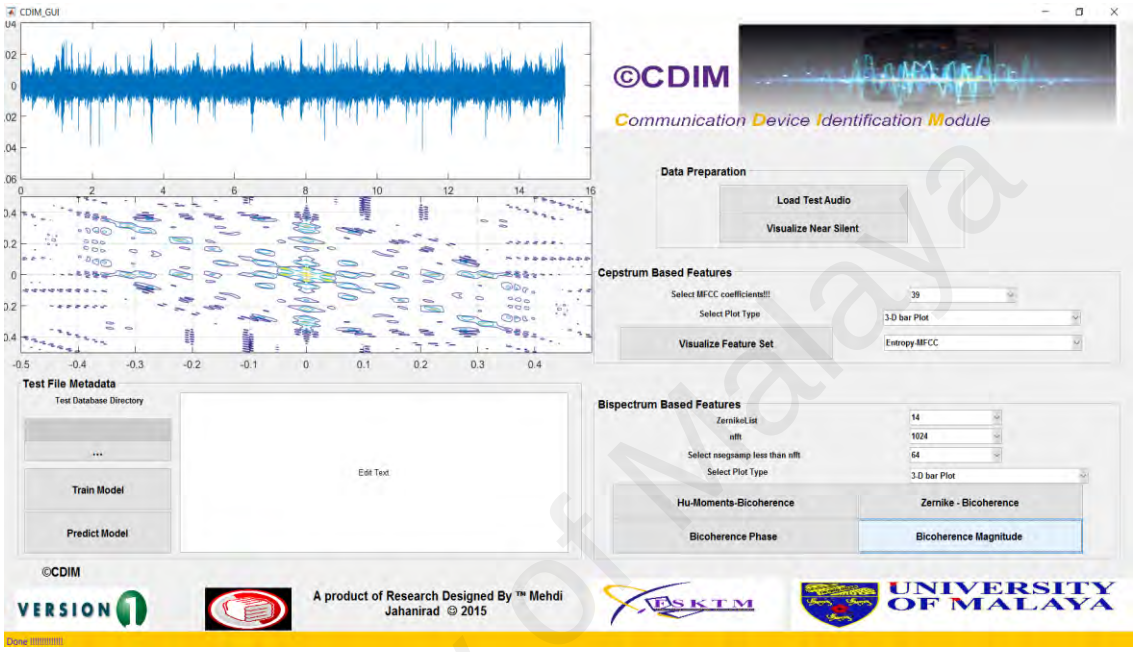


Figure 6.20 : Visualizing the Bicoherence Magnitude Using the Contour Plot

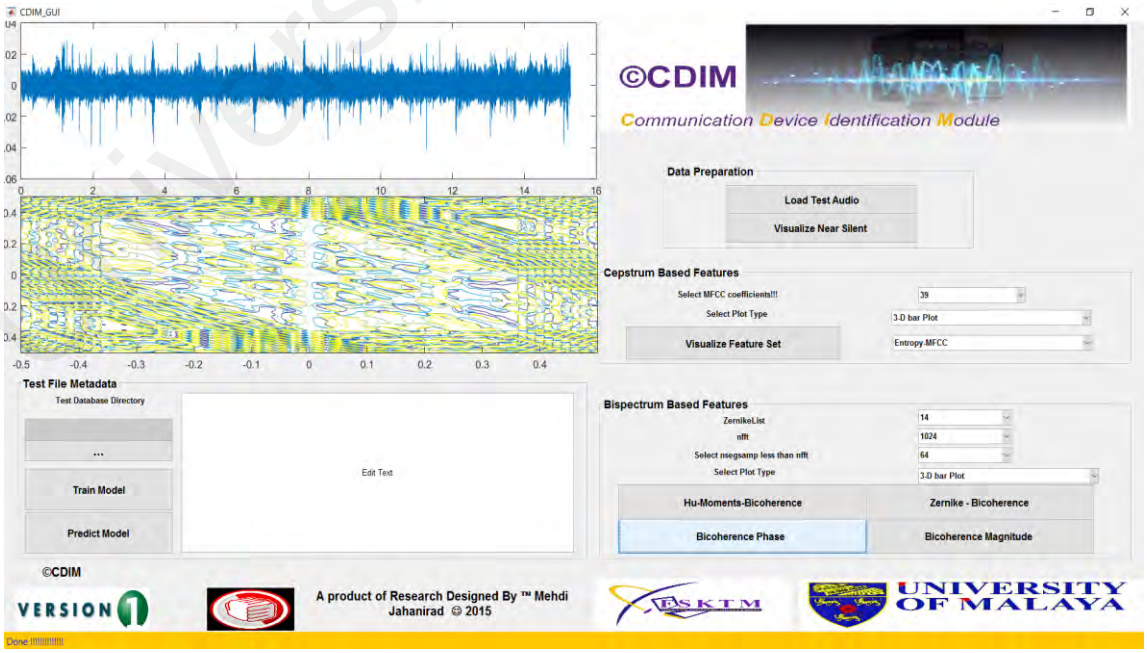


Figure 6.21: Visualizing the Bicoherence Phase Using the Contour Plot

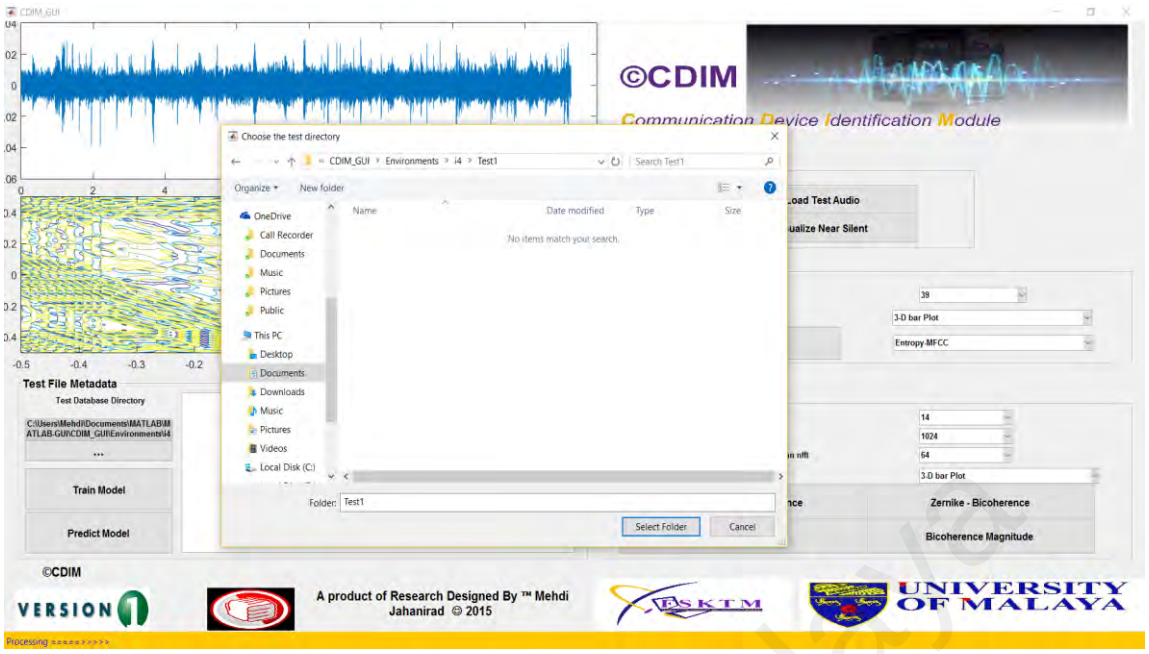


Figure 6.22: The Steps of Introducing the Path for the Test Directory Folder

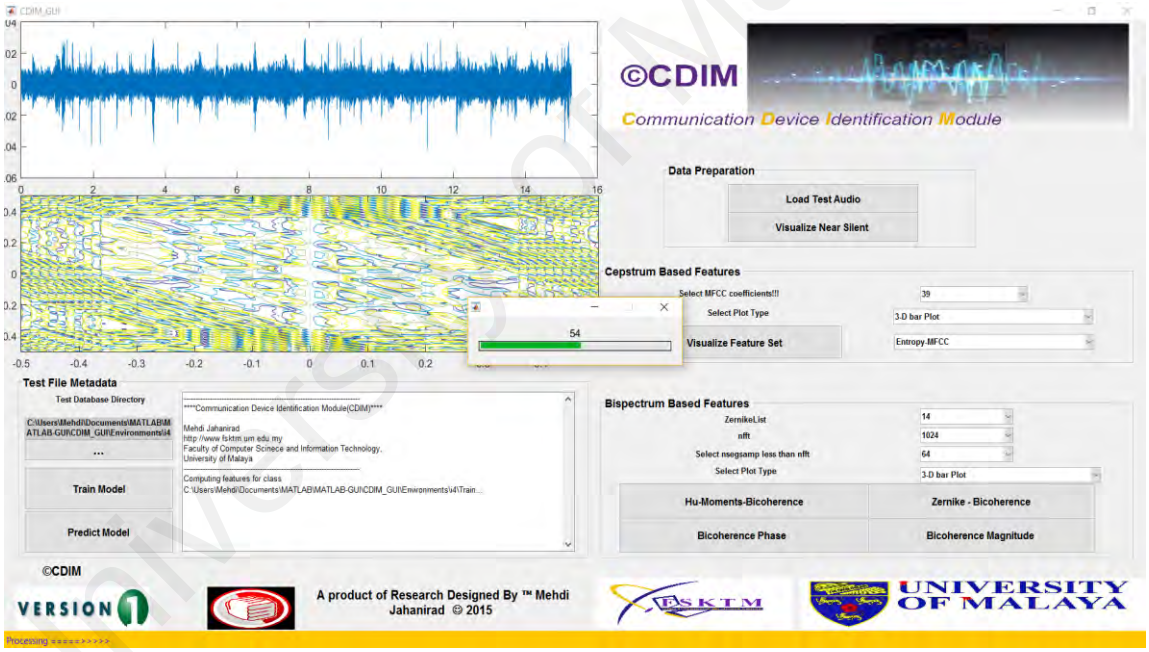


Figure 6.23 : The Training Model for the Multi-Class Classifier

6.3.5 Test File Metadata Identification

This part of the CDIM is considered as the main part, and it is the heart of this prototype. This part contains three important sections. The first section is ***Test Database Directory***, the second section is the ***Train Model***, and finally the third section is the ***Predict Model***. By pressing the *Test Database Directory* button, the user gives the path for the test directory. Figure 6.22 illustrates the steps for introducing the path to the test

directory folder. Furthermore, the second button in the *Test File Metadata* module is *Train Model* button, which starts its processing by pressing the button by the user through training the model based on the currently available classes in the training database, as shown in Figure 6.23. By completing the process, the new model will be restored on top of the pre-existing model.

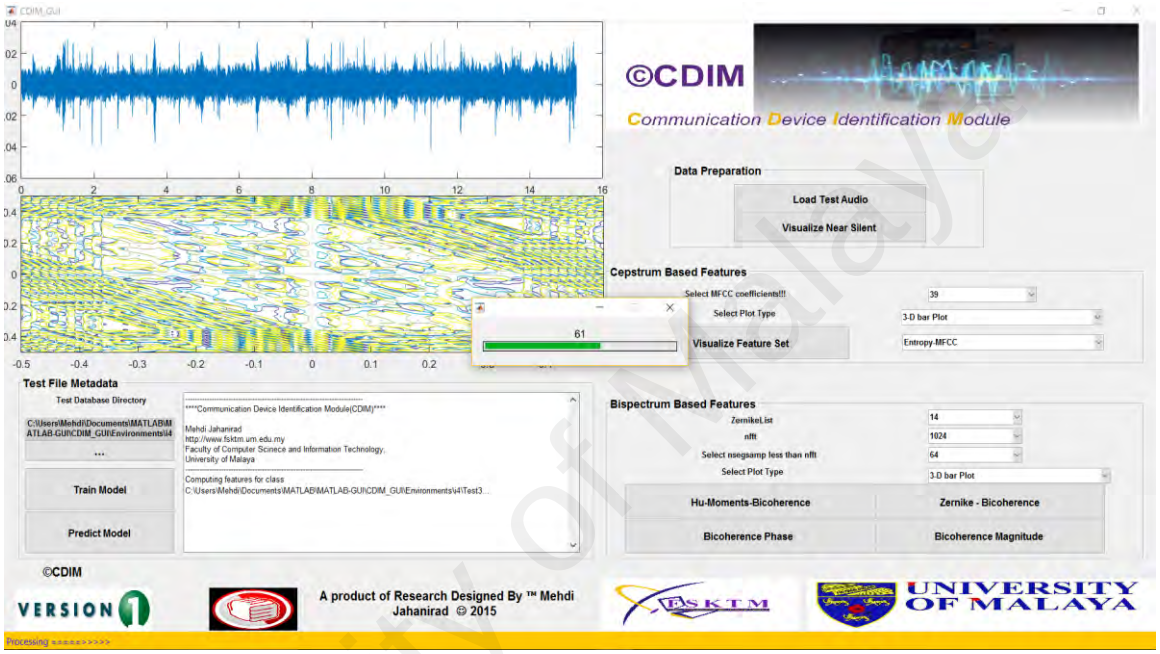


Figure 6.24 : Predicting the Test File Class Label

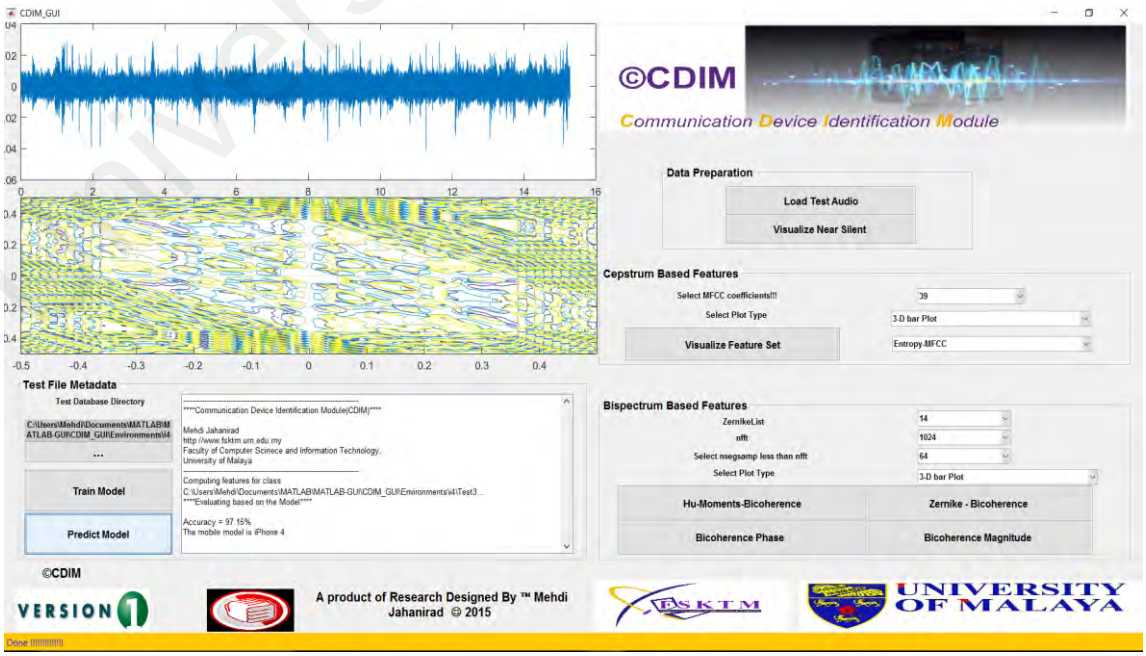


Figure 6.25: Create Test File Metadata Based on the Predicted Class Label

Finally, the third button in the *Test File Metadata* module is the *Predict Model* button which is pressed for predicting the test file class label. Figure 6.24 illustrates the processing of the *Predict Model* to identify the test file class label, whereas Figure 6.25 details the prediction results based on the prediction accuracy and the source mobile device model utilized for making the recorded call(s) in the test database directory.

6.3.6 Advantages and Limitations

In providing a flexible platform to forensic investigators for configuring, analyzing and making a wise decision using SVM classifier prediction results, the MATLAB GUI modules give the following advantages:

(a) *Analytic results*: The MATLAB GUI modules provide numerical and visual analytics solutions that give a strong understanding about simpleness of the optimizing acoustic features by using the spectral analysis techniques. Simplicity, easy-to-use, modifiable and adaptable modules that help optimizing feature extraction results allow potential users to perform spectral analysis on call recording signal. The MATLAB GUI modules offer different types of visualization and provide a wider perspective of the existing condition regarding statistical properties of the feature sets and the test file metadata.

(b) *Simplicity with graphical interfaces*: In replacement of performing numerical values, the MATLAB GUI modules simplify comparison and analytics by providing graphical interfaces in call recording signal spectrum and statistical properties of the feature sets with the aid of the usage of different colors and plots. The use of feature values in the 2-D line plots, color bars in the 3-D bar plot and contours in the contour plot allows potential users to interpret statistical properties of different feature sets; moreover, it gives the additional advantage to non-technical people to learn about different acoustic features.

(c) *Simple operation and user-friendly interfaces*: The use of MATLAB GUI modules provides user-friendly interfaces because the users have an ability to load test file, playback the call recording file(s) under investigation, change the configuration settings for different feature extraction techniques, build the model and analyze the feature extraction results from the test database directory and make a decision by using the predicted class labels. The MATLAB GUI along with its back-end applications can be stored and accessed in any platforms anywhere without any problem.

(d) *Adaptable setting to optimize the feature extraction process*: The MATLAB GUI modules provide an adaptability mode upon the module configurations that facilitate users to alter them dynamically, in order to optimize the acoustical features through spectral analysis techniques. With the optimized feature sets, the MATLAB GUI modules summarize the test file metadata in the text box by using the predicted class labels and the corresponding identification accuracy.

(e) *Multifunction spectral analysis tool*: The MATLAB GUI modules allow to playback and study the signal spectrum and its near-silent segments, in addition to the configuration, computing and analysis of the cepstrum- and bispectrum-based features. Thus, the CDIM can be used as a multifunction spectral analysis tool in any audio forensic investigation such as audio authenticity and interpretation.

(f) *Practical and cost effective*: With the customizable and adaptability of the MATLAB GUI modules, the training database can be expanded, contracted or replaced at any time, and the new model is built and updated the old model. A practical solution for law enforcement agencies, to extend the usage to have a source mobile device model of any unknown call is considered cheap and very cost effective.

In addressing the advantages of the MATLAB GUI modules, they also took over some limitations as follows:

(a) *Applications Dependent*: As the MATLAB GUI modules are created using GUI design environment (GUIDE) tools, they rely on the efficiency of the GUIDE tools itself. Although MATLAB GUI is easy to design it is limited in application compared to the commercial products.

(b) *Inherit other vulnerabilities*: Due to the use of VoIP and cellular call recordings received from different environments and recorded with different stationaries, the CDIM prototype is vulnerable with the call recordings collected under poor Wi-Fi strength, lack of cellular network coverage, interference, signal loss, and strong background noise. In addition to that, the CDIM prototype is also vulnerable to the other vulnerabilities such as stationary device and the recording software.

As a result, to provide a better performance in the future, it is important to address the limitations using other approaches and countermeasures.

6.4 Chapter Summary

This chapter established the implementation phase of the proposed framework by providing some examples and snapshots from Communication Device Identification Modules. The details of its modules, system architecture, state diagrams and MATLAB GUI modules have been presented to show how they act together.

The main purpose of exhibiting and explaining the aspects of the modules is to provide a better understanding of how the proposed framework works, and how its internal modules are affected by the external resources such as the call recording dataset. Due to time constraints, it was impossible to implement the full operation of some modules, such as configuring the training and testing database directory for different call recording subsets. The limitation, among others, is argued in the subsequent chapter.

CHAPTER 7: CONCLUSION

This chapter outlines the study by revisiting the research findings. It emphasizes on the most important findings, in addition to its limitations. Subsequently, it examines the capability of alternative studies in the domain, presenting how the proposed framework could be improved in the future.

7.1 Achievements of the Study

The study began with investigating different types of audio source device identification approaches, exploring issues regarding the source communication device identification based on recorded call. It also introduced a control system model for the mobile device transmission system to identify the contaminating influences on call recording signal. The study proposed a novel framework that controls or eliminates such influences and facilitates spectral analysis techniques in order to optimize acoustic features for source mobile device identification. Several feature extraction and machine learning techniques were explored, and their capabilities evaluated in order to satisfy the aim of this study.

More accurately, the overall aim of this study is to produce a novel framework to optimize acoustic features using spectral analysis techniques for source mobile device identification. Within the proposed framework, which consists of the experiments in addition to the MATLAB GUI modules of the CDIM prototype, this study has been successful. Details are as follows:

(a) *An audio source device identification model for Forensic Categorization of communicating mobile devices*: This study has established a new direction known as communication device identification based on recorded calls received from mobile devices. This approach assumes that call recording signals contain intrinsic artifacts

of both transmitting- and receiving-end devices. Using a new perspective, the establishment of the model allows the proposed framework to be designed to capture the transmitting device artifacts from calls that traverse cellular and VoIP networks (see Chapter 3). The model helps the proposed framework to work flawlessly in identifying the individual mobile device units, mobile device models or brands, each of which has its own unique characteristics. A survey study was conducted to investigate the acoustic features and machine learning techniques when applied to audio source device identification (see Chapter 2). To show the feasibility and suitability of the model, in addition to other models, several experiments were conducted and their results suggested positive outcomes (see Chapter 5).

(b) *Challenges of controlling the influences contaminating mobile device response function*: In Chapter 3, this study established a critical analysis of a control system model for wireless communication signal processing pipeline when addressing the main influences of the undesired sources such as the speaker, environment and communication channel contaminating the influence of the mobile device response function on call recording signal. With an aim to establish a framework to discriminate the mobile device response function from other influences, several issues were discovered in the data preparation and feature extraction process. By presenting the strength and weakness of these issues, several strategies were identified, which address the limitations of the state-of-the-arts feature sets. Hence, by optimizing the acoustic features, they are more robust against such influences in the source mobile device identification process.

(c) *Cepstral analysis techniques to optimize acoustic features*: The study proposed cepstral analysis techniques to assist the feature extraction process. The concept of the cepstrum estimation is developed as a special case of homomorphic filtering. Homomorphic filtering uses a logarithm to transform convolved or nonlinearly related

signals to additive signals and then to process them by linear filters. Motivated by this, the proposed feature extraction algorithm used entropy of Mel-cepstrum coefficients. Eventually, the study utilized *Tsallis entropy* along with *Shannon entropy* in an attempt to increase robustness and the discriminating ability of the *entropy-MFCC* feature set (see Section 4.1.3).

(d) *Higher-order spectral analysis techniques to optimize acoustic features*: The study proposed to model mobile device response function through analysis of the higher-order cumulant spectra of near-silent segments of the call recording signal (assumed to be stochastic). Theoretically, a transform to a higher-order cumulant domain suppresses the Gaussian noise, reconstructs the true phase and magnitude response of signals, and in addition, it detects and characterizes nonlinearities in call recording signal. Therefore, bicoherence is introduced as an effective function to identify and characterize nonlinearities in call recording signal through phase relations of their harmonic components. As a result, the *ZMs of the bicoherence magnitude and phase* spectrum are used as intrinsic mobile device fingerprints for source mobile device identification (see Section 4.1.4).

(e) *A novel framework which identifies the source mobile device unit, model, and brand*: Using models and multi-strategies in pre-processing and data preparation as well as the feature extraction and analyses, this study proposed a novel framework to address the source mobile device identification process in a systematic way. With the aid of the LIBSVM package library and its predefined functions, it is possible to train the SVM classifier with proposed features extracted from call recordings collected corresponding to each individual mobile device, mobile device model or brand under investigation. Furthermore, the trained machine learning algorithm can identify the corresponding source mobile device of a call recording under investigation through detecting the closest match of feature values (see Section 4.1.5).

(f) *Comprehensive evaluation stages for the proposed framework*: In addressing the source mobile device identification, the proposed framework outlined several models and strategies. These need to be evaluated. The purpose of the evaluation is to examine the proposed framework and to decide whether it is sufficiently applicable to facilitate the inter- and intra-mobile device model identification using the large database of VoIP and cellular call recordings collected in different setup conditions. The evaluation was performed in five different phases: the proposed framework was analyzed in terms of its effectiveness and performances when related to the models and strategies selected (see Chapter 5). The progressive results presented from one phase to another demonstrated the suitability and feasibility of the proposed framework for optimizing the acoustic features for source mobile device identification. More importantly, the evaluation phases satisfied the main criteria to support open set evaluation, in particular, the ability of the framework to assign the source mobile device model of the call recordings from unknown sources to the outlier class, besides to operate with high identification accuracy and robustness in investigating the call recordings that their source mobile device model is known by the training database.

(g) *Implementation of the proposed framework*: To extend the study of the feasibility of the proposed framework, and demonstrate its practical application in the forensic investigation with the blind mode, a proof-of-concept study was designed and realized (see Chapter 6). As an extension to the evaluation study, the implementation stage has developed a MATLAB-based system that focuses on the MATLAB GUI modules of the proposed framework. In order to illustrate the implementation stage, the detail of the proposed framework was presented using several modeling languages. These included case diagrams and state diagrams, as well as some snapshots of the prototype pages.

To conclude, it is inferred therefore that this study has achieved its aims and objectives as stated in Chapter 1.

7.2 Limitations of the Study

The discussions of the previous chapters have shown that this research has adequately achieved its aims and objectives: the establishment of a novel framework to use when optimizing acoustic features for source mobile device identification. However, a number of limitations and challenges were encountered during the study, and they are listed here for future reference:

(a) *Intra- and Inter-model similarity*: In conducting the experiments during the evaluation Phase II (see Section 5.3), this study found some theoretical limitations. The practical investigations suggested employing optimized entropy-MFCCs for source mobile device model identification because it showed larger inter-model distances in compare to ZMBic feature set. Meanwhile, both optimized entropy-MFCC and ZMBic feature sets showed considerable intra-model distances, which also allows the strong individual source mobile device identification. However, for source mobile device model identification, it is important to discriminate among different mobile device models instead of individual devices. Hence, the classifier performance might be reduced when the training and testing data instances are received from different mobile device units of the same model. Because of this limitation, it is suggested that, in future studies, an entropy-MFCC feature set needs to be optimized further, in order to reduce the intra-model distances and increase the inter-model distances.

(b) *Network Connection, Coverage, and Speed*: Although the results presented in this study were considered a reasonable result, they were also influenced by the different network connections (i.e. WLAN, WAN), Wi-Fi strength and coverage as well as the random effects of the wireless communication channel. For example, the Wi-Fi will

be automatically shifted to cellular data when Wi-Fi connectivity is poor; the quality of this connection also varies with the cellular data speed. This limitation increases the variance of the feature values with respect to different call recordings corresponding to the same mobile device unit. This causes an overfitting problem. Therefore, in order to reduce the limitation and produce the open set evaluation in practical forensic investigations despite these influences, it is suggested to consider methods to overcome an overfitting problem such as regularization, which forces the magnitude of the parameters to be smaller.

(c) *Data Collection and Test Setup*: Some limitations and drawbacks exist in the dataset: (a) some files were initially recorded in '.mp3'format, meanwhile converting the files from '.wav' to '.mp3' does not necessarily help in improving the quality of a signal, as the former is a lossy format; (b) although there were different devices considered and tested, for every device a limited number of communication software (Skype) was included; (c) although the test setup aimed to control the environmental effects such as echo and reverberations (by recording all calls in four specific environments), it is impossible to completely control such effects without having the soundproof experimental setup.

(d) *Multi-class open set recognition*: The evaluation study optimized the source mobile device identification framework based on the estimation of the mobile device intrinsic fingerprints only. The time constraint did not permit the development of the more sophisticated open set classifier. The proposed framework built the multi-class SVM classifiers, which should be retrained if any data or class were added to the training database. By contrast, OCC approach should be more practical, cost effective and easier solution to the source mobile device model identification problem. This is because the OCC approach characterizes only the target class, and only data instances of the target mobile device model are required during training process of OCC model.

(e) *Processing Time*: The evaluation study implies that the machine learning based source mobile device identification approaches require a large group of call recordings and a large set of features that can perfectly reflect different characteristics of the wireless communication signal processing pipeline. The large dimensionality of call recordings for training a large group of mobile device models and feature sets introduces extensive computation time for real-time scenarios. Similarly, the speech enhancement as well as the estimating the bicoherence and ZMs require too much time and a large amount of space for storage. In spite of that, processing time remained as an immense challenge for source mobile device identification. Future work should, therefore, consider minimizing part of these difficulties by implementing signal processing in distributed computing environment.

(f) *MATLAB GUI-based Modules*: The study extended the implementation stage to develop the CDIM prototype by using the MATLAB GUI modules. As the MATLAB GUI modules are created using GUI design environment (GUIDE) tools, their performance relies on the efficiency of the GUIDE tools itself. Although MATLAB GUI is easy to design it is limited in its applications compared to the commercial products. Therefore, future work should focus on converting the MATLAB GUI to stand alone program by adopting the object-oriented based environment.

7.3 Suggestions and Scopes for Future Work

A number of suggestions for future work outside the scope of this study have been identified. Several issues have arisen, and they are as follows:

(a) *Building new feature sets*: As mentioned above, the optimized entropy-MFCC used in the evaluation study and implementation phase as the feature set for source mobile device model identification still shows considerable intra-mobile device model distances because it characterizes the individual mobile device response function.

Given this limitation, it would be beneficial that future works be directed towards building new, better features suitable for source mobile device model identification with respect to the characteristics that are common to mobile devices of the same model.

(b) *Regularization and* variance: The results of the open set evaluation in the fifth phase sometimes demonstrated false detection due to the use of VoIP call recordings collected in the uncontrolled test setup with mobile devices connected to the different network. Further analysis on applying regularization to reduce large variances in the dataset due to such variations would be useful, especially in dealing with real-world call recording dataset. This investigation is significant because it can be used to strengthen the proposed framework as well as to improve the source mobile device identification strategy in responding to false detections.

(c) *One-class open set classification*: One of the limitations mentioned in the previous section was the use of multi-class open set classifier for the large call recording database. Further studies could use other approaches, such as OCC to characterize only the target class in which only data instances of the target mobile device model are required during training process of OCC model.

(d) *Distributed computing environment*: The other recommendation for future studies is to consider executing source mobile device identification in a distributed computing environment. For example, it is possible to improve the computational efficiency of the feature extraction and machine learning classifiers that are carried out by applications such as distributedWeka and Mahout on top of Hadoop.

7.4 Summary-The Future for Source Mobile Device Identification

Communication devices such as mobile devices are supplied with built-in signal processing components such as mixers and power amplifier devices that enable wireless transmission of audio signals. The study has presented a novel framework that focuses on

the process of identifying the brand/model/individual communication device rather than the recording device through its recorded call. The framework helps the forensic investigators to interpret the result automatically. For example, it is possible to identify if the query call recording from an unknown source has been made from the suspect mobile device discovered during the investigation. In fact, the entire study has shown the advantages of using the proposed framework to optimize acoustic features for source mobile device identification, which not only focuses on building the classifier for individual source mobile device identification, but also provides a way to differentiate between mobile device models, assessing whether the test dataset is from the source known by the built classifier, or not, and allowing high identification accuracy in response to them. Most importantly, the study has focused on improving the identification accuracy and robustness of source mobile device identification, in order to provide a way to handle call recordings from real-world forensic investigations. The important concept behind the proposed framework is the spectral analysis and machine learning techniques, which include data selection, pre-processing, feature extraction and analysis as well as the decision making.

With models and strategies such as cepstrum and higher-order spectral analysis, the adaptation of the proposed framework in source mobile device identification has given a new perspective in data selection, pre-processing as well as in optimizing acoustic features, with generic performance metrics introduced in the proposed framework. Literally, the future of an automatic source mobile device identification is a step forward from the entropy-MFCC and ZMBic feature sets proposed in this study, because with all its positive results, it contributes significantly to the identification of real-world call recordings despite the existence of contaminating influences such as different speakers, environments, stationaries as well as the communication channel. It means that it eliminates speech influences and, at the same time, shows robustness against the

environment, stationaries and communication channel influences. Furthermore, the use of MATLAB GUI modules in visualizing different models and strategies of the proposed framework means that automatic identification mode can be employed. Additional visualization studies would be beneficial, as they would be able to summarize several source mobile device model identification results based on different parameter settings for all cepstrum-based and bispectrum-based feature sets (i.e. not just the entropy-MFCC results).

In reality, the future of the blind source mobile device identification from the recorded call in audio forensics still relies on the standards and practices of the legal system, whereby the evidence is usually acquired as part of a civil or criminal law enforcement investigation or as part of the official inquiry into an accident or another civil incident. Proving the admissibility of the proposed method is required in the court of law in order to prove that the applied techniques are unbiased, reliable, nondestructive and widely accepted by the experts in the field. As a result, because of these special legal considerations, it is hoped that one day, the proposed framework becomes admissible for the forensic investigations in practice.

REFERENCES

Abbas, O. A. (2008). Comparison between data clustering algorithms. *Int. Arab J. Inf. Tec., 5*(3), 320-325.

Alam, M. J., Ouellet, P., Kenny, P., & O'Shaughnessy, D. (2011). *Comparative evaluation of feature normalization techniques for speaker verification.* Paper presented at the Proc. NOLISP.

Arce, G. R., & Hoboken, N. (2005). *Nonlinear Signal Processing*. USA: J. Wiley & Sons.

Balasubramaniyan, V. A., Aamir, P., Ahamad, M., Hunter, M. T., & Trayno, P. (2010). *PinDr0p: using single-ended audio features to determine call provenance*. Paper presented at the Proc. CCS, New York, NY, USA.

Beigi, H. (2011a). *Fundamentals of Speaker Recognition*. New York, NY, USA: Springer.

Beigi, H. (2011b). *Fundamentals of Speaker Recognition*. New York, NY, USA: Springer.

Beigi, H. (2011c). Information Theory *Fundamentals of Speaker Recognition* (pp. 265-299). New York, NY, USA: Springer.

Beigi, H. (2011d). Probability Theory and Statistics *Fundamentals of Speaker Recognition* (pp. 239-247). New York, NY, USA: Springer.

Bhatt, C. A., & Kankanhalli, M. S. (2011). Multimedia data mining: state of the art and challenges. *Multimed Tools Appl, 51*, 35-76.

Bingham, E., & Mannila, H. (2001). *Random projection in dimensionality reduction: applications to image and text data.* Paper presented at the Proceedings of the 7th ACM International Conference on Knowledge Discovery and Data Mining.

Bogert, P. B., Healy, M. J. R., & Tukey, J. W. (1963). *The Quefrequency Anaysis of Time Series for Echoes: Cepstrum, Pseudo-Autocovariances, Cross Cepstrum, and Shap Cracking.* Paper presented at the Proc. Symposium Time Series Analysis, New York.

Brew, A., Grimaldi, M., & Cunningham, P. (2008). An evaluation of one-class classification techniques for speaker verification. *Artificial Intelligence Review, 27*(4), 295-307.

Buchholz, R., Kraetzer, C., & Dittman, J. (2009). Microphone classification using Fourier coefficients *IH, LNCS* (Vol. 5806, pp. 235-246). Darmstadt, Germany: Springer Berlin Heidelberg.

Callrecorder. (2013). *MP3 Skype Recorder v.3.1*.

Campbell, W. M. (2002). *Generalized linear discriminant sequence kernels for speaker recognition.* Paper presented at the Proc. ICASSP, Orlando, Florida.

Campbell, W. M., & Assaleh, K. T. (2002). Speaker Recognition With Polynomial Classifiers. *IEEE Trans. Speech Audio Process., 2*(4), 205-212.

Campbell, W. M., Campbell, J. P., Gleason, T. P., Reynolds, D. A., & Shen, W. (2007). Speaker verification using support vector machines and highlevel features. *IEEE Trans. Acoust., Speech, Signal Process., 15*(7), 2085–2094.

Campbell, W. M., Sturim, D. E., & Reynolds, D. A. (2006). Support vector machines using gmm supervectors for speaker verification. *IEEE Signal Process. Lett., 13*(5), 308–311.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM : a library for support vector machines. *ACM Trans. Intell. Syst. Technol., 2*(3), 1-27.

Chen, N., & Xiao, H.-d. (2013). Perceptual audio hashing algorithm based on Zernike moment and maximum-likelihood watermark detection. *Digital Signal Processing, 23*(4), 1216-1227.

Chitode, J. (2008). *Digital Signal Processing* (1st ed.): Technical Publications.

Choudhury, M. A. A. S., Shah, S. L., & Thornhill, N. F. (2006). *Linear or nonlinear? a bicoherence based metric of nonlinearity measure.* Paper presented at the Proceedings of the 6th IFAC Symposium of SAFEPROCESS., Beijing, China.

Choudhury, M. A. A. S., Shah, S. L., & Thornhill, N. F. (2008). *Diagnosis of Process Nonlinearities and Valve Stiction: Data Driven Approaches*: Springer Berlin Heidelberg.

Coelho, G., da Silva, A., & Von Zuben, F. (2007). Evolving Phylogenetic Trees: A Multiobjective Approach. In M.-F. Sagot & M. T. Walter (Eds.), *Advances in Bioinformatics and Computational Biology* (Vol. 4643, pp. 113-125): Springer Berlin Heidelberg.

Cooper, A. J. (2008). Detection of copies of digital audio recordings produced using analogue interfacing. *International Journal of Speech Language and the Law, 15*(1), 67-95.

Cooper, A. J. (2009). An automated approach to the Electric Network Frequency (ENF) criterion: theory and practice. *International Journal of Speech Language and the Law, 16*(2), 193-218.

Cooper, A. J. (2011). Further considerations for the analysis of ENF data for forensic audio and video applications. *International Journal of Speech Language and the Law, 18*(1), 99-120.

Costa, F. d. O., Silva, E., Eckmann, M., Scheirer, W. J., & Rocha, A. (2014). Open set source camera attribution and device linking. *Pattern Recognition Letters, 39*, 92-101. doi: 10.1016/j.patrec.2013.09.006

Dhanalakshmi, P., Palanivel, S., & Ramalingam, V. (2011). Classification of audio signals using AANN and GMM. *Applied Soft Computing, 11*(1), 716-723.

Dileep, A. D., & Sekhar, C. C. (2012). Speaker recognition using pyramid match kernel based support vector machines. *Int J Speech Technol, 15*, 365–379.

Eskidere, Ö. (2014a). Identifying acquisition devices from recorded speech signals using wavelet based features. *Turkish J. Electr. Eng. Comput. Sci.*

Eskidere, Ö. (2014b). Source microphone identification from speech recordings based on a Gaussian mixture model. *Turkish Journal of Electrical Engineering & Computer Sciences, 22*(3), 754–767.

FaNT (Producer). (2015). Filtering and noise adding tool (faNT). Retrieved from http:/ /dnt .kr.hs-niederrhein .de /download .html.

Farid, H. (1999). Detecting digital forgeries using bispectral analysis: MIT AI Memo.

Farokhi, S., Sheikh, U. U., Flusser, J., & Yang, B. (2015). Near infrared face recognition using Zernike moments and Hermite kernels. *Information Sciences, 316*(0), 234-245.

FengJuan, G., ShuQian, S., & XiaoHui, W. (2010). *Using one-class svms and mp for audio recognition of action scenes.* Paper presented at the Proceedings of the 2nd International Workshop on Education Technology and Computer Science.

Freire, I. L., & Apolin´ario, J. e. A. (2010). *Gunshot detection in noisy environments.* Paper presented at the Priceedings of the 7th International Telecommunications Symposium.

Garcia-Romero, D. (2012). *Robust Speaker Recognition based on Latant Variable Models.* (Doctor of Philosophy), University of Maryland, College Park, USA. (UMI 3543277)

Garcia-Romero, D., & Epsy-Wilson, C. Y. (2010). *Automatic acquisition device identification from speech recordings*. Paper presented at the Proc. ICASSP, Dallas, Texas.

Garcia-Romero, D., & Espy-Wilson, C. (2010). Speech forensics: Automatic acquisition device identification. *J. Acoust. Soc. Am., 127*(3).

Garcia-Romero, D., & Espy-Wilson, C. Y. (2009). Automatic acquisition device identification from speech recordings. *J. Acoust. Soc. Am., 125*(2530).

Garg, R., Varna, A. L., Hajj-Ahmad, A., & Wu, M. (2013). "Seeing" ENF: Power-Signature-Based Timestamp for Digital Multimedia via Optical Sensing and Signal Processing. *Ieee Transactions on Information Forensics and Security, 8*(9), 1417-1432.

Garofolo, J., Lamel, L., Fisher, W., Fiscus, J., Pallett, D., Dahlgren, N., & Zue, V. (1993). TIMIT Acoustic-Phonetic Continuous Speech Corpus. Philadelphia: Linguistic Data Consortium.

Geetha, S., Ishwarya, N., & Kamaraj, N. (2010). Evolving decision tree rule based system for audio stego anomalies detection based on Hausdorff distance statistics. *Information Sciences, 180*, 2540–2559.

Gerkmann, T., & Hendriks, R. C. (2012). Unbiased MMSE-Based Noise Power Estimation With Low Complexity and Low Tracking Delay. *IEEE Audio, Speech, Language Process., 20*(4), 1383-1393.

Gharaibeh, K. M. (2011). Introduction *Nonlinear Distortion in Wireless Systems* (pp. 1-20): John Wiley & Sons, Ltd.

Giannakopoulos, T. (2010). A method for silence removal and segmentation of speech signals, implemented in Matlab (pp. 1-3). Greece: Department of Informatics and Telecommunications, University of Athens.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*: Addison-Wesley Longman Publishing Co., Inc.

Gupta, S., Cho, S., & Kuo, C. C. J. (2012). Current Developments and Future Trends in Audio Authentication. *Ieee Multimedia, 19*(1), 50-59.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations, 11*(1), 10-18.

Hammond, J., & White, P. (2008). Signals and Systems. In D. Havelock, S. Kuwano & M. Vorländer (Eds.), *Handbook of Signal Processing in Acoustics* (pp. 3-16): Springer New York.

Hanilci, C., & Ertas, F. (2013). *Optimizing Acoustic Features for Source Cell-phone Recognition Using Speech Signals.* Paper presented at the Proc. IH&MMSec '13,, New York, NY, USA.

Hanilçi, C., Ertaş, F., Ertaş, T., & Eskidere, Ö. (2012). Recognition of brand and models of cell-phones from recorded speech signals. *IEEE Trans. Forensics Security, 7*(2), 635-634.

Hanilci, C., & Kinnunen, T. (2014). Source cell-phone recognition from recorded speech using non-speech segments. *Digital Signal Processing, 35*(0), 75-85.

Hao, L., Lewin, P. L., Hunter, J. A., Swaffield, D. J., Contin, A., Walton, C., & Michel, M. (2011). Discrimination of multiple PD sources using wavelet decomposition and principal component analysis. *IEEE Trans. Dielectr. Electr. Insul., 18*(5), 1702-1711.

Haque, M. A., & Kim, J.-M. (2013). An enhanced fuzzy c-means algorithm for audio segmentation and classification. *Multimedia Tools and Applications, 63*(2), 485-500.

Hasse, J., Gloe, T., & Beck, M. (2013). *Forensic identification of gsm mobil phones.* Paper presented at the Proceedings of the First ACM Workshop on Information Hiding and Multimedia Security, New York, NY, USA.

Hermansky, H. (1990). Perceptual linear predictive (plp) analysis of speech. *J. Acoust. Soc. America, 87*, 1738–1752.

Hinich, M. J. (1982). TESTING FOR GAUSSIANITY AND LINEARITY OF A STATIONARY TIME SERIES. *Journal of Time Series Analysis, 3*(3), 169-176.

Hofmann, A., Walsh, R., McCurdy, I., & Heintz, J. (2012). Digital Audio. In Joachim Heintz & I. McCurdy (Eds.), *Csound Floss Manual*. USA: Free Software Foundation.

Hsu, C.-W., & Lee, L.-S. (2009). Higher order cepstral moment normalization for improved robust speech recognition. *IEEE Audio, Speech, Language Process., 17*(2), 205-220.

Hu, M. (1962). Visual pattern recognition by moment invariants. *IRE Trans. Info. Theory, vol. IT-8*, 179–187.

Hubeika, V., Burget, P., andMatejka, L., & Schwarz, P. (2008). *Discriminative training and channel compensation for acoustic language recognition.* Paper presented at the Proc. Interspeech.

Ikbal, S., Misra, H., Hermansky, H., & Magimai-Doss, M. (2012). Phase Auto Correlation (PAC) features for noise robust speech recognition. *Speech Communication, 54*, 867-880.

Ikram, S., & Malik, H. (2010). *Digital Audio Forensics Using Background Noise.* Paper presented at the Proc. IEEE International Conference on Multimedia and Expo.

Indyk, P. (2006). Stable distributions, pseudorandom generators, embeddings, and data stream computation. *Journal of the ACM, 53*(3), 307-323.

Jahanirad, M., Wahab, A. W., & Anuar, N. B. (2014). Blind Source Computer Device Identification from Recorded Calls *Lecture Notes in Electrical Engineering* (Vol. 315): Springer.

Jenner, F. (Nov. 2011). Non-intrusive identification of speech codecs in digital audio signals. M.S. thesis, Computer Engineering. Dept., KGCOE, RIT. Univ., Rochester, New York, NY, USA: ProQuest.

Ji-Kai, C., Hao-yu, L., Shi-yan, Y., & Bao-quan, K. (2012). A new method for extracting transient signal feature in transmission system based on Tsallis wavelet entropy. *Advance Materials Research, 433-440*, 2417-2422.

Juan Garcia-Hernandez, J., Feregrino-Uribe, C., & Cumplido, R. (2013). Collusion-Resistant Audio Fingerprinting System in the Modulated Complex Lapped Transform Domain. *PLoS One, 8*(6).

Khan, S., Divakaran, A., & Sawhney, H. S. (2010). *Weapon identification across varying acoustic conditions using an exemplar embedding approach.* Paper presented at the Proceedings of the SPIE 7666, Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense IX.

Khanna, N., Mikkilineni, A. K., Martone, A. F., Ali, G. N., Chiu, G. T. C., Allebach, J. P., & Delp, E. J. (2006). A survey of forensic characterization methods for physical devices. *Digital Investigation, 3, Supplement*, 17-28.

Kinnunen, T., Saeidi, R., Sedlák, F., Lee, k. A., Sandberg, J., Hansson-Sandsten, M., & Li, H. (2012). Low-Variance multitaper MFCC features: a case study in robust speaker verification. *IEEE Audio, Speech, Language Process.*, *20*(7), 1990-2001.

Koçal, O. H., Yürüklü, E., & Avcıbas, I. (2008). Chaotic-Type Features for Speech Steganalysis. *Ieee Transactions on Information Forensics and Security, 3*(4), 651-661.

Koenig, B. E., & Lacey, D. S. (2009). Forensic Authentication of Digital Audio Recordings. *Journal of the Audio Engineering Society, 57*(9), 662-695.

Koenig, B. E., & Lacey, D. S. (2012). Forensic Authenticity Analyses of the Header Data in Re-Encoded WMA Files from Small Olympus Audio Recorders. *Journal of the Audio Engineering Society*, *60*(4), 255-265.

Koenig, B. E., Lacey, D. S., Grigoras, C., Price, S. G., & Smith, J. M. (2013). Evaluation of the Average DC Offset Values for Nine Small Digital Audio Recorders. *Journal of the Audio Engineering Society, 61*(6), 439-448.

Korycki, R. (2013). Time and spectral analysis methods with machine learning for the authentication of digital audio recordings. *Forensic Science International, 230*(1-3), 117-126.

Kotropoulos, C. (2013). *Telephone handset identification using sparse representation of spectral feature sketches.* Paper presented at the Proc. 2013 IEEE Conference.

Kraetzer, C. (Mai, 2013). *Statistical Pattern Recognition for Audio Forensics - Empirical Investigations on the Application Scenarios Audio Steganalysis and Microphone Forensics.* (PhD Thesis), Otto-von-Guericke-University Magdeburg, Germany.

Kraetzer, C., & Dittmann, J. (2007). *Mel-Cepstrum Based Steganalysis for VoIP-Steganography.* Paper presented at the Proceedings of the Security, Steganography, and Watermarking of Multimedia Contents IX, vol. 6505 of Society of Photo-Optical Instrumentation Engineers (SPIE) Electronic Imaging Conference Series, San Jose, CA, USA.

Kraetzer, C., & Dittmann, J. (2008). *Impact of feature selection in classification for hidden channel detection on the example of audio data hiding*. Paper presented at the Proceedings of the 10th ACM workshop on Multimedia and security, Oxford, United Kingdom.

Kraetzer, C., & Dittmann, J. (2010). *Improvement of information fusion-based audio steganalysis* Paper presented at the Proc. IS&T/SPIE 7542.

Kraetzer, C., Oermann, A., Dittmann, J., & Lang, A. (2007). *Digital audio forensics: a first practical evaluation on microphone and environment classification.* Paper presented at the Proc. MM&Sec Dallas, Texas.

Kraetzer, C., Qian, K., & Dittmann, J. (2012). *Extending a context model for microphone forensics*. Paper presented at the Proc. SPIE 8303, Burlingame, CA.

Kraetzer, C., Qian, K., Schott, M., & Dittmann, J. (2011). *A context model for microphone forensics and its application in evaluations*. Paper presented at the Proc. SPIE-IS&T, San Francisco. CA.

Kraetzer, C., Schott, M., & Dittmann, J. (2009). *Unweighted Fusion in Microphone Forensics using a Decision Tree and Linear Logistic Regression Models.* Paper presented at the Proc. ACM multimedia and security workshop.

Krishnamurthy, N., & Hansen, J. (2009). Babble noise: modeling, analysis, and applications. *IEEE Trans. Audio Speech Lang. Process., 17*, 1394–1407.

Kuenzel, H. J. (2013). Automatic speaker recognition with crosslanguage speech material. *International Journal of Speech Language and the Law*, *20*(1), 21-44.

Li, W., Xiao, C., & Liu, Y. (2013). Low-order auditory Zernike moment: a novel approach for robust music identification in the compressed domain. *EURASIP Journal on Advances in Signal Processing*.

Liu, F.-H., Stern, R. M., Huang, X., & Acero, A. (1993). *Efficient cepstral normalization for robust speech recognition.* Paper presented at the Proceedings of the Workshop on Human Language Technology.

Mahajan, V. N., & Dai, G.-m. (2006). Orthonormal polynomials for hexagonal pupils. *Optics Letters, 31*(16), 2462-2464.

Mahajan, V. N., & Dai, G.-m. (2007). Orthonormal polynomials in wavefront analysis: analytical solution. *Journal of the Optical Society of America A, 24*(9), 2994-3016.

Maher, R. (2007). *Acoustical characterization of gunshots.* Paper presented at the Proc. IEEE SAFE, Washington, D.C.

Maher, R. (2009). Audio forensic examination. *IEEE Signal Processes. Mag., 26*(2), 84-94.

Malik, H., & Mahmood, H. (2014). Acoustic environment identification using unsupervised learning. *Security Informatics, 3*(11), 1-17.

Malik, H., & Miller, J. W. (2012). *Microphone Identification Using Higher-Order Statistics*. Paper presented at the Audio Engineering Society Conference: 46th International Conference: Audio Forensics.

Martin, A., Doddington, G., Kamm, T., Ordowski, M., & Przybocki, M. (1997). *The det curve in assessment of detection task performance*. Paper presented at the Proc. EUROSPEECH.

Martinez, C. E., Goddard, J., Di Persia, L. E., Milone, D. H., & Rufiner, H. L. (2015). Denoising sound signals in a bioinspired non-negative spectro-temporal domain. *Digital Signal Processing, 38*, 22-31.

McLoughlin, I., Haomin, Z., Zhipeng, X., Yan, S., & Wei, X. (2015). Robust Sound Event Classification Using Deep Neural Networks. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on, 23*(3), 540-552.

Mitrović, D., Zeppelzauer, M., & Breiteneder, C. (2010). Chapter 3 - Features for Content-Based Audio Retrieval. In V. Z. Marvin (Ed.), *Advances in Computers* (Vol. Volume 78, pp. 71-150): Elsevier.

Molla, M. K. I., & Hirose, K. (2004). *On the effectiveness of MFCCs and their statistical distribution properties in speaker identification.* Paper presented at the Proc. VECIMS, Boston, MA.

Muhammad, G., & Alghathbar, K. (2013). Environment Recognition for Digital Audio Forensics Using MPEG-7 and Mel Cepstral Features. *International Arab Journal of Information Technology, 10*(1), 43-50.

Nagarajan, B., & Devendran, V. (2012). Vehicle Classification under Cluttered Background and Mild Occlusion Using Zernike Features. *Procedia Engineering, 30*(0), 201-209.

Nikias, C. L., & Petropulu, A. P. (1993). *Higher-order Spectra Analysis: A Nonlinear Signal Processing Framework*: PTR Prentice Hall.

NIST-SRE. (2006). 2006 NIST Speaker Recognition Evaluation (Web Download). from Multimodal Information Group http://www.itl.nist.gov/iad/mig//tests/sre/2006/index.html

NIST-SRE. (2008). 2008 NIST Speaker Recognition Evaluation (Web Download). from NIST Multimodal Information Group http://www.itl.nist.gov/iad/mig//tests/sre/2008/index.html

NIST-SRE. (2010). 2010 NIST Speaker Recognition Evaluation. from NIST Multimodal Information Group http://www.itl.nist.gov/iad/mig//tests/sre/2010/index.html

Noisex-92 (Producer). (2015). Retrieved from http :/ /www.speech .cs .cmu.edu /comp .speech /Section1 /Data /noisex .html

Oermann, A., Lang, A., & Dittmann, J. (2005, 2005). *Verifier-tuple for Audio-Forensic to Determine Speaker Environment*

ACM Multimedia and Security Workshop, New York, NY.

Ojowu, O., Jr., Karlsson, J., Li, J., & Liu, Y. (2012). ENF Extraction From Digital Recordings Using Adaptive Techniques and Frequency Tracking. *Ieee Transactions on Information Forensics and Security, 7*(4), 1330-1338.

Oppenheim, A. V. (1969). Speech Analysis-Synthesis System Based on Homomorphic Filtering. *J Acoust Soc Am, 45*(2), 458-465.

Pamela. (2013). Pamela for Skype — Professional Edition 4.8. from http://www.pamela.biz/en/

Panagakis, Y., & Kotropoulos, C. (2012a). *Automatic telephone handset identification by sparse representation of random spectral features*. Paper presented at the Proceedings of the on Multimedia and security, Coventry, United Kingdom.

Panagakis, Y., & Kotropoulos, C. (2012b). *Telephone handset identification by feature selection and sparse representations*. Paper presented at the Proc. WIFS, Tenerife.

Pardede, H. F., & Shinoda, K. (2012). *Non-extensive statistics for feature normalization in speech recognition.* Paper presented at the International Workshop for Statistical Machine Learning for Speech Processing, IWSML'2012, Kyoto, Japan.

Pawera, N. (2003). *Microphone Practice: Tips and Tricks for Stage and Studio* (4 ed.): PPVMEDIEN.

Peeters, G. (2004). A large set of audio features for sound description (similarity and classication) in the CUIDADO project (pp. 1-25). 75004 Paris, France: Ircam, Analysis/Synthesis Team.

Pitas, I., & Venetsanopoulos, A. N. (1990). Homomorphic Filters *Nonlinear Digital Filters* (Vol. 84, pp. 217-243): Springer US.

Rabaoui, A., Kadri, H., Lachiri, Z., & Ellouze, N. (2008). One-class svms challenges in audio detection and classification applications. *EURASIP Journal on Advances in Signal Processing*, 1-14.

Rabiner, L., & Juang, B.-H. (1993a). *Fundamentals of speech recognition*. Englewood Cliffs, N.J.: PTR Prentice Hall.

Rabiner, L., & Juang, B.-H. (1993b). Pattern-comparison techniques *Fundamentals of Speech Recognition* (pp. 141-238). Englewood cliffs, New Jersey: Prentice-Hall International, Inc.

Rao, T. S., & Gabr, M. M. (1980). A test for linearity and stationarity of time series. *Journal of Time Series Analysis, 1*(1), 145–158.

Reynolds, D. A. (1997). *HTIMIT and LLHDB: Speech Corpora for the Study of Handset Transducer Effects.* Paper presented at the Proc. ICASSP.

Ross, A., Nandakumar, K., & Jain, A. (2006). *Handbook of Multibiometrics*: Springer Verlag.

Saeedi, J., Ahadi, S. M., & Faez, K. (2015). Robust voice activity detection directed by noise classification. *Signal Image and Video Processing*, *9*(3), 561-572.

Sarikaya, R., & Hansen, H. (2000). High resolution speech feature parameterization for monophone-based stressed speech recognition. *Ieee Signal Processing Letters, 7*(7), 182-185.

Sarikaya, R., Pellom, B., & Hansen, H. (1998). *Wavelet packet transform features with application to speaker identification.* Paper presented at the Proc. of IEEE Nordic Signal Processing Symp.

Senan, N., Ibrahim, R., Nawi, N. M., Yanto, I. T. R., & Herawan, T. (2011). Rough set approach for attributes selection of traditional Malay musical instruments sounds classification. *Int. J. Database Theory Appl, 4*(3), 59-76.

Shannon, B. J., & Paliwal, K. K. (2003). *A comparative study of filter bank spacing for speech recognition.* Paper presented at the In Proceedings of the Microelectronic Engineering Research Conference.

Shannon, C. E. (1949). Communication in the Presence of Noise. *Proceedings of the Institute of Radio Engineers, 37*(1), 10-21.

Sharma, D., Hilkhuysen, G., Hilkhuysen, N., Naylor, P., Brookes, M., & Huckvale, M. (2010). *Data driven method for non-intrusive speech intelligibility estimation*. Paper presented at the Proc. EUSIPCO, Aarlborg, Denmark.

Sparavigna, A. C. (2015). On the Role of Tsallis Entropy in Image Processing. *International Scientific Research Journal, 1*(6), 16-24.

Starck, J., & Hilton, A. (2008). Model-based human shape reconstruction from multiple views. *Computer Vision and Image Understanding, 111*(2), 179-194.

Steinebach, M., Zmudzinski, S., & Petrautzki, D. (2012). Forensic Audio Watermark Detection. In N. D. Memon, A. M. Alattar & E. J. Delp (Eds.), *Media Watermarking, Security, and Forensics 2012* (Vol. 8303).

Stevens, S. S., Volkmann, J., & Newman, E. B. (1937). A Scale for the Measurement of the Psychological Magnitude Pitch. *J Acoust Soc Am, 8*(3), 185-190.

Suski II, W. C., Temple, M. A., Mendenhall, M. J., & Mills, R. F. (2008). Radio frequency fingerprinting commercial communication devices to enhance electronic security. *Int. J. Electron. Secur. Digit. Forensic, 1*(3), 301-322.

Swami, A., Mendel, J. M., & Nikias, C. L. (1995). *Higher-order Spectral Analysis Toolbox: For Use with Matlab*: math Works.

Swaminathan, A., Wu, M., & Liu, K. (2007). Nonintrusive components forensics of visual sensors using output images. *IEEE Trans. Inf. Forensics Security, 2*(1), 91-106.

Takala, T., & Hahn, J. (1992). Sound rendering. *SIGGRAPH Comput. Graph, 26*, 211-220.

Teague, M. (1980). Image analysis via the general theory of moments. *J Opt Soc Am.*(70), 920–930.

Terrell, G. R., & Scott, D. W. (1992). Variable kernel density estimation. *The Annals of Statistics, 20*(3), 295-307.

Theodoridis, S., & Koutroumbas, K. (2009). Chapter 7 - Feature Generation II. In S. Theodoridis & K. Koutroumbas (Eds.), *Pattern Recognition (Fourth Edition)* (pp. 411-479). Boston: Academic Press.

Tsallis, C. (1988). Possible generalization of boltzmann-gibbs statistics. *Journal of Statistical Physics, 52*, 479–487.

Tsallis, C. (2009). *Introduction to Nonextensive Statistical Mechanics- Approaching a Complex World*: Springer Science+Business Media.

Tzanetakis, G. (2002). *Manipulation, analysis and retrieval systems for audio signals.* (Doctor of Philosophy), Prinston Univ., Princeton, NJ (TR-651-02)

Valenzise, G., Gerosa, L., Tagliasacchi, M., Antonacci, F., & Sarti, A. (2007). *Scream and Gunshot Detection and Localization for Audio-Surveillance Systems.* Paper presented at the Proceedings of the Ieee.

Vaseghi, S. V. (2008). *Advanced Digital Signal Processing and Noise Reduction* (4 ed.). United Kingdom: WIELY.

Vu, H. Q., Liu, S., Yang, X., Li, Z., & Ren, Y. (2012). Identifying Microphone from Noisy Recordings by Using Representative Instance One Class-Classification Approach. *Journal of Networks, 7*(6).

Wang, X.-Y., Ma, T.-X., & Niu, P.-P. (2011). A pseudo-Zernike moment based audio watermarking scheme robust against desynchronization attacks. *Computers & Electrical Engineering, 37*(4), 425-443.

Wang, X.-Y., Yu, Y.-J., & Yang, H.-Y. (2011). An effective image retrieval scheme using color, texture and shape features. *Computer Standards & Interfaces, 33*(1), 59-68.

Witten, I. H., Frank, E., & Hall, M. A. (2011a). Chapter 2 - Input: Concepts, Instances, and Attributes. In I. H. Witten, E. Frank & M. A. Hall (Eds.), *Data Mining: Practical Machine Learning Tools and Techniques (Third Edition)* (pp. 39-60). Boston: Morgan Kaufmann.

Witten, I. H., Frank, E., & Hall, M. A. (2011b). Chapter 6 - Implementations: Real Machine Learning Schemes. In I. H. Witten, E. Frank & M. A. Hall (Eds.), *Data Mining: Practical Machine Learning Tools and Techniques (Third Edition)* (pp. 191-304). Boston: Morgan Kaufmann.

Witten, I. H., Frank, E., & Hall, M. A. (2011c). Chapter 8 - Ensemble Learning. In I. H. Witten, E. Frank & M. A. Hall (Eds.), *Data Mining: Practical Machine Learning Tools and Techniques (Third Edition)* (pp. 351-373). Boston: Morgan Kaufmann.

Witten, I. H., Frank, E., & Hall, M. A. (2011d). Data Transformation. In I. H. Witten, E. Frank & M. A. Hall (Eds.), *Data Mining: Practical Machine Learning Tools and Techniques* (Third ed., pp. 305-349). Boston: Morgan Kaufmann.

Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., & Yi, M. (2009). Robust Face Recognition via Sparse Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 31*(2), 210-227.

Xiang, Y., Natgunanathan, I., Peng, D., Zhou, W., & Yu, S. (2012). A Dual-Channel Time-Spread Echo Method for Audio Watermarking. *Ieee Transactions on Information Forensics and Security, 7*(2), 383-392.

Xu, R., & Wunsch II, D. (2005). Survey of clustering algorithms. *IEEE Trans. Neural Netw.*, *16*(3), 645-678.

Yang, R., Qu, Z., Huang, J., & Acm. (2008). *Detecting Digital Audio Forgeries by Checking Frame Offsets*.

Yang, R., Shi, Y. Q., & Huang, J. (2010). Detecting double compression of audio signal. In N. D. Memon, J. Dittmann, A. M. Alattar & E. J. Delp Iii (Eds.), *Media Forensics and Security Ii* (Vol. 7541).

Yang, R., Shi, Y. Q., Huang, J., & Acm. (2009). *Defeating Fake-Quality MP3*.

Zeng, Y., Lan, J., Han, C., Huang, K., Li, J., & Shi, X. (2014). Aircraft recognition based on improved iterative threshold selection and skeleton Zernike moment. *Optik - International Journal for Light and Electron Optics*, *125*(14), 3733-3737.

Zernike, F. (1934). Beugungstheorie des Schneidenverfahrens und Seiner Verbesserten Form, der Phasenkontrastmethode. *Physica*, *8*(1), 689–704.

Zheng, R., Zhang, S., & Xu, B. (2006). *A comparative study of feature and score normalization for speaker verification.* Paper presented at the Proceedings of the 2006 International Conference on Advances in Biometrics, Berlin.

Zilovic, M., Ramachandran, R., & Mammone, R. (1998). Speaker identification based on the use of robust cepstral features obtained from pole-zero transfer functions. *IEEE Trans. Speech Audio Process.*, *6*, 260-267.

Zolotarev, V. (1986). One dimensional stable distributions. *Translations of Mathematical Monographs, American Mathematical Society*, *65*, 307-323.

Zwicker, E., Flottorp, G., & Stevens, S. S. (1957). Critical Band Width in Loudness Summation. *Journal of the Acoustical Society of America*, *5*(29), 548–557.

LIST OF PUBLICATIONS AND PAPERS PRESENTED

1. **M. Jahanirad,** A. W. A. Wahab, N. B. Anuar, M. Y. I. Idris, M. N. Ayub, Blind source mobile device identification based on recorded call, Engineering Applications of Artificial Intelligence 36 (0) (2014) 320 – 331, 2014.
2. **M. Jahanirad,** N.B. Anuar, A. W. A. Wahab, An Evolution of Image Source Camera Attribution Approaches, Forensic Science International, Vol. 262, May 2016, Pages 242-275.
3. **M. Jahanirad,** A. Wahab, N. Anuar, Blind Source Computer Device Identification from Recorded Calls, in: H. A. Sulaiman, M. A. Othman, M. F. I. Othman, Y. A. Rahim, N. C. Pee (Eds.), Advanced Computer and Communication Engineering Technology, vol. 315 of Lecture Notes in Electrical Engineering, Springer International Publishing, 263–275, 2015.
4. **M. Jahanirad,** A. W. Abdul Wahab, N. B. Anuar, M. Y. Idna Idris, M. N. Ayub, Blind identification of source mobile devices using VoIP calls, in: Proceedings of the IEEE Region 10 Symposium, 486–491, 2014.
5. **M. Jahanirad,** Y. AL-Nabhani, R. Md. Noor, Security measures for VoIP application: A state of the art review, Scientific Research and Essays 6 (23) (2011) 4950–4959, 2011.
6. **M. Jahanirad,** Y. AL-Nabhani, R. Md. Noor, Comprehensive Network Security Approach: Security Breaches at Retail company- A Case Study, IJCSNS International Journal of Computer Science and Network Security 12 (8) (2012) 107–112, 20