DATA SCRUTINY FOR NEURAL NETWORK-BASED BURMESE SPEAKER IDENTIFICATION

WIN LAI LAI PHYU

UNIVERSITY OF COMPUTER STUDIES, YANGON

MARCH, 2024

Data Scrutiny for Neural Network-based Burmese Speaker Identification

Win Lai Lai Phyu

University of Computer Studies, Yangon

A thesis submitted to the University of Computer Studies, Yangon in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

March, 2024

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

…..…… ……..……

Date Win Lai Lai Phyu

ACKNOWLEDGEMENTS

First of all, I would like to thank the Union Minister, the Ministry of Science and Technology for granting me for full facilities support during the Ph. D Course at the University of Computer Studies, Yangon.

I would like to express my special appreciation and deepest gratitude to Dr. Mie Mie Thet Thwin, Former Rector of the University of Computer Studies, Yangon, for giving me kindness and morality supports.

Secondly, I would like to express very special thanks to Dr. Mie Mie Khin, the Rector, the University of Computer Studies, Yangon, for allowing me to do this research and giving me technical supports during the period of my research.

I would like to describe my special thanks to my supervisor, Dr. Win Pa Pa, Professor, Natural Language Processing Lab., University of Computer Studies, Yangon. She took care of me. Because of her blessings I got the confidence to manage this research and all other hardships encountered in my research journey. This research would not be possible without her supervision, invaluable guidance and constructive ideas. I really gratitude for her supports which lead me to this point. She helped me in developing skills to present the progress of research work and enhancing my interests in the area of Speaker Recognition.

I would like to thank and respect to Dr. Sabai Phyu, Professor, and former Dean of the Ph.D 11th Batch, University of Computer Studies, Yangon, for her clear guidance, inspiration and encouragement.

I would like to thank to Dr. Tin Thein Thwel, Professor and Dean of Ph.D 11th Batch, University of Computer Studies, Yangon, for her advice and encouragement during my research.

I would like to express my gratitude regard to Daw Aye Aye Khine, Professor and Head of English Department for her careful assistance from the language point of view and pointed out the correct usage in my dissertation.

I would like to extend my special thanks and appreciations to Dr. Khin Mar Soe, Professor, Head of Natural Language Processing Lab., University of Computer

Studies, Yangon, for her great patience, many insightful discussions, comments, suggestions and supports in everything.

Last but not least, I would like to express my indebtedness and gratitude to my beloved parents and siblings for their positive encouragement in morality, eternal love, endless support and always believing in me. They are always supporting and encouraging me during the research journey and also helping the financial supports.

ABSTRACT

This dissertation aims to investigate the data augmenting and scrutinizing methods in developing a speech dataset for text independent Burmese speaker identification in open-set case which means the test speaker may not pre-modeled and included in the classifier. The training acoustic models are built based on Gaussian Mixture Model-Universal Background Model (GMM-UBM) and Time Delay Neural Network (TDNN) model. The speech dataset for speaker identification is firstly constructed because there is no available speech dataset for speaker identification research in Burmese. The data are collecting from the two domains: the web-based news data and recorded daily conversations. By this dataset, state-of-the-art acoustic speaker models for Burmese speaker identification are constructed.

Speaker identification is the task of analyzing the speakers' characteristics in speech to exactly identify individuals. The identification task performs better when there is enough background training data. The sufficient amount of speech data collection is a very challenging task in a short time for building Burmese speaker identification system because Burmese language can be considered as an under resourced language due to its linguistic resource availability. For getting sufficient amount of background training data, MUSAN speech dataset is used as speech data augmenting. For high quality training data, many other scrutinized techniques are investigated. Among them, the two data scrutinizing methods: increasing the speech intensity in SNRs to 10 dB and downing the tempo factor 0.2 times without affecting the pitch of utterances are applied to the original speech dataset. Moreover, white noise-added dataset is also created from the original dataset in order to prove that any kinds of noise can cause trouble the identification performance. Mel Frequency Cepstral Coefficient (MFCC) is used to extract the speaker specific features as front-end processing. In this work, TDNN and GMM-UBM based acoustic speaker models are constructed based on original, scrutinized and white noise-added training data. It can indicate that the impacts of speech data quality in constructing speaker models by using scrutinized training data and points out the important role of speaker models in identification process. The speakers' identities are assessed with probabilistic linear discriminant analysis (PLDA) approach. The system performance is presented in the form of Equal Error Rate (EER) and detecting accuracy (Acc).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

CHAPTER 1
INTRODUCTION

The most advanced mean of conveying thoughts, desires, and emotions among human beings is speech, the most crucial tool for communication in our daily lives. Nowadays, speech processing is growing a branch of emerging applied areas of digital signal processing and plays the key role in many speech processing research areas such as Automatic Speech Recognition, Speech Synthesis, Speaker Recognition or something else. Speaker recognition is a type of biometric authentication technology. It is also known as voiceprint recognition with the branches of Speaker Verification and Speaker Identification.

In order to solve daily security problems and improve their speech signal processing technologies, researchers around the world have explored the speaker identification. It is a natural portrayal of human computer interaction (HCI) that naturally recognized the person's identity from their voiceprints. The sound of human beings involves numerous distinct acoustic characteristics which can be determined who they are. The vocal tract's formation structure is distinctive for everyone. There are two types of speaker identification: one where the text is needed and one where it's not. While text dependent speaker identification must say an exact identical phrase to determine who they are, text independent speaker identification has no limit or restrictions on the words spoken. It is more applicable and flexible applications in real world. The quality and appropriate volume for modeling speaker models are required in order to enhance the performance of the speaker identification scheme. Many variations can be also encountered in speaker identification. The duration of utterances is the first one of many variations. The long utterances can better recognize the corresponding speaker than the short utterances. Any kinds of noise can lead to hinder the identification process. It is the second variation. To prove the second variation, the assessments of speaker models constructed with white noise-added dataset are implemented in this dissertation. Accent (speaker specific facts) comes as the third variation. The system can easily identify these speakers, as long as the speaker has spoken a standard dialect or one which corresponds to the speech information contained in training. The last variation is the speech recorded conditions. Various noises can contain in the data because these are collected and came from

various sources. If the data quality is low, the system performance is degraded. To prove the last variation, data scrutinizing methods are applied to the speech dataset for being the data quality uniformly.

Universal background model (UBM) is constructed using audio samples from multiple speakers and an adaptive Maximum A Posteriori (MAP) is used to obtain individual speaker models. By using Expectation-Maximization (EM) algorithm, GMM parameters are trained [4]. Firstly, a diagonal covariance UBM is trained basically in UBM training. To obtain full covariance model, the expectation maximization is implemented with mixture weights and fixed means [17] per iteration. In recent years, Gaussian Mixture Models (GMMs) have been used for speaker identification. But, GMMs did not resolve the channel distortions that occur when its assumptions about the corresponding talker are not exactly identical when different speech signals are used under different recording conditions. To address this issue, Joint Factor Analysis (JFA) modeled speaker variability and channel variability as two separate subspaces. Nevertheless, the recognition process may not perform well because there may be useful speaker related information in session variability subspace. To deal with these problems, i-vector method evolved from GMM super vectors is used. To solve these problems, i-vector method developed by GMM super vectors is used. Vectors called i-vectors that occur in low-dimensional and smaller spaces are used to reduce the detecting time [2]. Compressing the information to a low dimensional vector and modeling the total variations in the training data are the main goals of i-vector based systems that collect statistics from spoken utterances. Moreover, changing the parameters like the Gaussian components number and vectors' dimensions can affect the system performance.

A network with many hidden layers, which contains many nonlinear units and a large output layer, is called a deep neural network (DNN). It has recently been used to obtain special features of speakers and relies on a lot of model training data, but as a result, it can produce accurate results even though it takes a lot of time to train the model. Time Delay Neural Networks (TDNN) are designed to express the relationship between input features in time and learn transfer invariant feature transformations, which is a beneficial architecture for DNN-based speakers and models phonetic information directly [15]. It can be considered the predecessor of a convolutional neural network. Due to its efficiency in capturing long-range temporal

context dependencies and its capacity to extract stronger speaker features, it improves x-vector training ability [18]. While the higher layer uses a wider temporal context, TDNNs implement the initial transforms are learned within narrow temporal contexts [19]. The three main components of TDNN architecture are feature learning, statistical pooling, and finally, the speaker classification.

Subspace covariance modeling allows a Gaussian PLDA backend coupled with fixed-length feature vectors to leverage multiple input feature frames. In subspaces, the model extracts strongly associated feature vectors using generative hierarchical probabilistic modeling. The computational cost and feature dimension can be reduced by PLDA. In addition, it can also improve detection rate [13]. It is applied for scoring in feature vector-based speaker detection. The projection and centering of representations is done with LDA (Linear Discriminant Analysis). Vector representations are taken by length normalization and modeled with PLDA after dimensionality reduction. PLDA scoring method is used as in [3, 20]. It used the batch-likelihood ratio between the target and test i-vectors as the scoring method, appraised the wrong detecting percentage with equal error rate (EER) for measuring the performance of identification engines [6] and with Detecting Accuracy (Acc) for measuring the performance of detecting rate.

1.1 Research on Speaker Recognition

Speaker recognition research has been accomplished by many researchers with their respective languages and the recognition rate has also been improved by adopting particular properties of their target language or exploring the new architecture. Speaker verification and speaker identification are the two main categories of speaker recognition. Claiming to be an exact identity with their voices respectively is called the speaker verification. It is one to one match because one speaker’s voice is matched to one template. Speaker identification determines as the identity recognizing who is speaking. It is one to N match where the voice is compared against N templates. Identification process is slower than the verification process because of the processing time of matching. Speaker identification has the two branches: text independent and text dependent. In text dependent identification, the speakers need to say exactly the same speech for determining who they are but text independent identification has no boundaries and restrictions on the words that

are talked. In this work, open set (independent of text and speaker) identification is implemented by applying particular features of Burmese language because open set systems are most applicable in real world applications and this research is for the aim of developing speaker identification in Burmese firstly.

1.2 Intentions of Research

Speaker Identification is trying to develop by many researchers for improving the technologies in their respective national languages. The main objectives of research are:

1. To develop an efficient and accurate Automatic Burmese Speaker Identification system as the first work on Burmese speech from various sources
2. To construct the Speech Dataset for Burmese Speaker Identification
3. To analyze the scrutinizing techniques for getting good quality speech data needed to use in building the acoustic models
4. To prove augmenting on Burmese Speech Dataset improves in Acoustic Modeling
5. To save time and cost through automatic authentication

1.3 Focus of Research

This research focuses on developing a neural network-based speaker identification system in Burmese. The focused works include the following:

1. Building Burmese speech dataset for the first time to apply in speaker identification system
2. Applying augmenting methods to increase the diversity and volume of data in speech dataset
3. Scrutinizing the collected speech dataset to get the higher quality of data
4. Analyzing the results from the three models built with the original, scrutinized and white noise-added speech datasets
5. Comparing the system performance from both sides of text dependent (ClosedTestSet) and text independent (OpenTestSet) assessments

1.4 Contributions of Research

This research has four main contributions.

1. The speech dataset needed for Automatic Burmese Speaker Identification system builds as the first contribution of this research. Speech dataset building is the first step in any automatic speaker identification research. As Burmese language is an under resourced language, there is no already built speech dataset like resourced rich language, English and cannot be freely available the speech dataset. It is also the first important requirement and essential for developing Burmese Speaker Identification research. The speech dataset is built by two ways: collecting recorded speech data in Web news from various sources and recording the daily conversational speech ourselves.
2. The second contribution is augmenting the existing training data in order to enhance the system performance. Data augmentation can find out more speaker-oriented information due to its increasing length of utterances (durability). By finding the speaker specific information more and more, the system performance improves more. There are many data augmentation techniques for enhancing the system performance. Contaminating the environmental noise, adding room impulse responses to reverberate the original audio, adding additive noises to corrupt the original audio with babble, music and general noise, speed perturbation, feature warping, pitch shifting and spectrogram augmentation, etc., are used to contaminate the data. In this work, adding room impulse responses to reverberate the original audio, and adding additive noises to corrupt the original audio with babble, music and general noise by MUSAN dataset are used for augmenting the training data.
3. The third contribution is to apply the data scrutinizing methods to the existing dataset to improve the acoustic models' performance. It is the main component of this work and these techniques can show the significant improvements of speaker identification system. The proposed scrutinized dataset is created by applying the intensity level

raises to 10 dB and decreasing the tempo down to 0.2 times (20%) in the original dataset. To create this scrutinized dataset, the intensity levels of -10 dB, -5 dB, 5 dB and 10 dB are firstly applied to the original raw dataset. And then, the results of these four types of datasets are compared with the original dataset's performance. Among them, the dataset with the intensity level of 10 dB outperforms the original dataset because the speech having reasonable loudness can recognize well than the lower loudness. But, the loudness of speech beyond human hearing perception damages the tone of speaker and speech spectrum. Moreover, analyzing in the tempo factor up and down to 0.2 times comparing with the original dataset is done. From these experiments, decreasing the tempo factor down to 0.2 times give better results than the original dataset because speaking slowly can recognize the speaker well and understand what they are saying in clarity. Therefore, setting the intensity raises to 10 dB and the tempo slows down to 0.2 times to the original dataset forms the scrutinized dataset giving the significant improvement in speaker modeling.

4. The final contribution is to contaminate the noise in the original data. Although white noise that helps to induce a more relaxed state is a pleasing sound for many people, adding it in the human speech can cause the disturbance. According to the experimental results, the error rate increases the rate of original and scrutinized dataset although it can be able to cover up intrusive noises. Therefore, Noise-added dataset is created with white noise to prove that any noise can hinder the system performance because it has full-spectrum coverage than the other noises: pink, brown and black. They cover only a section of the spectrum for different impact (pink and brown emphasize in lower frequencies and black noise emphasizes the sound of complete silence).

1.5 Organization of Research

This dissertation is comprised with seven chapters including literature review, related work and background theory of speaker recognition research, building speech

dataset for speaker identification, description of proposed system architecture, nature of speech data in Burmese dataset, feature extraction process, implementing GMM-UBM based and TDNN based acoustic models with data augmentation and scrutinizing methods, adding white noise, experimental results, conclusion and future work of research on Burmese speaker identification.

Chapter 1 describes the introduction, objectives, focus and contributions of the speaker recognition research work. Chapter 2 expresses the classification of speaker recognition: identification, verification and diarization, the speaker recognition criteria, the types of speech and speaker modes, applied areas, diversifications of speaker identification and literature reviews concerning with the dissertation. Background theories required for speaker identification process is described in Chapter 3. It includes data preprocessing suitable for Burmese language, feature extraction technique, the acoustic models about GMM-UBM, and TDNN, Probabilistic Linear Discriminant Analysis (PLDA), likelihood score computation and finally describes the performance metric. Chapter 4 explains how to collect and prepare the speech data from various sources and building scrutinized and white noise-added speech datasets for speaker identification. Moreover, the speaker information and statistics of Burmese speech datasets are also reported in this chapter.

Chapter 5 describes general architectures of speaker recognition system, and design and implementation of proposed system architecture for speaker identification. Chapter 6 describes the performance analysis for speaker identification. It includes building GMM-UBM and TDNN based acoustic speaker models, the experimental setup, performance results (with and without data augmentation) and discussion about the acoustic models. Finally, Chapter 7 presents the conclusion extracted from this research work with the advantages and limitations of research work and describes the future research lines to continue it.

CHAPTER 2

PRINCIPLES OF SPEAKER RECOGNITION SYSTEM

This chapter discusses the literature reviews and recent publications related to the Automatic Speaker Identification. Since early 1960's, researchers have been endeavoring to develop the computer system that can record, interpret and understand human beings' voice because speech signal carries the mixed types of information that express speaker specific characteristics or identity such as vocal tract, excitation source, behavior feature and language. For developing nations, the usage of speech may help as the language technologies for interacting with the computer and are implemented for e-government system.

2.1 Introduction to Automatic Speaker Recognition

Automatic Speaker recognition is one of the important research topics in the field of speech processing and is also known as voiceprint recognition. It can recognize the speaker by analyzing the speech signals and speaker characteristics elicited from their voices [73]. It focuses on the identity information of the speaker while speech recognition focuses on the text information corresponding to the voice. The individuals' sounds are not identical because of the different voice production organs and everyone has their own speaking style, vocabulary usage, pronounced pattern and so on. Speaker recognition systems attempt to verify and identify the speakers' individuality by their unique voice characteristics and control access to services such as banking transactions over a telephone network, telephone marketing, telephone shopping, voice dialing, security control access to confidential areas, voice mail and remote access to computers. The evolution of speaker recognition system from the late 1900s to the early 2000s is shown in Figure 2.1.

2.1.1 Classification of Speaker Recognition System

This section describes the three branches of speaker recognition. The speaker recognition system can be classified into three different types as speaker verification, speaker identification and speaker diarization in detail. They are considered to be the most natural and economical methods to avoid unauthorized access to the computer systems or physical locations. In general, the speaker recognition system has two

stages: training and testing. In the training phase, the respective speaker models are developed by using the training data. The model is labeled with the identity of speaker and is stored in the database containing the speaker models together with their corresponding identities. In the testing phase, the utterance of an open speaker is tested against the models existed in the database. This testing process relies on the problem, whether it is the speaker verification case, identification case, or diarization case.

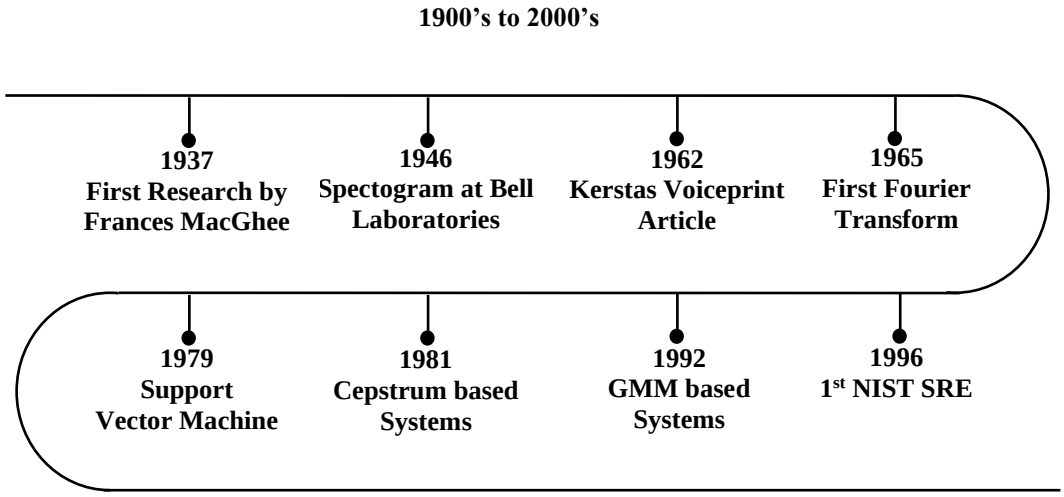


Figure 2.1 Evolutions of Speaker Recognition System

2.1.1.1 Speaker Verification

Speaker verification can be considered to be a special case of speaker classification in an open-set case. It makes a binary decision on whether the speaker he/she claims be. Verification means that the speaker affirms to be of a certain identity and the uttered voice is used to verify this claim. It is also known as speaker authentication.

Because one speaker's voice is matched to one template, it is called a one-to-one match. If the similarity score between the template and the voice sample exceeds a predefined decision threshold, the speaker is determined as accepted person, and otherwise this speaker is rejected. Setting the high threshold makes it difficult for accepting the different speakers by the system. It can cause rejecting the speaker falsely. Otherwise, valid users can be accepted consistently by a low threshold but accepting impostors can cause as the risk. Therefore, data showing distributions of target and impostor scores is essential to set the threshold at the acceptable level of false rejection and false acceptance. For this, one or more enrollment sessions are

required for obtaining the training utterances. Depending upon the application sensitivity, the matching score should be assessed with a predefined threshold. Moreover, the performance of verification task is not, at least in theory, affected by the population size since only two speakers are compared. Therefore, identification process takes longer than the verification process because of the processing time of matching. It is effortless for the system to match the testing speaker only with the claimed model from the database and not with the entire database and the testing speaker is either accepted or rejected.

2.1.1.2 Speaker Identification

The process of determining in which the acoustic speech signals is associated to its respective speaker is known as speaker identification. The input speaker's utterance is analyzed, the features are extracted, a speaker model is developed by this input utterance and it is assessed against all the existing speaker models in the database. This takes a protracted process because it relies on the number of the speaker models existed in the database. On the other hand, it classifies the test speaker into one of the pre-modeled classes (close-set case) and identifies the test speaker as a new speaker (open-set case). In open set case, the test speaker is defined as new or unknown speaker if it does not match any of the speaker models in the database. It is a one-to-N speaker match where the voice is compared against N templates. It is one of the challenging topics in signal processing and the validation task of claimed identity by machine. Of the verification and identification tasks, verification task is not difficult to implement. Identification task is generally considered more difficult. The probability of an incorrect decision increases when the number of registered speakers increases. This is intuitive. In speaker identification, it can be sub-divided into text dependent and text independent cases, based on whether or not the speech used is known for each speaker [13].

2.1.1.3 Speaker Diarization

Speaker diarization is also an essential part of speaker recognition system. It is the process of splitting up an audio recording stream containing a number of speakers into homogenous segments. These segments represent the unique characteristics associated with each individual speaker. It is a task of answering the question "who

spoken when?" It is identified which speakers talk when and also referred to as the step-by-step process that is the system discriminates the speaker segmentation of the speech signal, speaker clustering of these developed segments into homogenous groups with respect to the changes in speaker and then followed by some hypothesis result. All these steps are performed within the same stream. The system has no prior knowledge about the speakers' identities and how many speakers are participating in the input stream. It has the applications of many fields such as video captioning, content structuring, audio information retrieval, understanding the content of any conversations, speaker indexing and segmentation (locating the boundaries by finding acoustic changes in the signal), etc. [75].

2.2 Speaker Recognition Criteria

There are many criteria in speaker recognition from the aspects of text usage (dependent or independent), included speakers (open or close), type of word usage (isolated or connected) and type of speech uttered (continuous or spontaneous). Text- and speaker-independent (open set case) are more convenient and applicable in practice because it can freely test the system without restricting the text usage and speech uttered.

2.2.1 Classification on Usage of Text

This section presents the text criteria which can be either text independent or dependent. The training and test data use the same transcript in text dependent system. However, the system without depending on text does not need to use the same text. Although text independent system is more accurate in system performance, text independent system is more convenient in practice because the speakers can freely speak to the system without any constraints and limits on speech contents.

2.2.1.1 Text Dependent

Text dependent systems need to use absolutely the identical utterance to decide who they are. Both training and testing datasets use the same transcript of text. The text used during the testing phase contains as a subset of the whole text during the enrollment phase. Therefore, the test speaker has prior knowledge of the system and text dependent systems are more accurate than text independent system.

In other words, the text spoken by a person is known and the speaker must say a fixed or prompted sentence. In security-oriented applications, where user input is strongly controlled with respect to access to personal organization, text dependent recognition of the speakers is applied. The advantage of this type is that the system has an early knowledge of spoken text, making it more efficient.

2.2.1.2 Text Independent

Speaker recognition without depending on text is applied for identifying any type of informal speech and colloquial of user. It has no constraints on the speech contents and limits on the spoken words that are uttered. Training and testing data are entirely unconstrained. It does not know the previous information of the text spoken by person. Applications with a lack of user input controls typically use this type.

Compared to text dependent speaker recognition, it is usable and more convenient in real world applied areas because the speaker does not have any prior knowledge about the contents of the training phase and can speak freely to the system. To get better accuracy, text independent speaker recognition system needs training and testing data more.

2.2.2 Classification on the Types of Speakers Mode

This section presents the criteria of speaker aspects can be either speaker dependent or speaker independent or speaker adaptive.

2.2.2.1 Speaker Dependent

One speaker at a time can be identified using a speaker dependent system. It is trained on all of the data per speaker and generally easier to implement, cheaper to purchase and more accurate in recognizing. This type of system is more applicable for close-set speaker identification systems.

2.2.2.2 Speaker Independent

This type of system is built to recognize the voice of anyone, no matter who is speaking. This kind is very hard to develop and most expensive. It has a lower accuracy rate than speaker-dependent applications because the incoming speaker is or

isn’t known by the system, but more flexible in practice. It is more suitable for automatic speech recognition systems and interactive voice response systems.

2.2.2.3 Speaker Adaptive

The speaker adaptive systems utilize the speaker dependent data to adapt the speaker independent system. They also adapt to the best appropriate speaker to recognize the speech and increase the accuracy rate by adaption [22]. Speaker adaption technique is mostly used in automatic speech recognition.

2.2.3 Open and Close-set Speaker Recognition

Speaker recognition is further categorized into open-set and close-set tasks. The recognition task is a close-set problem if the target speaker is assumed to be one of the registered speakers. The system makes a forced decision simply by choosing the best matching speaker from the speaker database – no matter how poor this speaker matches. In other words, close set problem has only a specified (fixed) number speaker of registered to the system.

The task is called an open-set problem if there is a possibility case that the target speaker is none of the registered speakers. It is much more challenging in general. The system must have a predefined tolerance level so that the similarity degree between the unknown speaker and the best matching speaker is within this tolerance. The verification task can be seen as a special case of the open-set identification task, with only one speaker in the database.

Classification of speaker recognition system is described in Figure 2.2. Text independent open set speaker identification in Burmese is implemented as research work because text independent system is applicable and flexible in real world applied areas.

2.2.4 Classification on the Types of Speech

This section presents the criteria of speech uttered by the speakers in the system. The types of speech used in this work are continuous speech (broadcasting news), and spontaneous speech (radio talks, conversational talks like interview, delivered speeches) comprising with the connected words. Isolated words have

shorter length than others and are mostly used in automatic speech recognition and speaker verification system.

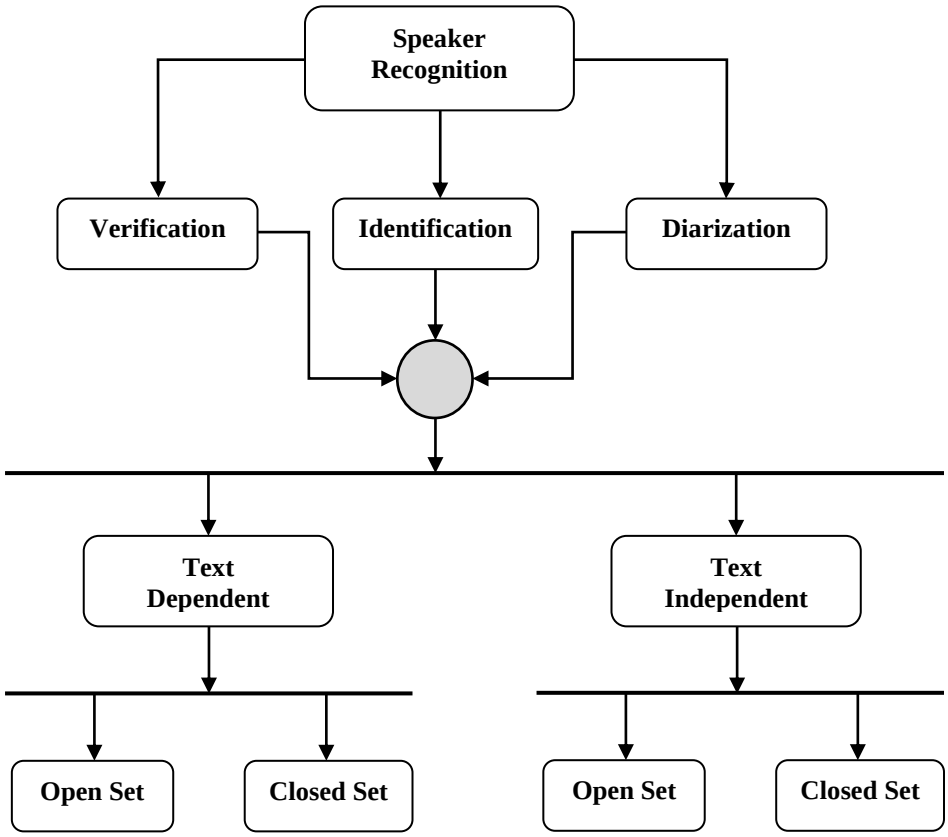


Figure 2.2 Classification of Speaker Recognition System

2.2.4.1 Continuous Speech

The users speak almost naturally in continuous speech systems without silent pauses between words and the contents. The recognizers are difficult to develop on continuous speech because they need special methodologies to decide the boundaries of the utterance and allow the users to talk the system without stops and pauses. It can recognize more utterances than a command-and-control system.

2.2.4.2 Spontaneous Speech

It can be thought of as a casual way of speaking style in basic, and a speech that is not rehearsed and automatic natural sounding. It is opposed to read-aloud speech and generated in real time. Examples of spontaneous speech are interviewing speech, delivered speech and conversational talk.

2.2.4.3 Isolated Word

Isolated word recognizers set the condition of each utterance having with little noise or clean on both sides of sample window by recognizing single word or utterance at a time. These types of speech have the states of "Listen/ Not-Listen" because the speakers have to pause between utterances.

2.2.4.4 Connected Word

The connected word requires being the separate utterances with the minimum pause or stop between utterances to be smooth. These are similar to the isolated word having the states of "Listen/ Not-Listen".

2.3 Application Areas of Speaker Recognition

Speaker recognition has many application areas, namely: law enforcement, access control: physical facilities, computer networks and websites, speech data management, online transaction authentication: telephone banking and remote credit card purchases, surveillance, forensic speaker recognition, security, multi-speaker tracking, multimedia and personalized user interfaces. The following are the example applied areas from some of them.

- Access control: Confidential computer databases as well as secure physical locations can be entered through sound. Access can also be allowed to restricted and private websites. One of real world example is door locks. In this, authentication is performed by speaking freely while the door button pressed.
- Law enforcement: Additional information for forensic analysis can be provided by using speaker recognition systems. At prison, convict roll-call monitoring can be implemented automatically.
- Online transactions: An access phrase to buy an item over the phone or to pass bank information, person's speech signal can be used as an additional security layer.
- Management in Speech Data: Audio mining applications, voicemail services, and live or recorded meetings' annotation can use to label the text spoken by the speakers automatically.

- Telephone Banking: To check whether an authorized person is attempting to enter the accounts, personally and private information, voice control access may be used when entering a bank account. In order to respond and adapt to users, intelligent machines may be installed.
- Personalization and Multimedia: Singer name and track information are labeled on sound tracks and music automatically. To send e-mail messages over a phone, E-mail sending service application allows the callers as part of personalization. As an attachment, voice message can be recorded and sent to the E-mail. A special recognizer is used for spelling e-mail addresses, subject and sender of the message. To improve the recognition performance of the spelling recognizer by loading speaker dependent data, identification system is used. In this, the sender is identified by a combination of ASR scores and verification scores.

2.4 Diversifications of Speaker Identification

Many diversifications are still encountering in every speaker identification research like duration, environmental condition, voice tones (quiet, normal, shouted), speed (slow, normal, fast), noise robustness, accent, speaker characteristics, recorded conditions, speaking styles, speaker variability, sex, age, and so on.

Among them, the duration of utterances is the first one in variations. The longer the utterances, the more recognize the specific facts of the corresponding speaker. Noise is the second variation because various kinds of sounds that causes disturbance can lead to decline the performance of identification. The accent of speaker specific facts varying among speakers comes as the third variation. It's easy to tell who the speakers are if they speak normally or the way they were trained to speak. If the speaker is speaking in a common local dialect or language that corresponds with the training data, the speech is easier to understand. The fourth variation is the recorded conditions. The voice that is recorded in quiet conditions can raise the system performance. The other variation is the speaker variability such as the health of the speaker, vocal effort, emotional condition, stress, phonation style, and disguise.

Moreover, there are still remaining the variations such as the impacts on the shape of the spectral envelope produced in speaker identification.

2.5 Literature Reviews

This section points out about the literature and incremental efforts of speaker recognition system. In 1952, Audrey at Bell Laboratories invented understanding number as voice recognition first. The attempt as the first work for automatic speaker recognition was made in the early 1960s after one decade later than that for automatic speech recognition. The first research corresponding to speaker recognition was approached by Pruzansky at Bell Labs in 1960s, where he employed filter banks and correlated the two digital spectrograms for measuring the similarity [27]. Prunzansky and Mathews [40] enhanced upon these techniques, and linear discriminators are used by Li et al. for further development [41]. In 1970's, the first automatic speaker verification system was developed by Texas Instruments by replacing formant analysis on behalf of filter banks. Furui suggested using the combination of cepstral coefficients together with their first and second polynomial coefficients as frame-based features to make the sound clearer when it is changed by the telephone system. In [42], an online system was tested and used for six months with 120 users making many calls. Later, not only for speech recognition, but also for speaker recognition, the cepstrum-based features become standard. In 1980's, a major breakthrough was the development of the hidden Markov model which used statistics. Robustness became a central theme on increasing research in the 1990s. Text-prompted speaker recognition was proposed by Matsui et al. [43] in which every time used by the system; the key sentences are absolutely changed. A mixture of a syllable based HMM and a GMM system adapted by MAP system is evaluated with 35 speakers of NTT dataset [28]. In this, MFCCs were used as speech features and the recognition accuracy was found to be 99 % in 1995. HMM techniques and Vector quantization (VQ) were investigated by T. Matsui and S. Furui to make more robust at speaker recognition in 1996 [29]. In 1999, speaker identification without depending on text was studied based on a segmental approach. Final decisions and outlier rejection was based on a confidence measure [30]. In this work, Mel-frequency Cepstral Coefficients were used as acoustic features.

To raise the confidence level of speaker recognition systems and to make the systems more robust, research became focusing on adding higher level information such as pitch and energy to speaker recognition systems in the 2000's. In 2001, the idealistic features of speech such as word unigrams (monograms) and bigrams as language models from manually transcribed conversations were applied by G. R. Doddington [44] to characterize a certain speaker in a traditional target/background likelihood ratio and score. Evaluation was carried out on the task of NIST extended data, consisting of the long duration speech conversation recorded on telephone from 400 speakers. A missing rate of 40 % was observed at every FAR of 1% [31]. D. A. Reynolds has been done robust text independent speaker identification by using GMM speaker models since 1995. The focus of this work aimed for practical applied works like voice mail labeling and retrieval [32]. In 2003, D. A. Reynolds et al. were also investigated successfully in text independent speaker recognition with the use of Gaussian Mixture Models (GMM) [33]. They used high-level information which were fused and modeled with the use of multi-layer perceptron (MLP) to combine n-grams, Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs) [34]. The extended dataset of NIST was employed for evaluation and TAR of 98% was observed at every 0.2% FAR. In 2006, the NIST's multilingual dataset of 310 speakers was also used to identify multilingual speakers. Using GMMs, N grams, and SVMs, several speaker-related features were modeled from short-term acoustics, prosodic behavior, pitch, duration, phoneme, and phone usage [35]. Many modeling systems using a multi-layer perceptron (MLP) were experimented together in this work. It has been reported that the recognition rate is 60% and the FAR is 0.2%. P. D. Bricker et al. experimented the speaker recognition on independent of text using averaged auto-correlations in 2005 [36]. J. M. Naik et al. approached the research using HMM techniques, rather than template matching speaker recognition depending on the text [37]. In 2006, the speaker models were implemented on the features of Mel-frequency Cepstral Coefficients (MFCCs) together with the adoption of speaker adaptive modeling and phonetically structured GMMs. This method was measured on Mercury dataset including 38 speakers on the quality of speech recorded with telephone and YOHO dataset comprising of clean speech data from 138 speakers. The error rates were commemorated to datasets of 18.3% on Mercury and 0.25% on YOHO [38]. Moreover, GMM-UBM system on MLP fusion and speaker adaptive ASR system were used to model the acoustic features. In this work, MFCCs and its

first order derivatives were used as acoustic features. The rate of 7.3% has reported as a miss rate when evaluated on the Orion and Mercury data involving 44 speakers in total [39]. In most speaker recognition systems, universal background models (UBM) are constructed on Gaussian mixture models (GMM) from training a large amount of data using expectation maximization (EM) algorithm.

Nowadays, many researchers become apply the neural architectures in speaker recognition not only in front-end analysis but also in backend processing to promote the recognition accuracy. Among them, time delay neural networks (TDNNs) become one of the popular neural architectures due to its ability of time shifting and input context modeling. It is a kind of multilayered neural network which involves the properties of its classifying patterns with shift-invariance and modeling context at each layer of the network. The first architecture of Time Delay Neural Network, TDNN is developed by Alex Waibel since 1987 to apply in phoneme classification for ASR in which the automatic determination of precise segments of feature boundaries was difficult [7]. One of the methods he was investigating is stage learning to speed up the training time of the networks. It was found that the first subsets are small and fast to learn, and the size of each successive subset increases rapidly until the entire training dataset is used. It is easy to learn the smaller set of data. In order to minimize errors and improve generalization, the larger sets fine tune the networks. From 1989, the success of TDNN is emphasized by many speech researchers. TDNN has the two properties of dynamic structure of speech: the temporal structure and relationships between acoustic events. The advantages of TDNN are reducing the number of weights; requiring fewer examples in the training set, faster learning and executing compared to the fully connected multi-layer perceptron (MLP). According to the benefits using TDNN, it was evolved and applied to speaker and speech recognition until the success reaches today. From the success in [7], TDNN architecture applied with many modifications in network layers and changing time steps in the input contexts. In 2015, the data derived from using the two types of data augmentation methods: reverberation (RVB) and speed perturbation (SP) were used for building the acoustic models on TDNN which are robust to training data distortions. Three different distorting databases (Aachen impulse response database, REVERB challenge database, RWCP-SSD sound scene database) were used to get the multi-condition training data. Three different copies of individual original speech were formed by randomly sampling. The results were compared with three different

systems constructed using three types of input contexts frames namely [t - 22, t + 12], [t - 16, t + 12], and [t - 13, t + 9] in various training data creating of with only reverberation and both reverberation and perturbation extracting i-vectors or without extracting i-vector. According to the results, speed perturbation was not effective in this work. Moreover, although acoustic models constructing with i-vectors leads to better results, the acoustic model constructing with the use of features directly increase word error rate. In extracting i-vectors, extracting from reliable speech segments can achieve better performance than extracting from the speech segments comprising with speech and non-speech segments [45]. The effectiveness of learning wider temporal dependencies of TDNN architecture on both small and large dataset was proposed in [18]. It was used to model the temporal dependencies in long term from short-term speech features like MFCCs. To emulate perturbation on speaking rate and vocal tract length, applying the method of speed perturbation on training data was done. This was provided the progress across several LVCSR tasks with 4.3% relative improvements. The speed perturbations of 0.9, 1.0 and 1.1 were applied to obtain the three copies of the corresponding perturbed training data. The volume perturbation compared to speed perturbation reduced the word error rate (WER) with 1.5% relative improvement across test sets. The results showed that DNNs are not as efficient at processing larger temporal contexts as TDNNs. The temporal input context frames of [t - 13, t + 9] was observed the excellent temporal context. Compared with baseline configuration, an average relative improvement of 6% was shown across six different LVCSR tasks. David Snyder compared the system performance of acoustic models among GMM-UBM, supervised GMM-UBM and multi-splice TDNN for speaker recognition [15]. The system performance is evaluated on the condition five extended task of SRE10. Six layers multi-splice TDNN with left input contexts of 13 and right input contexts of 9 was used as in [18]. Several thousand of mixture components on full covariance GMM was employed to train the background model on 5 iterations of EM. Full covariance supervised GMM (sup-GMM) was used in a lightweight model for creating the speaker recognition features and using DNN posteriors to implement the GMM for the purpose of modeling the phonetic content. The only difference between unsupervised and supervised GMMs is in the UBM training process. The system performance was evaluated with three points: Equal Error Rate (EER), minDCFs (10^{-2} and 10^{-3}) on gender dependent and gender independent models. According to the results, TDNN

based systems achieved the best outcomes resulting 1.09 % and 1.2 % EERs on gender independent and gender dependent respectively.

2.6 Summary

This chapter discusses the principles of speaker recognition system in detail. It describes what the speaker recognition system is and the difference among speaker verification, speaker identification and speaker diarization contained in the speaker recognition system. It also presents the criteria of speaker recognition depending on the text independent or dependent, types of speech and speaker dependencies. Moreover, some applied areas of speaker recognition are described and the variations encountered in speaker identification research are also explained. In the end, literature reviews and evolution history corresponding to speaker identification are expressed in brief.

CHAPTER 3

SPEAKER IDENTIFICATION METHODOLOGIES

This chapter describes the theoretical backgrounds and methodologies to speaker identification system in detail. It describes about the signals of speech as well as the mechanism for its development and representation. It also describes the speaker identification process consisting of preprocessing the speech data to achieve the high recognizing rate, eliminating the non-speech data frame, extracting features from the well-prepared input speech data, taking the highly correlations of feature vector, computing the log likelihood ratio for scoring and appraising the system performance in terms of equal error rate and detecting accuracy.

3.1 Speech Signals

A speech signal consists of a mixture of voices that are created by exhaling air from the lungs and stimulating the vocal tract. The irregular acoustic tube is the glottis which its length is determined by the distance between the lips and the vocal folds, and its cross-section is determined by the position of the tongue, velum, jaw, and lips. When the area of cross sectional and the vocal tract's length are changed, dissimilar sounds are produced. Because of divergences in shape and length of vocal tract, the speech signal seems different although different speakers who produce the same phonemes convey the same information. The resonant frequencies, which are also known as formants contains in the spectrum of the vocal tract. One formant is generally presented every 1 kHz on average meaning that 3 to 4 formants may be contained in the band of speech signal limited to 4 kHz. The nasal tract is another important part of speech production starting with the velum at the end of the throat and ending with the nostril. When the velum is lowered, the nasal tract is acoustically connected to the vocal tract to produce the nasal sounds of speech. Air flows through the vocal folds and vocal tract. The sound of excitation is produced because the vocal tract forms the spectrum of the speech signal.

The speech signal is separated into two categories based on how the vocal cords work: unvoiced speech and voiced speech. The generation of unvoiced and voiced speech is separated by silence region in the producing process of speech

because the speech is not complete with the absent of silence region between unvoiced and voiced speech. There is no speech sound output in the silence section because there is no excitation supplying to the vocal tract. It is not important but it is very essential for intelligible speech.

Unvoiced speech occurs when the vocal cords are kept high and air continues to flow at open speed through the vocal folds into the vocal tract. It is producing an aperiodic noise which is similar to the natural signals. This is the main difference between voiced and unvoiced speech. Compared to voiced speech, the formants at the lower frequencies are generally lower in magnitude than those even at the higher frequencies in unvoiced speech.

However, when the vocal cords vibrate at a fundamental frequency of the voice and take the part of an active role, the voiced speech is generated. Vocal cords receive air in small puffs that cause the vocal folds to open and close on a quasi-periodic basis for voiced speech. This gives rise to a glottal wave, which is formed by energy at fundamental frequency and its harmonics. To produce the voiced speech, this wave is traveled through the vocal tract. The formants are the resonant frequencies that the vocal tract responds to. In order to define voice speech, higher frequencies formants which are usually lower in magnitude than lower frequencies formants shall be considered. As the frequency decreases, the level of formants increases; however, there are certain exceptions to this rule. The magnitude level of formants continues to increase as the frequency decreases. However, there are certain exceptions to this rule.

According to age and gender, the fundamental frequency existed in voiced speech is ranging. In the range of 200-400 Hz, the fundamental frequency of children is present. Male speakers' frequency rate fluctuates between the ranges of 50-200 Hz while the fundamental frequency of females is changing from 150 Hz to 300 Hz. This fundamental frequency determines the pitch of the speakers. The ones who have higher pitch possess the higher fundamental frequency. Due to higher frequency rate, children's voices are the loudest, followed by women and then men.

3.2 Data Preprocessing

Preparation of the data for a proper implementation of further processing steps such as features extraction and speaker modeling, is an essential part of front-end analysis. The quality of data preprocessing in front-end analysis performs the important role in speaker identification systems. From high to low quality speech signal subspaces, the front-end analysis is transformed while preserving discriminative characters and features of speakers [8]. Speech data collection is the very first step in any statistical based speaker identification tasks especially for under resourced language. Burmese speaker identification encounters some problems due to the lack of proper data. For text independent speaker identification, the speech dataset for low-resourced tonal language like Burmese is not existed in publicity although English that is resource rich languages are easily available. Therefore, as the first contribution of this work, the speech dataset needs to build first systematically for Burmese language. The speech data were obtained from two main sources: online sources[1] (Web-based) and collected ourselves in this work.

The videos are formed into different formats types (.mp4,.wma,.mp3) in a variety of frequency. It is needed to convert the wave (.wav) file format uniformly with the help of the command *'ffmpeg*' for obtaining Web data. These are formatted as 16-bits mono channel PCM in frequency rate of 16 kHz after converting to wave files. The features extracted from this frequency rate enhances and affects for further processing because this rate is the most suited one for Burmese's spoken tone. Then, the whole wave file is split to multiple speech segments with ***Audacity***[2], *open source and cross-platform recorder and audio multi-track editor software* and then is cut out the silent part simultaneously. The non-speech (silence) frames are removed as a task of detecting the voice activity. The data collected by this way have clear and accurate sounds because the speakers are well experienced and professional. Both global and local news about sport, health, politics, speech, crime, education, weather and business news, etc., contain in Web-based data.

[1]Democratic Voice of Burma (DVB), Voice of America (VOA) Burmese, Eleven Broadcasting media, British Broadcasting Corporation (BBC) Burmese news, Radio Free Asia (RFA), Mizzima News Myanmar, One News Myanmar Channel, Irrawaddy Burmese News, 7days TV, Myanmar Radio and Television (MRTV)

[2] https://www.audacityteam.org/

For recorded data, the text corpus of the daily conversational dialogue is prepared first and the transcripts in the text corpus is recorded with microphone and telephone which are already setting to 16-bits mono PCM in 16 kHz of frequency rate. The dialogues for recording were obtained from U-STAR Universal Speech Translation Advanced Research which is ASEAN language speech translation. These data were recorded by 25 Lab members and 24 internship students from three academic years. In a quiet room at University of Computer Studies, Yangon, Myanmar, these data are recorded. It does not suffer from external disturbance such as room's echo and environmental noise. The recordings are performed with the device naming Tascam DR-100MKIII[3]. The duration of every daily conversational dialogue is more diminished than the duration of the news, talks, and delivered speech. These are the words of conversational dialogue in restaurants, parks, hotels and traveling.

3.2.1 Data Augmentation Techniques

The system development has reached to a certain condition but robustness and noise tolerant systems are remained problems which cause the system inconvenient to use. There are many researches around the world that are currently being conducted to the development of speaker identification systems in robustness. Like Burmese language, tonal languages such as Vietnamese, Thai, Mandarin, etc., augmented the tone corresponding features in building acoustic speaker models in order to increase the recognizing rate and robustness of their identification system. Moreover, unlike them, Burmese speaker identification system also shows the development by getting the data from scratch because Burmese language is the low-resourced language. Therefore, this research aims to develop automatic speaker identification with the use of state-of-the-art features and technologies applicable for Burmese speech. The speech quality matters for more precise recognition, in addition to the efficient volume of speech that is required for each speech processing task.

Data augmentation can be regarded as a method to generate additional training data for gathering the substantial volume of data in a short time. This will increase the volume and diversity of the training data. It is a technique to artificially create new and different data from existing data by contaminating the artificial value to existing data with adding the recorded data information derived from surroundings.

[3] https://tascam.com/us/product/dr-100mkiii/top

Augmented with noise and reverberation as a low-cost method multiply the diversity and quantity of existing training data without actually collecting new data and improve the system robustness. It can be beneficial in training powerful. It can also find out the more speaker specific information due to the increasing length of utterances (durability).

As part of augmenting the data, additive noise and reverberation are employed to the original training data. And also, convolving simulated room impulse responses (RIRs) with audio is applied to add the echo to the speech data artificially. For additive noise, MUSAN[4]: Music, Speech and Noise dataset consisting of over 900 noises, 60 hours of speech derived from twelve languages and 42 hours of music collected from various genres is used for augmenting the existing training data [21].

- **babble**: Three to seven speakers are randomly picked from MUSAN speech, summed together, then added to the original signal (13-20 dB SNR).
- **music**: A single music file is randomly selected from MUSAN, trimmed or repeated as necessary to match duration, and added to the original signal (5-15 dB SNR).
- **noise**: MUSAN noises are added at one second intervals through the recording (0-15 dB SNR).
- **reverb**: The training recording is artificially reverberated via convolution with simulated RIRs.

These augmented data are randomly selected for combining to the original "clean" training data. By doing this, the data size becomes doubling the size of the original dataset. It is a strategy adapting to raise the abundance of training data, enhance the robustness of the models and avoid over-fitting [14]. Augmenting makes increasing the volume and diversity of data and the derived results are acceptable. Therefore, augmenting the training data is applied as the second contribution of this work for the aim of enhancing the system performance.

3.2.2 Data Scrutiny

This section presents the third contributions of this work. Nowadays, researchers are keeping analyzing and investigating the performance of their

[4] https://www.openslr.org/17/

researches not only from the aspect of data quality but also from theoretical surveys. Although the facts that the theoretical perspectives enhance the performance progress of system, the next processes following data collection are convenient and the system is more robust only if the high-quality data is used. The researchers become more and more careful in preparing data because it is the most important and essential part in every research development. Moreover, the data done as possible as exact can give the more compatible results and enhances the system performance. Therefore, the speech dataset is well-preprocessed and scrutinized as the main contributions of this research work according to these facts.

Many audio analyses in tempo, speed and volume, etc. are investigated in [14, 45]. Among them, the two scrutinizing methods are contributed in preparing Burmese speech dataset to implement the speaker identification system more robust. Firstly, since changing the speech intensity of speech can change the structure of the sound spectrum, the Signal to Noise Ratios (SNRs) levels are adjusted to examine the intensity of speech segment. Different intensity levels are tested to find out whether changes in voice improve. The performance of data on these SNRs levels (-10 dB, -5 dB, 5 dB, 10 dB) is analyzed because the form of the loudness of sound collected from different sources varies. These data comprised with different dBs were implemented to take out of which dB scale is suitable for the timbre of Burmese. This is implemented by setting the same SNRs uniformly to all of the speech segments. According to the results, setting the intensity level up to 10 dB gave the assured results than the performance of original data.

The further method is the detailed inspection in tempo factors. This is done by making the tempo up and down the speech segments. Although the target speaker may recognize slightly slower rhythm, a speaking rhythm that is too fast may not accurately capture the speech. However, as a consequence, reduction in speech rate may cause the duration of the initial utterance to increase. When the speed factors increase and decrease in tempo with 0.2 times (20%) apply on existing original speech dataset, decreasing the tempo factor on speech yields more reasonable results than increasing tempo coefficient and rhythm are normal. The cause of performing the tempo factor down than the tempo factor up is that speaking more slowly can catch and identify well what they are talking.

There are two variations of pace on spoken speech namely tempo and speed. In this work, tempo is altered rather than speed because tempo factor investigation had no effect on speech pitch. But, altering the speed rate leads to not only change the spectral shape of speech but also influence on both tempo and pitch. This can lead to loss of precise information about the speaker contained in the speech clips. According to pointed out by the experiments, it can be seen that speaking slower can better recognize the speaker's voice and better understand what we are saying due to the clarity of vocabulary. Moreover, it may also make the improvement in automatic speech recognition.

The examined dataset is then created by allowing the SNR level to rise to 10 dB and the tempo factor to slow to 0.2 times (20%), in order to approach the nature and efficacy of intensity levels and tempo factor in voice signals. Acoustic speaker models are then built using the features extracted from the original, scrutinized and white noise-added datasets to substantiate that the scrutinized dataset decreases the error rate than the original and white noise-added datasets.

3.2.3 Impacts of Adding White Noise

White noise refers to the sound of all audible frequencies at the same amplitude or volume. It has full spectrum coverage as opposed to the other noise colors: pink brown and black noises covering only a portion of the spectrum for different levels of impact. While white noise that promotes relaxation is a pleasant sound for most people, adding it to human speech can lead to the disturbance. The aims of adding white noise to the clean data is to simulate the corruption and to obtain the multi condition noisy training data [77]. Although one advantage of containing white noise in speech is that it can be able to cover up the intrusive noises, the accuracy rate still degrades than the original clean data and scrutinized data. This is because any kind of noise can hinder in recognition process. To cover the noises of different spectral shapes and bandwidths, the experiments were done in this work with adding white noise to the clean data. According to the results, white noise leads to degrading the system performance on acoustic speaker model.

3.3 Feature Extraction

The absolute speech is derived through a Front-end analysis module that separates speech components from non-speech parts. From these absolute speeches, the acoustic features required for speaker modeling are extracted. Feature extraction is a fundamental preprocessing step to pattern recognition and machine learning problem. It is a part of dimensionality reduction process in which the raw data are diminished and split to more manageable states.

In other words, the main goal is to compute the feature vectors sequence representing a compact description of the input signal but effective depiction that is more stable and discriminative than that the original signal. It is a unique kind of dimensional reduction that is used to get rid of data that is too big for an algorithm to handle. The given input data is converted into a feature set, which produces the necessary data to carry out the intended task using the reduced set without requiring the full-size data.

Feature extraction plays the important role not only in speech synthesis, analysis, recognition, coding and enhancement but also in speaker recognition, voice modification and language identification. It is the process of keeping the useful information from the speech signal while unnecessary information is discarded. In this, some useful information may lose in removing the unnecessary information from the speech signal. Moreover, low level (short-time) features are more powerful than the high-level features. Therefore, the acoustic features can be easily extracted. However, the extraction process is more difficult than the low-level features, although the high-level features contain more information related to the speakers [4].

3.3.1 Categories of Audio Features

Transforming audio streams into acoustic features can be efficiently applied to extract unique characteristics of individuals. The proposed features used for speaker identification are separated into long time (high-level) features, medium time features and short time (low level) features depending on the duration [74]. It is advantageous to divide the basic time period for features, so that they can be chosen according to the decision timeframe. Short time (low level) features like Mel Frequency Cepstral Coefficients (MFCCs) and Perceptual Linear Prediction (PLP) are easy to extract

speaker specific information but long time (high-level) features like pitch are complicated to compute. Some of the audio features are shortly expressed below.

3.3.1.1 Short-Time Features

Short time features are considered to be the speech properties obtained from frames of approximately 30 m long, and have been referred to as low level characteristics. In extracting the instant frequency or timbre of the signal, such features are effective. Moreover, they are very easy to excerpt speaker specific information over the frames. For speaker identification, MFCCs have been regarded as the most effective feature in short time level, have surpassed than other features at a similar time and have given the accurate recognizing result [74].

3.3.1.2 Medium-Time Features

The medium-time features are defined as features which are taken once every 740 ms and can be retrieved from the signals of a longer frame length, or short time features that have continuous frames. They are also efficacious for retrieving modulation components of the signals. MFCCs, and mean and variance of Filter Bank coefficients are short-time features. To obtain medium-time features, another technique used for feature consolidation of MFCCs is autoregressive model [74].

Low short time energy ratio (LSTER) and high zero crossing rate ratio (HZCRR) are other instances of medium-time features extracted from short time energy and zero crossing rate (ZCR) of the signal respectively. HZCRR shall be defined as the number of frames in which a zero-crossing frequency is at least 1.5 times higher than an average ZCR. Although they offer some benefits in ease of implementation and reduced computational costs, the ratio of the numbers of frames with temporal energy less than half to these two features can be significantly impaired due to noise identified as LSTER.

3.3.1.3 Long-Time Features

In frames whose length changes from 4.81 seconds to 9.62 seconds, the features obtained are called long-time features and it can capture the phonetic, prosodic and lexical information. It also refers to the features that are extracted over

regions longer than a frame. Although long time features can provide useful metadata about a speaker for discriminating, they are very hard to compute.

3.3.2 Mel Frequency Cepstral Coefficients (MFCCs)

Speech features isolate a speaker from other speakers with the aim of exploring robust and discriminative phonetic features that improve detecting rate in acoustic data. Mel-frequency Cepstral Coefficient (MFCC) is the most useful feature extraction technique in every speech processing system. In 1980, it is a nonlinear mapping of the audible frequency range introduced in speech processing domains [24]. In speech classification problems such as speech and speaker recognitions, it has become one of the most widely used short-time features because they produce an accurate and compact representation of the speech magnitude spectrum.

Short-term speech representation is a commonly utilized feature in speech processing systems, drawing inspiration from the human auditory system. It is also a type of feature extraction method that extracts short term speech features from long term temporal dependencies. Low-level spectrum (short term) features are more potent and simpler to extract than high-level features. Although high-level features contain more information about speakers than low-level features, feature extraction from high-level features is a more difficult and time-consuming operation. Figure 3.1 shows the extracting steps of MFCC features.

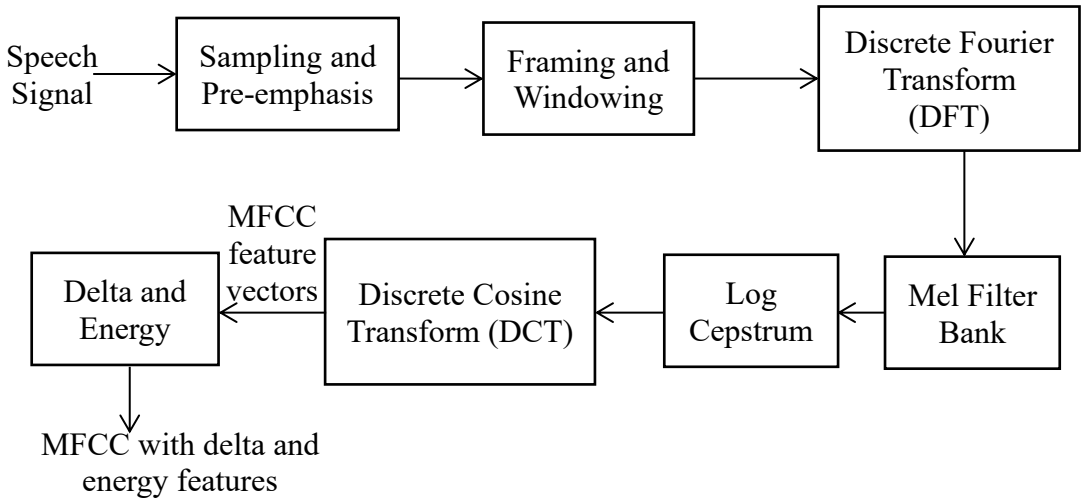


Figure 3.1 Mel Frequency Cepstral Coefficients (MFCCs) Extraction Steps

Therefore, for the next processing step, the quality of the speech features should be improved. The 13rd feature combining 12-dimensional Mel-frequency Cepstral Coefficients (MFCCs) features and 1 energy feature is extracted by using the Hamming window. It extracts the features every 10 ms from the frame size of 25 ms long for short-time Fourier transform (STFT). Double delta features are added to have 39 dimensional MFCC features for more recognizing the speakers' identity and improving the system performance.

3.3.2.1 Sampling

The process of measuring the instant values of continuous time signal into a discrete form is defined as sampling. It is the first step of converting the analog representation into digital signal in processing speech. Continuous analog signals are digitized by converting to discrete time, discrete valued signals. Converting analog to digital signal involves two steps: sampling and quantization.

A sample is analyzed by measuring its amplitude at a particular time; the total number of samples taken per second is defined as the sampling rate and is generally used between 8 kHz and 20 kHz for speech processing systems. 16 kHz range is suitable for speaker identification system. A wave needs at least two samples in each cycle in order to be measured precisely. The first one gauges the wave's positive component, while the second gauges its negative component. If there are fewer than two samples in a wave, the frequency will be entirely missed; however, if there are more than two samples in a cycle, the accuracy of the amplitude is increased. Consequently, a frequency wave with a frequency half of the sample rate is the highest frequency wave that can be detected. The Nyquist frequency is the highest frequency for a given sampling rate. Most information in human speech exists in frequency below the range of 10 kHz; therefore, the sampling rate existing in the range of 20 kHz would be necessary for complete accuracy. Wideband sampling rate which has 16 kHz is mostly used in every speech related system.

Quantization determines what scale is used to represent the signal intensity. It is usually stored as integers, either 8 bits ranging the values from -128 to 127 or 16-bit ranging the values from -32798 to 32767. Quantization is the process of expressing real-valued numbers as integers. This is because any values that are closer together

than this quantum size are represented similarly, and there is a minimum quantum size (the granularity). Generally, it appears that 11 bits numbers capture sufficient information, although by using a log scale, 16 bits per sample provide 2^{16} = 65536 quantization levels. In 16 kHz of sampling frequency is suitable for 16 bits pulse code modulation (representing the speech signal by binary-coded quantized samples) for enough processing power, a higher bit resolution for the sampled values is preferable. Each sample in the digitized quantized waveform is referred as x [n], where n acts an index over time. A digitized, quantized representation of the waveform is gained and it is ready to extract MFCC features [23].

3.3.2.2 Pre-emphasis

Pre-emphasis refers to filtering that emphasizes the higher frequencies. That is, the speech signal is lesser in magnitude during high frequencies. In other words, it is performed for flattening the magnitude spectrum and balancing the low and high frequency components. The purpose of pre-emphasis is to balance the spectrum of voiced sounds that have a steep roll-off in the high frequency region. Prior to transmitting or recording to a storage medium, the initial objective is to increase the amount of high frequency energy in order to improve the signal-to-noise ratio. Boosting high frequency energy gives more information because more energy exists in the spectrum of voiced segments at lower frequencies than higher frequencies. It is used in speech processing because of the rapid decaying spectrum of speech. This decay in high frequency part is seen to be suppressed during the sound production mechanism of humans with amplifying the importance of high frequency formats.

The formula for pre-emphasis filter is

◖ ▪ سي ● ▪ طي ♪ ᶻ ● ▪ طي ٥ (3.1)

where the per-emphasis coefficient is ؞ and its values should be in the range between 0.9 and 1, the output signal is defined as Ù Î , the input signal is defined as Ø Î and Ø Î طي ρ defines as the last input signal. The most common used coefficient of ؞ is 0.97.

3.3.2.3 Framing and Windowing

Framing makes the non-stationary speech signals to stationary signals. The purpose of windowing is to reduce the effect of the spectral artifacts that result from the framing process [25]. Speech is a non-stationary signal. Spectral features are extracted from a small window of speech that characterizes a particular sub-phone. It can be done by using a window which is non-zero inside some region and zero elsewhere, running this window across the speech signal and extracting the waveform inside this window.

A signal is divided into shorter frames having distorting spectral information in segmented speech. To minimize this distortion, all samples in frames are then multiplied with a window weighting function before feature extraction. Multiplying the speech waveform by using a window function can gradually prevent in any change between frames. The time for which the signal is considered for processing is called a window, data acquired in a window is called a frame, and features extracted for a certain millisecond are called a frame rate. The window length is bigger than the frame interval for preventing any loss of information during transitions between the frames. Overlapping frames are used to avoid missing data, so that the current frame covers a part of information from the previous frame. Therefore, the windowing process has three parameters: the width of window in milliseconds, the offset between successive windows, and the shape of the window. Usually, 30 to 50 frames per second can give intelligible speech. The frames can be independently analyzed and represented as a single feature vector. The speech signal is usually divided into frames of small duration typically 20ms to 40ms with overlap of 10ms to 15ms for short-term spectral analysis. In this research, the speech signal is split into a number of overlapping frames; each frame is 25ms long and shifted by 10ms. By multiplying the value of the signal at time n, x[n], with the value of the window at time n, w[n], the windowing can be done.

◗ ▪ سي ◌ ▪ z ● ▪ ع جى ▪ جى لـ طي ٥ (3.2)

There are many window functions available for signal processing applications namely: Triangular window, Hamming window, Rectangular window, Hanning

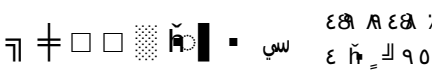
window, and so on. Having a narrow main lobe and low side lobe levels in their transfer function is a good window function.

The most useful windowing in feature extraction is Hamming window shrinking the values of the signal toward zero at the window boundaries, avoiding discontinuities. It is used to minimize the effects of transmission overlap and spectrum flux in the Discrete Fourier Transform (DFT) by highlighting the signal content at the center of the frame that corresponds to its edge. A Mel scaled filter bank passes the DFT's magnitude spectrum. Instead of a linear scale, Mel scale is used because the human auditory system consists of filter. Its center frequencies and crucial bandwidths resolve non-linearly over the audio spectrum. The following equations are for rectangular window, Hanning window and Hamming window.

(3.3)

The simplest window is the rectangular window but it abruptly cuts of the signal at its boundaries which are replacing N values by zeros, and the waveform will be suddenly turned on and off.

(3.4)



(3.5)

In this work, Hamming window is used to minimize the signal discontinuities at the beginning and end of each frame. The choice of window is critical for analysis of speech signal utilizing because the proper use of windowing in the preprocessing step not only reduce the frequency component leakage but also make spectrum smoother. Rectangular window has easily lost the details of the waveform of speech signal.

Therefore, Hamming window is better for using to truncate the long signal sequence into short time sequence and is more effective to decrease frequency

spectrum leakage with the smoother low pass effect. Moreover, it has relatively stable spectrum for speech signal and it helps to enhance the characteristics of original signal.

3.3.2.4 Discrete Fourier Transform

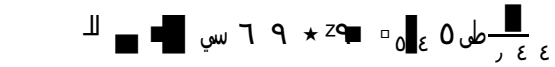
Discrete Fourier Transform (DFT) performs extracting the spectral information from the windowed signals. It is used to get the spectral content for the discrete frequency bands of a discrete temporal signal. The output ώ ̃Ω is a complex number in the original signal representing the phase and magnitude of the frequency component. The formula for DFT is as follows:

ñ̃ ï سي B̈F̈٩٥ ȯ Ĩ ̋Ḧ٩ ̈E (3.6)

For converting the signals from time domain to frequency domain, the commonly used algorithm for computing the DFT is the Fast Fourier Transform (FFT). It is also used to increase the speed of computation time and computes for N values that have the powers of 2 because of some limitations.

3.3.2.5 Mel Filter Bank

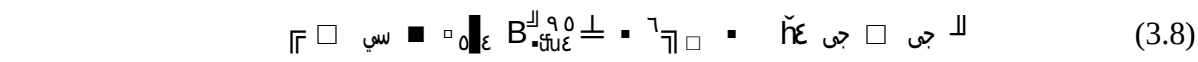
Mel frequency warping means converting the frequency domain signal to ‘Mel’ frequency scale (a measuring unit of the perceived frequency or pitch of a spoken tone) based on how the human hearing perceives frequencies. The frequency content of sounds for speech signals in human perception does not follow a linear scale because human earing is not equally sensitive to all frequency bands. Like the actual, normal frequency measure in Hz, the subjective pitch is measured on ‘Mel’ scale, a unit of pitch. The ‘Mel’ frequency scale is the linear frequency spacing below 1 kHz and the logarithmic spacing above 1 kHz. Removing the pitch of speech signal and smoothing the magnitude spectrum is the main tasks of Mel-filter bank. The formula for computing ‘Mel’ scale for a particular frequency is



(3.7)

3.3.2.6 Computing Log

Taking the log of the power at each of the ‘Mel’ spectrum values for compressing dynamic range of values is done in this step. Using a log makes the feature estimates less sensitive to variations in input such a power variation due to the speakers’ mouth moving closer or further from the microphone. In general, the human response to signal level is logarithmic. Humans are less sensitive to slight differences in amplitude at high amplitudes than at low amplitudes. Taking the log implements that intuition by establishing a bank of filters that accumulate energy come from each frequency band having 10 filters spaced linearly below 1 kHz and the remaining filters spread logarithmically above 1 kHz [23]. The log of each ‘Mel’ spectrum is calculated as follows:

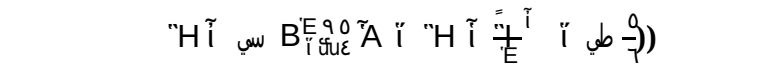


(3.8)

where M is the number of ‘Mel’ filters, ώ è is the N point FFT of the input speech signal’s specific window frame, and ΎΟ è is the ‘Mel’ filter transfer function.

3.3.2.7 Discrete Cosine Transform

Discrete Cosine Transform converts the log ‘Mel’ spectrum back to time domain to obtain the MFCC features. DCT gathers most of the information existing in the signal to its lower order coefficients by discarding the higher order coefficients and also de-correlates the energy because filter banks are all overlapping and the filter bank energy are quite correlated with each other.



(3.9)

where ώ̃ è is the resulted MFCC features. Moreover, additional energy feature is added as the 13rd feature to MFCC. The features with 12 plus 1 energy dimensional cepstral coefficients with double delta are used in this work.

3.3.3 Delta MFCC

Adding time derivatives to the basic static parameters enhances the performance of a speaker recognition system. By analyzing frame, the speech signal in time domain is lost in the frame. Delta feature is a widely method used to encode some of the dynamic information of spectral features. The time derivatives of the features are estimated by some method, and then this derivative estimation is appended to the feature vectors, giving a feature vector having the higher dimensional feature. As an example, if 12 plus 1 (energy) dimensional Mel-Frequency Cepstral Coefficients are appended with their time derivative estimates, the dimensionality of the new feature vector is 13 + 13 = 26. As one of the interesting features of the classification system, one of the important characteristics is channel invariance. But this property does not exist for MFCC features that will be distorted by the spectral divergence in a channel.

However, the property of channel invariance exists in the dynamic features reflecting changes over time. Thus, the first- and second-time derivatives of MFCC are currently adopted by the researchers in the research community as extra features. The improvements of identification accuracies when used with MFCC have been shown by these time derivations [26]. Delta is the type of velocity feature. The acceleration feature of a frame is a double delta. The cepstral coefficients of 12 dimensional for each frame are obtained by extracting the cepstrum from the previous step with the use of DCT. The energy from the frame is added as a 13rd feature. The change between frames existed in the corresponding cepstral or energy feature is represented by each of the 13 delta features.

The computation of the delta MFCC Ω_{δ} for frame δ is mathematically given by equation.

$$\Omega_{\delta} = \frac{\sum_{\theta=1}^{\Theta} \theta \left(\omega_{\delta+\theta} - \omega_{\delta-\theta} \right)}{2 \sum_{\theta=1}^{\Theta} \theta^2} \qquad (3.10)$$

where the delta window size is represented as Θ and ω_{δ} represents the MFCC at frame δ. The use of delta cepstral features is a disadvantage to an increase in feature vector size, although they have the benefit of improving accuracy at certain cases. But delta features are important for enhancing the robustness of the recognition. Delta and

double delta features improve the accuracy of the speaker recognition system. Therefore, 39 dimensional Mel-Frequency Cepstral Coefficients (MFCC) are used as the acoustic feature in this work.

3.4 GMM-UBM based Speaker Model

Gaussian mixture models (GMMs) are widely used in the speaker recognition. Literature shows that GMM based probabilistic models yields better and more reasonable outcomes for training of speaker recognition applications in both text-dependent and text-independent tasks. A speaker probabilistically is represented through a multivariate Gaussian probability density function (pdf) recognizing that speech production is inherently non-deterministic. Each statistical variable corresponds to a single acoustic sound class as a state because this is a multi-dimensional structure. GMM takes as input a sequence of vectors provided by the MFCC and uses it to build one model per speaker.

3.4.1 Gaussian Mixture Model (GMM)

Gaussian mixture models (GMMs) can provide greater flexibility and precision in modeling the underlying statistics of sample data [32]. These are a type of density model which comprise a number of Gaussian component functions. These component functions are combined to provide a multimodal density. A Gaussian mixture density is a weighted sum of M component densities as shown in Figure 3.2 and the equation is as follows:

$$p(\vec{x}\mid\lambda)=\sum_{i=1}^{M} w_i\, g(\vec{x},\vec{\mu}_i,\Sigma_i) \tag{3.11}$$

where $\vec{x}$ is a D-dimensional random vector, $g(\vec{x},\vec{\mu}_i,\Sigma_i)$, i= 1,…,M are the component densities and w_i, i= 1,…,M are the mixture weights of i^{th} component. Each component density is a D-variate Gaussian distribution function of the form

$$g(\vec{x},\vec{\mu}_i,\Sigma_i)=\frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}}\exp\left\{-\frac{1}{2}(\vec{x}-\vec{\mu}_i)'\Sigma_i^{-1}(\vec{x}-\vec{\mu}_i)\right\} \tag{3.12}$$

with the number of dimensions D, mean vector μ_i extracted from feature matrices and covariance matrix Σ_i which provides information about the difference between features. The mixture weights satisfy the constraint that $\sum_{i=1}^{M} p_i = 1$. The complete Gaussian mixture density is parameterized by the mean vectors, covariance matrices and mixture weights from all component densities. These parameters are represented by the Equation 3.13.

$$\lambda = \{p_i, \mu_i, \Sigma_i\} \quad i = 1, \ldots, M \tag{3.13}$$

Each speaker is represented by a GMM and is referred their respective speaker model by λ. Each speaker is attributed a GMM. The spectral shape of the i^{th} acoustic class can be represented by the mean μ_i of the component density and covariance matrix Σ_i is represented as the average spectral shape's variations. Depending on the selection of covariance matrices, GMM can have several different structures. The three classifications of covariance matrix are (1) one covariance matrix represented for one Gaussian component (Nodal covariance), (2) one covariance matrix describing for all Gaussian components in a speaker model (Grand covariance) and (3) a single covariance matrix which shares to use by all the speaker models (Global covariance). In addition, the covariance matrix can also be full or diagonal. For speaker modeling, nodal (one covariance matrix intended for one Gaussian component) and full covariance matrices are taken into account in this work. Figure 3.2 shows the description of M component Gaussian mixture densities.

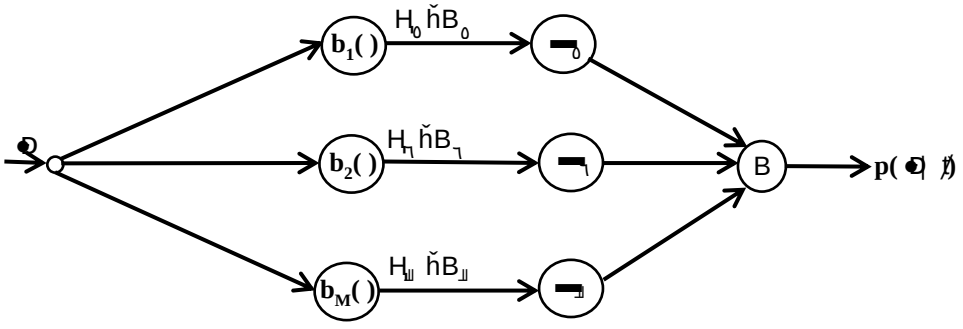


Figure 3.2 Description of M component Gaussian mixture densities. A Gaussian mixture density is a weighted sum of Gaussian densities, where p_i, and b_i ( ), i=1, … M, are the mixture weights and the component Gaussians.

The goal of speaker model training is to estimate the parameters of the GMM, λ which matches the distribution of the training feature vectors. The method employed in this situation is the Maximum likelihood estimation. For estimating the parameters of a GMM, there are several available techniques. Among them, Exception-Maximization (EM) algorithm is used to estimate the parameters of a GMM model in this work. It served as an initialization to estimate a full covariance UBM. The basic idea of this algorithm is beginning with an initial model λ to estimate a new model $\bar{\lambda}$ in order to such that $p(X|\bar{\lambda}) \geq p(X|\lambda)$. The new model then becomes the initial model for the next iteration and the process is repeated until some convergence threshold is reached that is until the parameters of λ reach a stable value.

On each EM iteration, the parameters are updated and re-estimation formulas are used which guarantee a monotonic increase in the model's likelihood values for mixture weights, means and variances. For re-estimating the mixture weights, Equation 3.14 is used. Equations 3.15 and 3.16 are used for re-estimating the means and variances.

$$\bar{w}_i = \frac{1}{T}\sum_{t=1}^{T} p(i|x_t, \lambda) \tag{3.14}$$

$$\bar{\mu}_i = \frac{\sum_{t=1}^{T} p(i|x_t, \lambda)\, x_t}{\sum_{t=1}^{T} p(i|x_t, \lambda)} \tag{3.15}$$

$$\bar{\sigma}_i^2 = \frac{\sum_{t=1}^{T} p(i|x_t, \lambda)\, x_t^2}{\sum_{t=1}^{T} p(i|x_t, \lambda)} - \bar{\mu}_i^2 \tag{3.16}$$

where w_i, μ_i and σ_i^2 are the arbitrary elements of the vectors w, μ and σ^2 respectively. The final step of maximum likelihood is to attain the *a posteriori* probability for each feature vectors. The *a posteriori* probability for acoustic class i is given by

$$p(i|x_t, \lambda) = \frac{w_i\, p_i(x_t)}{\sum_{k=1}^{M} w_k\, p_k(x_t)} \tag{3.17}$$

Selecting the order M of the mixture and initializing the model parameters prior to the EM algorithm are the two critical factors in training a Gaussian mixture speaker model [32]. In speaker identification, a group of S speakers' $S = \{1, 2, \ldots, S\}$ is represented in GMM as $\lambda_1, \lambda_2, \ldots, \lambda_S$. Finding the speaker model with the maximum *a posteriori* probability for a given observation sequence is the primary objective, and it is represented as

$$\hat{S} = \arg\max_{1 \le k \le S} \Pr(\lambda_k | X) \tag{3.18}$$

According to the Bayes' rule, the speaker model with the maximum *a posteriori* probability becomes

$$\hat{S} = \arg\max_{1 \le k \le S} \frac{p(X|\lambda_k)\Pr(\lambda_k)}{p(X)} \tag{3.19}$$

By (i) assuming equally likely speakers (equivalent to $\Pr(\lambda_k) = 1/S$) and (ii) observing that $p(X)$ is the same for all the speaker models, the above classification rule can be made even simpler. Therefore, Equation 3.19 reduces to

$$\hat{S} = \arg\max_{1 \le k \le S} p(X|\lambda_k) \tag{3.20}$$

Finally, the speaker identification system calculates

$$\hat{S} = \arg\max_{1 \le k \le S} \sum_{t=1}^{T} \log p(\vec{x}_t|\lambda_k) \tag{3.21}$$

using the logarithms and the independence between the observations in which $p(\vec{x}_t|\lambda_k)$ is given in Equation 3.11 [32].

3.4.2 Universal Background Model (UBM)

Universal background model (UBM) in speaker recognition systems is a widely used effective framework that has found great success. UBM is a large

Gaussian mixture models (GMMs) model representing general, person-independent feature characteristics to be compared against a model of person-specific feature characteristics when making accept and reject decision. It is used in biometric systems. It is an improvement in the field of speaker recognition using GMMs and typically characterized as a single GMM trained with a huge amount of data from a large set of speakers. The UBM technique is consolidated into the GMM based speaker identification system to reduce the required time for recognition significantly. It is the key element of i-vector system for collecting statistics from speech utterances. A large UBM with full covariance matrices is employed to collect statistics for the evaluation of i-vectors, which includes many computations. The method is to first select a speaker specific trained model, and then determines a likelihood ratio of the match score of a test speech sample with the trained model and the universal background model [59].

A GMM characterizing the specific features of all different speakers is a UBM in the speaker identification system. Each speaker model can be developed by Bayesian adaptation from the UBM using particular speaker training speech instead of employing the maximum likelihood training. The mixture components concerning with each adapted speaker models retain an assured correspondence with the UBM because the likelihood value for a feature vector is significantly contributed only in a few of the mixtures of a GMM according to the findings from prior experiments focused for speaker recognition. Therefore, log likelihood score of the speaker model can be evaluated by scoring only the more significant mixtures. The mixtures that have the highest scores from the UBM are computed to obtain these significant mixtures because of the correspondent relation of mixtures between the speaker models and the UBM [60].

3.4.3 GMM Based i-vectors Extraction

Modeling the overall training data variability and compressing the speaker information to a vector which has low dimensional are the aims of i-vector system. It represents the important information about the speaker and all other types of variability. GMM based approach for i-vector extraction estimates the mean values of speaker speech features associated to each component of Universal Background Model (UBM). Figure 3.3 shows the total variability space representations.

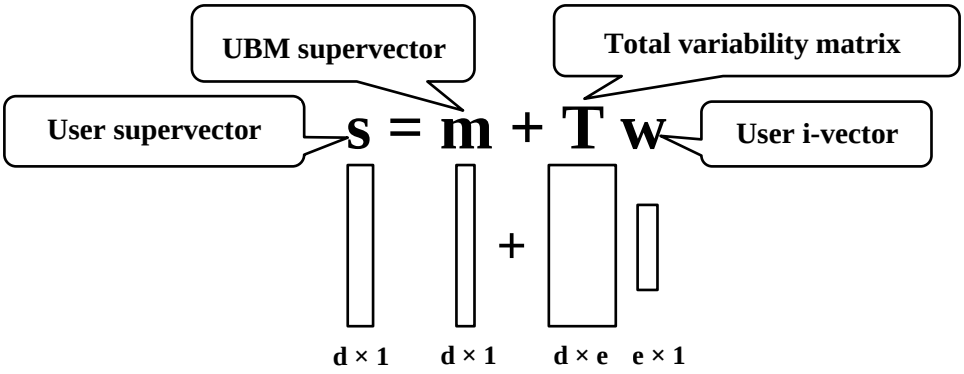


Figure 3.3 Total Variability Space Representations

Identity Vectors or i-vectors known as the lower dimensional vectors have a compact representation of speech signals. These are derived by transforming from supervectors. The speaker- and channel-dependent GMM supervector ί can be modeled as this main idea:

▼سي □ طى ╣ ◌ (3.22)

where ά is the GMM-UBM mean supervector, ″Yis a low rank matrix representing the total variability space, and ύ is the total i-vectors following a standard normal distribution. The global parameters ″Yand ά can be estimated by using EM algorithm. The MAP point estimate of ύ is i-vector, and i-vector extractor is referred by ″Y It can be easily extracted an i-vector ύ with ί, ά and a trained ″Ymatrix. After i-vectors are extracted, the PLDA model can be used to calculate the log likelihood ratio scores for the hypothesis test that determines if the two i-vectors of utterances are produced by the same speaker.

3.5 Neural Network based Speaker Model

In automatic speech recognition, successful applications of Deep Neural Networks (DNNs) have supported a forceful motivation to exploring attempts of possible consequences in speaker recognition task by implementing with DNN architecture. GMM with a lot of components is inefficient because only a small portion of the data are applied for each parameter whereas an overwhelming amount of data constrains to each parameter in a product model. The non-linearity of the two models is not the same. Although GMMs need uncorrelated data, DNN can exploit

correlated data forming multiple frames of input coefficients. Moreover, GMM learning employs Expectation Maximization (EM) algorithm which is much easier to parallelize while DNN learning uses stochastic gradient descent.

DNNs consist of a large output layer and numerous hidden layers with numerous non-linear units. It has been utilized recently to obtain speaker-related attributes and requires a large amount of processing data in order to train the model, but as a consequence, it can provide the optimal results although model training takes a lot of time. There are two approaches of using DNN in speaker identification: one uses DNN for extracting the acoustic features, and another uses DNN for modeling the acoustic features. Modeling is performed with the use of i-vector framework, and DNN is replaced instead of using the traditional universal background model [62]. In this work, the efficiency of TDNN implementation is investigated at feature modeling. When it comes to DNN-based speaker identification models and systems, TDNN is an effective architecture.

3.5.1 Time Delay Neural Network (TDNN) Based Model

Time Delay Neural Networks (TDNNs) have been used as the predominant form of neural network architecture for the purpose of speech and speaker recognition [61]. The first TDNN is developed by Weibel and Lang in 1989 designed to handle the frame-based analysis of speech [7]. It is a type of feed forward network architecture which has strong ability in its context modeling and effective in classifying patterns with shift invariance and modeling long term temporal contexts at each layer of network. TDNNs construct the initial transforms are learned within narrow contexts in lower layers and learned the hidden activations from the longer temporal relationships in deeper layers (higher layers) although the initial layers learns an affine transform for the whole temporal context in a standard DNN. It can also learn wider temporal dependencies in both large and small amount of training data. Therefore, when modeling long term temporal dependencies derived from short term acoustic features like MFCC, TDNN is utilized.

Because it is effective at obtaining features of long-range temporal context dependencies [7] and improving the x-vector learning capability by obtaining more robust speaker characteristics, TDNN is the progenitor of convolutional neural networks [18]. It is trained to extract “x-vectors” which has in segment level for text

independent speech recognition [69]. The architecture has a continuous input that is delayed and sent as an input to the network. As an example, consider training a feed forward neural network being trained for a time series prediction. The desired output of the network is the present state of the time series and inputs to the neural network are the delayed time series of past values. Hence, the output of the neural network is the predicted next value in the time series which is computed as the function of the past values of the time series [7]. Although DNN diminishes the system performance for under resourced data while there is a lack of the amount of training data required and the fully connected nature of DNN can't alleviate over fitting, TDNN outperforms DNN and Gaussian Mixture Model-Universal Background Model (GMM-UBM). It is because it can model the context information at each layer of network and captures long term temporal dependencies in shift invariance. The advantages of using TDNN are reducing the number of weights (require fewer examples in the training data and faster learning), and executing faster in the network in comparison of fully connected multilayer perceptron. Moreover, if there is a limited amount of training data because subsampling excludes duplicate weights, it has an advantage in fast convergence.

Feature learning, statistical pooling and identification process are the three parts of TDNN architecture for speaker identification. Five-time delay layers are used in feature learning in this study to learn frame-level speaker features that are modeled to deliver the required information in an appropriate fashion. The experiments are analyzed on eight different slicing parameters of time delay layers of network to look into which network context best captures the model's efficiency. Secondly, statistical pooling is used to calculate the mean and standard deviation of the frame level information extracted from a speech segment. To discriminate the speakers at segment level, the subsequent nonlinear layers get activations from this pooling layer. The third component, speaker classification uses one full connection layer to separate each speaker based on the number of speakers in training data. The penultimate full connected layer's 512-dimensional activations are retrieved as an x-vector after training. The network can receive these fixed length feature vectors that are generated for each utterance as input. During extracting the vectors, the only difference between GMM and TDNN based systems is the model implemented to compute posteriors. TDNN posteriors create the adequate statistics for extracting the vector in conjunction

with the speaker features. Figure 3.4 represents the fully connected TDNN architecture without subsampling.

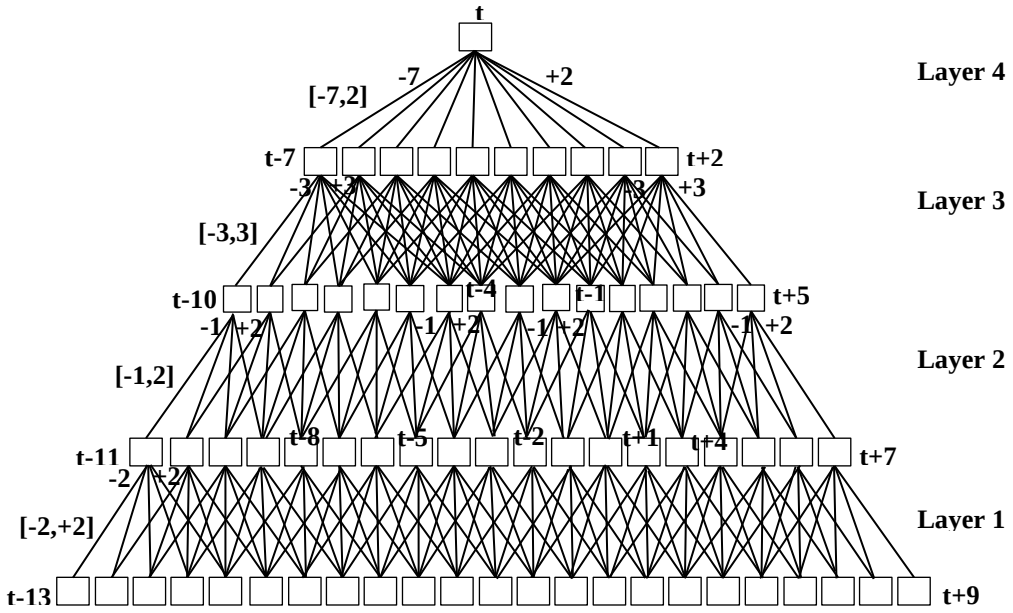


Figure 3.4 Example of TDNN Architecture without Subsampling

Moreover, a frame level VAD is used to filter the speaker recognition features. To maintain the correct temporal context, the feature frames cannot remove from TDNN input features. Instead, VAD results are reused to filter out posteriors corresponding to non-speech frames [15]. These fixed length vectors which are extracted from each utterance can be inputted to the network. By using greedy layer-wise supervised training, TDNN learns the way to update the network parameters [63]. To train neural networks, this supervised training was used together in Kaldi, speech recognition toolkit supporting multiple GPUs in the training. It was shown that x-vector based model can achieve better speaker recognition performance compared to the traditional i-vector approach. In the event that the system is trained utilizing a fully connected TDNN configuration without subsampling, the hidden layers of the network will compute a lengthy training period. But the training time and computational cost are saving and effective if subsampling technique is used in the network.

3.5.2 TDNN with Subsampling Technique

TDNN is optimized by removing duplicated weights in the networks. Subsampling technique is used together in TDNN with the aim of reducing duplicated weights between nodes in network. The duration of training is reduced when these duplicated updates are discarded. Subsampling decreases the size of the model and speeds up training time by allowing gaps between feature frames instead of splicing together consecutive temporal windows at each layer, which saves money on computing the hidden activations at all time steps.

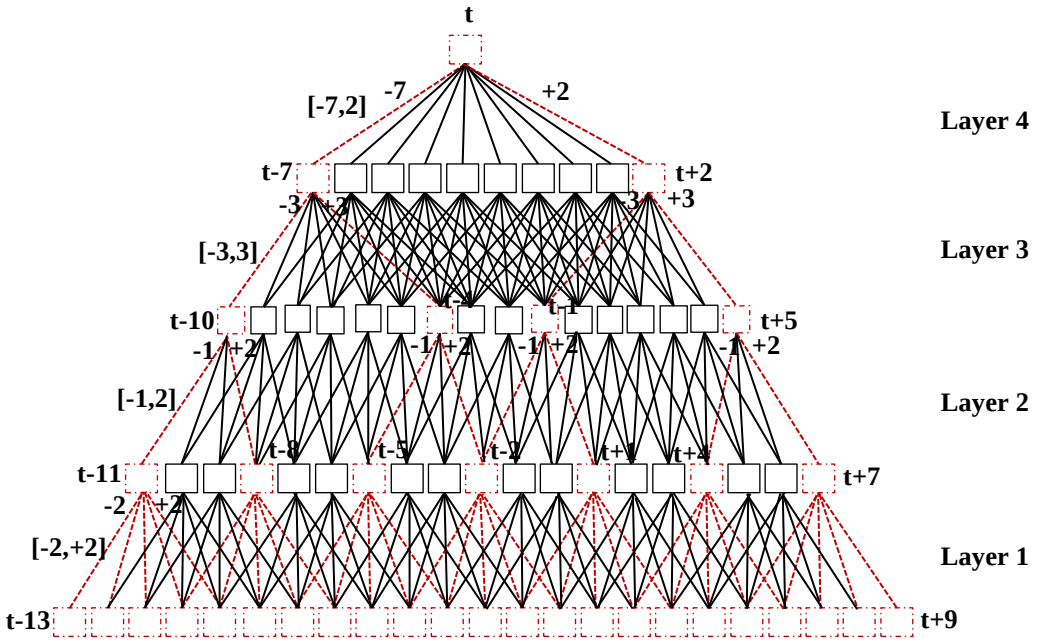


Figure 3.5 Example of TDNN Architecture with Subsampling

These are the justifications behind the subsampling approach used in TDNN architecture. If subsampling technique which has the property of selective computation of time steps is used, the forward pass and back propagation minimize all necessary computation. The benefits of subsampling technique include a reduction in the number of parameters and an increase in computational efficiency. By leaving a gap between frames, it does not connect two or more inputs in a hidden layer. All input features can be learned by the model if the interval between frames is permitted. This is because TDNN has a lengthy context reaching up to the upper layer. Moreover, by minimizing edges and nodes number in the network, the number of parameters which can represents the model is reduced. Nodes and weights

representations are depicted by red dashed line. These are only updated when subsampling technique is used as shown in Figure 3.5.

The frames are not spliced more than two frames in the hidden layers of the network. For instance, as shown in Figure 3.5, the frames are spliced together the input at the current frame minus 7 and the current frame plus 2 are described by the notation {-7, 2}. The configuration in Figure 3.5 splices together frames ὸطٍ ςthrough ὸطٍ ςat the input layer (written as context {-2, -1, 0, 1, 2} or more compactly as [-2, 2]) and then the frames are spliced at offsets {-1, 2}, {-3, 3}, and {-7, 2} in three hidden layers. Figure 3.6 describes an example of context modeling over 3 frames in a TDNN.

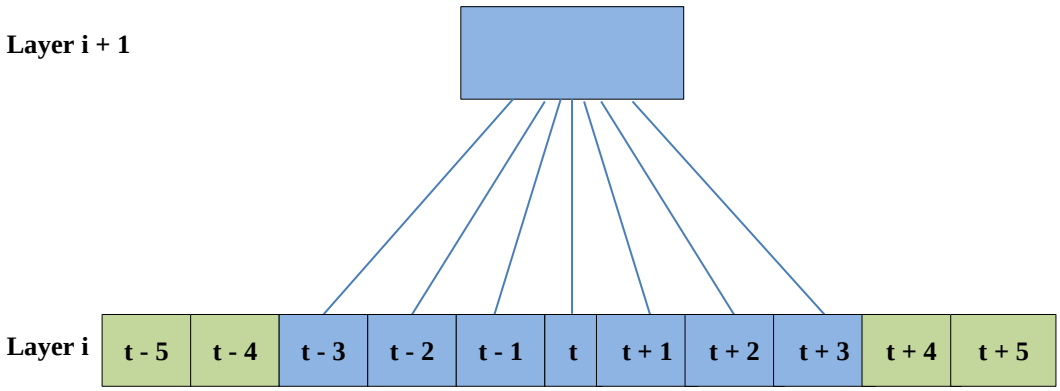


Figure 3.6 Example of Context Modeling over 3 Frames in a TDNN

Table 3.1 compares with a hypothetical setup without subsampling and shows these contexts (on the right). The differences between the offsets at the hidden layers were chosen to be multiple of 3. This way is implemented to compute a small number of hidden layer activations for each output frame. In Figure 3.5, the frames in red dashed lines are those needed to evaluate. The current subsampling approach reduces the total amount of computing required because time steps are computed selectively. The TDNN training time speeds up and the model size reduces by using subsampling technique. Contiguous frames that were spliced together at hidden layers would either drastically increase the number of parameters of drastically decrease the size of the hidden layer.

Table 3.1 Example Context Specification of TDNN

Layer	Input Context without Subsampling	Input Context with Subsampling
1	[-2, +2]	[-2, 2]
2	[-1, 2]	{-1, 2}
3	[-3, 3]	{-3, 3}
4	[-7, 2]	{-7, 2}
5	{0}	{0}

3.5.3 TDNN Based x-vectors Extraction (Speaker Embedding)

Deep Neural Network architecture for embedding is widely studied in [68, 69]. The Kaldi recipe from David Snyder [70] is used in this work. Input features to the network are 39-dimensional MFCCs extracted using a 25 ms Hamming window shifted by every 10 ms. The TDNN embedding can be divided into three parts. The first part operates on the frame level and begins with 5 layers of time delay architecture [18]. The first four layers contain each 512 neurons, the last layer before statistic pooling has 1500 neurons. The consequent pooling layer gathers mean and standard deviation statistics from all frame level inputs. The single vector of concatenated means and standard deviations is propagated through the rest of the network. This process aggregates information across the time dimension so that subsequent layers operate on the entire segment. The extracted embedding part of the network consists of two hidden layers each with 512 neurons and the final output layer. The output layer has a dimensionality related to the number of speakers. As nonlinearities in hidden layers, Rectified Linear Units (ReLU) is used by the network. On the output layer, soft-max is used. The network is trained by optimizing multi class cross entropy objective function as shown in Equation 3.23. The network is trained to classify N speakers in the training data for several epochs using natural gradient stochastic gradient descent [71]. After training, the embeddings are extracted from the affine component layer. The network is trained to predict speakers from variable length segments rather than frames. Suppose there are K speakers in N training segments. Then, the probability of speaker k given T input frames $x_1^{(n)}, x_2^{(n)}, \ldots, x_T^{(n)}$ is $P(spkr_k \mid x_{1:T}^{(n)})$. The quantity d_{nk} is 1 if the speaker label for segment n is k, otherwise it is 0.

(3.23)

3.6 Probabilistic Linear Discriminant Analysis (PLDA)

Multiple input feature vector frames can be employed with Simplified or Gaussian probabilistic linear discriminant analysis (also known as Simplified or Gaussian PLDA) which is used to analyze fixed length feature vectors. This is made possible through the use of subspace covariance modeling. Recently, identity vectors have been successfully modeled using this hierarchical generative latent variable model, which takes the feature vectors in highly correlated subspaces. It can also be seen as a probabilistic version of classical LDA. It was originally proposed by Price et al. for face recognition [47]. Later, it was adapted for modeling i-vector distributions for speaker verification by Kenny et al. [48-50] as a generative model.

PLDA provides a linear transformation of n dimensional feature vectors into m dimensional space (m<n), so that samples belonging to the same class are close together but samples from different classes are far apart from each other [51]. The class center is generated by using continuous non-linear functions even from single example of unseen class. In hypothesis testing, the two examples from previously unseen class can be compared by determining whether they belong to same class. On the other hand, the two examples of unseen classes' scores using PLDA can be compared in the task of recognition by comparing likelihood of examples from same class versus likelihood of examples from different. PLDA can only improve the identification rate, but also diminish the feature dimension and computational cost [13]. In feature vector-based speaker identification, it is employed for scoring. The backend consists of i-vector mean subtraction and length normalization followed by PLDA scoring. Scoring method of PLDA is as implemented in [3, 20].

3.6.1 Linear Discriminant Analysis (LDA)

A technique for dimensionally reduction that projects the data onto a subspace which satisfies the requirement of maximizing *between class* variance and minimizing *within class* variance is called the standard Linear Discriminant Analysis (LDA).

Centering and projecting the vector representations need 150-dimensional Linear Discriminant Analysis (LDA) tuning. LDA can improve the classification rate in addition to lowering the calculation cost and feature dimension. Therefore, the representations (i-vectors or x-vectors) are centered and projected using LDA. The LDA can be defined according to the following:

$$\text{LDA} = \arg\max_{W} \frac{|W^{T} S_{b} W|}{|W^{T} S_{w} W|} \tag{3.24}$$

The *between-speaker* covariance matrix and the *within-speaker* covariance matrix are S_b and S_w defined by:

$$S_b = \frac{1}{S} \sum_{s=1}^{S} (w_s - w)(w_s - w)^{T} \tag{3.25}$$

$$S_w = \frac{1}{N} \sum_{s=1}^{S} \sum_{i=1}^{n_s} (w_{si} - w_s)(w_{si} - w_s)^{T} \tag{3.26}$$

In above two equations, the number of utterances for speaker s is n_s, the total number of utterances is N, the i-vectors of sessions of speaker s is w_{si}, the mean of all the i-vectors of speaker s is w_s, and w represents the overall mean of the training data [65].

In the speaker recognition domain, the better performance achieved [66] by replacing S_b and S_w with the scatter matrices:

$$\tilde{S}_b = \sum_{s=1}^{S} (w_s - w)(w_s - w)^{T} \tag{3.27}$$

$$\tilde{S}_w = \sum_{s=1}^{S} \sum_{i=1}^{n_s} (w_{si} - w_s)(w_{si} - w_s)^{T} \tag{3.28}$$

After dimensionality reduction, the representations are length normalization and modeled by PLDA.

3.6.2 Probabilistic LDA (PLDA)

Probabilistic Linear Discriminant Analysis (PLDA) has begun to be the state-of-the-art configuration for speaker recognition. A simplified variant of Gaussian PLDA successfully establish an i-vector representation generative model. For the [illegible] speaker, the vector [illegible] representing the [illegible] recording can be expressed as

[illegible] (3.29)

Here, [illegible] is the speaker dependent part and [illegible] is the recording dependent part. The overall training data is denoted by the symbol [illegible]. [illegible] and [illegible] are sets of basic vectors that express the speaker's subspace *between speaker* variability, and the channel's subspace *within speaker* variability, respectively. It is assumed that the latent variables [illegible] and [illegible] which represent a specific channel and speaker respectively, have typical normal distributions. The remaining residual variability represented by [illegible] The residual term [illegible] is assumed to have a normal distribution with a diagonal covariance matrix. The *within speaker* variability is modeled by a full covariance residual term omitting the channel subspace. The generative model for i-vector is represented by

[illegible] (3.30)

The term [illegible] assuming as the residual of *within speaker* variability is used to have a normal distribution with full covariance matrix $\sum$. The two covariance models specify as a special case of the simplified PLDA model having the speaker factors S is full rank [64].

3.6.3 PLDA on Kaldi

Kaldi is an open-source and widely used speech recognition toolkit [46]. PLDA on Kaldi follows the formulation proposed in [72]. For Kaldi PLDA, the Simplified PLDA with full rank [illegible] is the conceptual starting point. Kaldi PLDA is

only associated with getting the average i-vector for each speaker [illegible], which is distributed according to

$$[illegible] \tag{3.31}$$

where [illegible] is the numbers of extracted i-vectors for speaker [illegible]. As the data likelihood function, Equation 3.31 is then applied. The average i-vector (a single sample) which is assumed to obey the decomposition collapses from all the extracted i-vectors for each speaker:

$$[illegible] \tag{3.32}$$

where [illegible] models the *between class* variability with the covariance [illegible] and the average residual

$$[illegible] \tag{3.33}$$

models the *within class* variability. The excepted complete log likelihood function for the EM algorithm is optimized to iteratively estimate Γ and Λ, as follows:

$$[illegible] \tag{3.34}$$

The total hidden variables [illegible], having good convergence are selected by the EM iteration in Kaldi PLDA. However, in Kaldi PLDA, only the additive decomposition applies to the average i-vector. For estimating *between class* covariance, although this is helpful, it is harmful for estimating *within class* covariance. Kaldi PLDA also carrys out simultaneously Λ diagonalization and Γ diagonalization at the testing phase. The significance of computational saving is less than others because Kaldi only calculates the likelihood of an individual average i-vector [67].

3.6.4 Likelihood Computation

There are various scoring strategies for utilizing the PLDA model to get a likelihood ratio for a given speaker trial after estimation of the PLDA meta-parameters through an iterative EM algorithm. Depending on whether the two vectors (target and test) were generated by the same speaker or not, the log likelihood ratio for the test case in the two vectors scoring is directly determined. The log likelihood ratio is calculated for each test case in order to identify which speaker's test vectors correspond to which ones in order to score numerous sessions.

3.6.4.1 Scoring for Two Vectors

The scoring approach in PLDA for both i-vector and x-vector has been applied. It can be employed to evaluate likelihood scores between test vectors and a set of enrollments once the PLDA model is trained. In cases of vectors w_t for test and w_e for enrollment, the score $S(w_e, w_t)$ of PLDA can be evaluated by deciding the likelihood ratio given by:

$$S(w_e, w_t) = \ln \frac{p(w_e, w_t \mid H_s)}{p(w_e \mid H_d)\, p(w_t \mid H_d)} \qquad (3.35)$$

Here, both i-vectors w_e and w_t comes from the same speaker is indicated as the hypothesis H_s and hence have the same speaker identity variable y in Equation (3.30) while both are independently derived from different speakers is indicated as H_d. According to the given Gaussian assumptions above, the log likelihood ratio can be evaluated in closed form as given in [20].

3.6.4.2 Multi-Session Scoring

The formula in Equation 3.35 is used for scoring the likelihood between two vectors. Moreover, it can be extended to evaluate the likelihood for multiple test utterances and enrollment. If $w_{t1}, w_{t2}, \ldots, w_{tn}$ are available for testing and multiple vectors $w_{e1}, w_{e2}, \ldots, w_{em}$ are available for enrollment, then the following equation can be used to generalize the log likelihood ratio computation function:

$$[illegible] \quad (3.36)$$

The closed form for one test vector w_t and N enrollment vectors is derived as given in [20].

3.6.4.3 Heuristic Scoring Techniques

The heuristic scoring methods are also used instead of multi session scoring. The enrollment i-vectors to be statistically independent given the speaker identity are assumed the likelihood ratio score for multiple enrollment i-vectors as defined in Equation 3.36. Rather than reflecting physical reality, this independence assumption is for mathematical convenience. Different i-vectors derived from the identical target speaker might have more in common than just the speaker identity (for instance, transmission channel or recording environment). In general, i-vectors retrieved from the speech features cannot be considered that it is truly statistically independent. The reason is that the likelihood ratio computation of multi-session scoring will be sub-optimal in a practical setting [20]. As a result, other heuristic scoring methods are used for controlling multiple enrollment i-vectors. Four of the most common heuristic scoring methods are:

1. Averaging vectors: N vectors are evaluated by using Equation 3.37 to obtain the average for a single vector representing the class and then use this average vector to compute the score with the test vector using Equation 3.35.

$$\bar{w} = \frac{1}{N}\sum_{n=1}^{N} w_n \quad (3.37)$$

2. Averaging scores: The score fusion is used in which the two i-vector scoring uses to score the individual enrollment i-vectors and then combined. Average scores of individual enrollment utterance with test utterance to obtain the final score.

$$score(\bar{w}, w_t) = \frac{1}{N}\sum_{n=1}^{N} score(w_n, w_t) \quad (3.38)$$

3. Max score: Take the maximum score value while scoring with the individual test utterance and enrollment utterance.

(3.39)

4. Session pooling: A single enrollment i-vector can be obtained from multiple enrollment utterances by pooling sessions in comparing with utilizing multiple enrollment i-vectors. This is done by pooling acoustic feature vectors from all the utterances and estimating zeroth and first order Baum-Welch statistics. A single enrollment i-vector then is obtained as if only a single enrollment utterance was available, and scored using the two i-vector scoring. In other words, concatenate features of all enrollment utterances and compute a vector representation of concatenated utterance.

3.7 Performance Metric

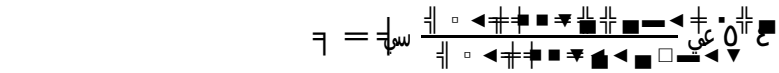
Equal Error Rate (EER) and visual investigations of the detection error tradeoff (DET) curves are commonly employed as the evaluation tools in speaker recognition literature. In most speaker recognition, the performance is shown with the detection error tradeoff (DET) curve. It is implemented by plotting the false negative rate versus false positive rate. The comparison between different DET curves becomes clearer by scaling x- and y-axes with logarithmic transformation. Comparing two curves can sometimes be complicated. This is especially true when two systems perform well in different regions (e.g., one is good at rejecting false candidates while the other is better at picking correct hits). Equal Error Rate (EER) is used to bypass this problem. EER refers to the point where false positive rate and false negative rate equal on DET curve.

Experiments are evaluated using equal error rate (EER) and detecting accuracy (Acc) for the identification task. The EER represents the value at which the false positive rate equals to the false negative rate when comparing each testing sample against all speakers in the test set. The identification accuracy is the percentage of

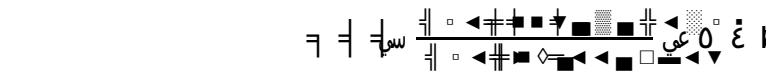
correct identification among all the test trials. Both of these metrics are commonly used for evaluating speaker recognition systems.

3.7.1 Equal Error Rate (EER)

This work appraised the automatic evaluation of Equal Error Rate (EER) for assessing the performance of speaker identification models. It is a common measure of the performance of speaker recognition system showing that of how many False Acceptance (FA) and how many False Rejection (FR) are there. The concession between False Acceptance Rate (FAR) and False Rejection Rate (FRR) is called EER and also the point where FAR and FRR are equal, optimal, and minimal. It is also known as cross over rate or crossover error rate (CER). Equation 3.40 refers to FAR that is a type of error permitting the impostor speaker is wrongly identified as the known speaker and Equation 3.41 referring to FRR computes that the value of refusing incorrectly the real speaker known by the system as impostor. A large number of testing samples are needed to evaluate the performance of EER in identification [6].



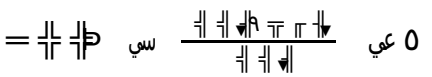
(3.40)



(3.41)

3.7.2 Detecting Accuracy (Acc)

To assess the performance of correctly identifying on every test speech samples, this work describes the recognizing rate with detecting accuracy in percentage of how many test set samples are correctly detected on the whole test set. It takes into account the number of test sets found on speaker models that are different from those which have been correctly tested shown in Equation 3.42.



(3.42)

Here, the test set's accuracy in percentage is ö ώ ώÞ , the total test speech samples is ˝Y˝Y˝Ÿand the wrong detected samples in the whole test set is ώ Ό˝Y.

3.8 Summary

This chapter discussed the methodologies and theoretical background of implementing the speaker identification system. It explains what are the speech signals, how to preprocess the data for processing, data augmentation techniques, data scrutinizing techniques, steps involved in Mel Frequency Cepstral Coefficients (MFCCs) feature extraction process, extracting the i-vector and x-vector, building acoustic speaker models based on GMM-UBM and TDNN, computing the likelihood score using probabilistic linear discriminant analysis (PLDA), different scoring methods and in the end, it describes the performance metric of what is the equal error rate and detecting accuracy for measuring the performance of speaker models.

CHAPTER 4
BUILDING SPEECH DATASETS

This chapter covers the development of speech datasets used in this research. Speech dataset building is an imperative and a very first task for implementing any speaker recognition system. The lack of accurate data in low-resourced, tonal language like Burmese is a major problem in conducting speaker recognition research. Although there are freely and widely available resources in well-resourced language like English, the speech dataset is needed to build first for Burmese which has no available speech dataset easily to use.

4.1 Building Original Speech Dataset

The first stage in any statistically based speaker identification tasks, especially for languages with limited resources, is to gather the speech data. Burmese language can be considered to be a low-resourced, tonal language. The main problem in speaker identification research for Burmese language is the lack of proper data. Therefore, the speech dataset is necessary to develop first. The speech dataset is an abundant collection of recorded audios of spoken languages and is important for statistical based speaker identification. The next steps will be exact to process if the data have been prepared properly. It can also affect the performance of a recognizer. Many speaker recognition systems are constructed on the statistical models based on the speech data. Therefore, speech dataset building is essentially needed to develop the speech related systems. Read speech and Spontaneous speech are the two types of speech. Broadcasting news, word lists, and number sequences are examples of Read speech. Interview speeches and narratives are included in the type of Spontaneous speech. The speech dataset used in this work is constructed by collecting from the two main sources: Web based collected news and daily conversational dialogue recorded for the purpose of training the speaker identification system. The speech required for Burmese dataset was collected with two ways. The first approach is taking the speech that is already been existed from online resources. The second approach uses prepared texts of everyday conversational dialogues to record speech after first gathering a corpus of texts.

4.1.1 Collecting Data from Web-based Sources

As the first approach, the speech data are collected from online sources that is already been recorded. Nowadays, speech data can be gathered from the Internet and is readily available for free access. There are various resources on Internet: news portals and blog posts are lengthier texts and more formal, and only audio files, whereas social media like Facebook and Twitter gives video files and short, conversational, colloquial texts. The collected speech includes both national and international news. The process of gathering Web data takes around two years.

4.1.1.1 Speech Dataset Preparation

Speech data collection is the most important effort in every speech related system. Nowadays, Burmese News, interviews, delivered speech and talks are available on many Web sites. From the sites of Democratic Voice of Burma (DVB)[5], Myanmar Radio and Television (MRTV)[6], Radio Free Asia (RFA)[7], Voice of America (VOA)[8], the speech data are obtained. Moreover, the speech data are also collected from social media, British Broadcasting Corporation (BBC) Burmese News[9], Eleven Broadcasting[10], 7days TV[11], ForInfo News[12], Good Morning Myanmar[13], Breakfast News[14], Irrawaddy Burmese News[15], Mizzima News Myanmar[16], and One News Myanmar Channel[17]. The speech dataset involves both foreign and local news about politics, health, education, sport, speech, crime, business and weather news, etc.

[5] http://burmese.dvb.no
[6] https://mrtv.gov.mm/mm
[7] https://www.rfa.org/burmese/audio
[8] https://burmese.voanews.com/
[9] https://www.facebook.com/bbcburmese/
[10] https://www.facebook.com/elevenbroadcasting
[11] https://www.facebook.com/7DayOnlineTV/
[12] https://www.facebook.com/forinfo/
[13] https://www.facebook.com/GoodMorningMyanmarLive/
[14] https://www.facebook.com/bmrtv/
[15] https://burma.irrawaddy.com/
[16] https://www.mizzimaburmese.com/
[17] https://www.facebook.com/onenewsmyan/

4.1.1.2 Speaker Distribution

Reporters, interviewers, speechwriters and commentators are well-experienced and professional. They have got a clear and concise tone in broadcasting News, making interviews, delivering speech and talks. Female presenters are mostly found on Web News in most fields. Therefore, female speakers are included more than male speakers. There are totaling 111 speakers including 47 male speakers and 64 female speakers. The speakers' age is ranging from 25 years to 70 years old. Figure 4.1 shows the gender distribution of the speakers in Web based speech data.

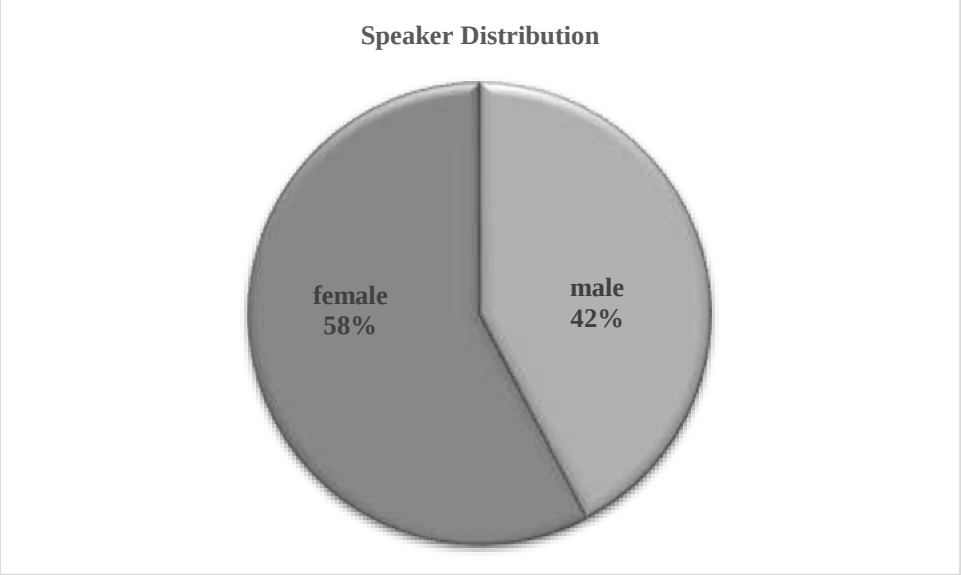


Figure 4.1 Distribution of Speakers in Web News Data with regard to Gender

4.1.2 Recording Data from Daily Conversations

The second way is preparing the text corpus of daily conversational dialogues first and then is recording this text with telephone and microphone. This recording process is done by the 36 internship students from three academic years, 31 Lab members and 31 persons from others. The speech utterances involved in the speech dataset have less in duration than that of Web News domain.

4.1.2.1 Text Corpus Preparation

Daily conversational texts in English with Burmese translations are collected from the Internet. Only Burmese translation texts are taken for text corpus preparation. These texts are the conversational dialogue spoken in restaurants, hotels,

parks, street, telephone and traveling. These collected text are used to record the speech utterances.

4.1.2.2 Speaker Distribution

The speech utterances are recorded by 42 male speakers and 56 female speakers including students, and Lab members from the University of Computer Studies, Yangon, Myanmar as well as those in unrelated fields. The female speakers are mostly found in the dataset because the female speakers outnumber male speakers in our University. Therefore, the male speakers from other fields contributed in the recording. The age of the speakers contained in recording is ranging from 20 years to 53 years old. Figure 4.2 shows the distribution of speakers in daily conversational data with regard to gender.

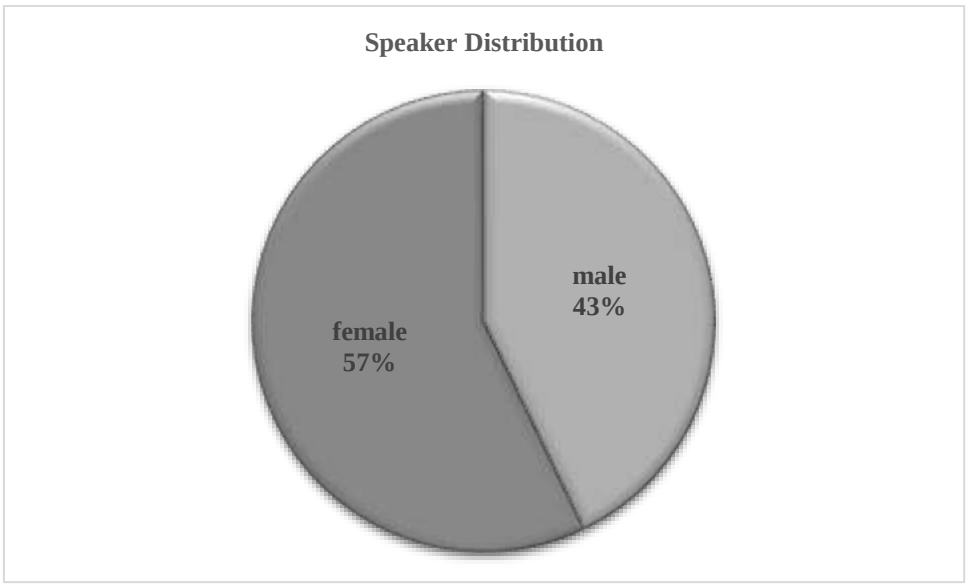


Figure 4.2 Distribution of Speakers in Conversational Data with regard to Gender

4.1.3 Speech Segmentation and Recording

For Web based data, the formats of the collected video and audio files that comes with various format types (.mp4, .wma, .mp3) in different frequency rate (8 kHz, 16 kHz, 44 kHz) are uniformly changed to the files format of .wav. Then, a type of mono channel in 16-bits PCM with the sampling rate of 16 kHz is uniformly fixed to these audio files. For recording the speech utterances, these recording settings are

already prepared before recording. For speech segmentation, Audacity[18] is used. It is cross-platform and open-source multi-track audio recorder and editor software. Silence part and background noise are also discarded as a task of voice activity detection while cutting the audio files. The duration of each segmented audio files is ranging from 3 seconds to 45 seconds. A quiet recording studio in University of Computer Studies, Yangon, Myanmar is used to capture the dialogue of daily conversational discussion. It is a place that has no external disturbance like background noise, environmental noise, and room echo. Tascam DR-100MKIII recording device recommended to be used for audio engineers and designers is used for recording. It can record into the .wav, .bwf, and .mp3 file formats and features the most dependable, user-friendly interface. It is also possible to choose between mono and stereo recording channels. For speech dataset used in this work, the speech utterances are recorded as .wav format of 16 kHz in16-bits mono PCM. All speakers read the utterances at normal pace. Although the speakers utter correct and smooth pace, the recordings are done repeatedly until the speakers do not have a clear tone.

4.2 Building Scrutinized Speech Dataset

Data that is as accurate as possible improves system performance and yields more logical results. The quality of speech is important in order to be able to recognize more accurately, and the amount of speech data required for each speech processing task is also important. There are many audio augmentation techniques in tempo, speed, volume, etc. [14]. To achieve the necessary speech quality for efficient operation, this section offers instructions on how to carefully examine the speech data. The scrutinized speech dataset is built by applying the two scrutinizing techniques to the original collected speech dataset for the purpose of enhancing the system performance.

The first technique is the changing of speech intensity. To know whether changing the intensity in speech improves or not, various intensity rates are analyzed. Since the loudness of sound waves collected from diverse sources has distinct forms, the intensity of each speech segment is analyzed by adjusting with different SNRs levels (-10 dB, -5 dB, 5 dB, 10 dB). In order to determine which dB scale is appropriate for Burmese tones, these data prepared with various dBs were analyzed. It

[18] https://www.audacityteam.org/

was done by setting the same SNRs uniformly to all speech segments. In these experiments, increasing the intensity by 10 dB on the data produced satisfactory results along with a relative improvement over the original dataset's performance on GMM-UBM and TDNN. In reducing the intensity to -10 dB and -5 dB, there is no change in relative improvement. This is because reduction the intensity level than the normal tone is not discriminated the spoken utterances well in Burmese tone. In the part of increasing the intensity to 5 dB and 10 dB, setting to 5 dB raises the performance but setting to 10 dB leads to better performance. Increasing the volume by more than 10 dB can cause the speech to change in tone from the original utterance. Consequently, in this work, the studies were conducted with the maximum sound level set at 10 dB.

Another scrutinizing technique is to analyze the tempo factor for each speech segment in detail. When the speech segments increase and decrease the tempo in the existing speech data by a factor of 0.2 (20%), slowing down the tempo of the current speech gives better results than increasing the tempo factor and the original pace with a relative improvement of GMM-UBM and TDNN. A too fast speech rate may not be accurately recognized, while a slow speech rate may well recognize whether the target speaker is present or not. As a result, slowing down the speaking rate can cause the original utterance to take longer. Based on the experiments, it can be observed that speaking more slowly recognizes the speakers more and better controls the understanding of what is said with the clarity of the vocabulary. It may also improve in automatic speech recognition. Although the speed of the speech utterances was also analyzed with the speed factor, this work does not use the speed factor because changing the index factor affects both pitch and tempo, causing changes in the spectral shape of speech segments. This can cause the loss of speaker-specific information in speech segments. Using the tempo factor for analyzing the pace of speech utterances does not impact on the pitch of speech segments.

Therefore, the scrutinized speech dataset is created by applying with the two scrutinizing methods of increasing the intensity level to 10 dB and downing the tempo factor to 0.2 times to all the speech utterances in the original dataset for the aim of increasing the system performance.

4.3 Building White Noise-Added Speech Dataset

The white noise-added speech dataset is built by using the following ***sox*** command to the original collected speech dataset to prove that noise hinders for enhancing the system performance. To add the white noise in original speech, the following ***sox*** command is used.

sox -m input.wav <(sox input.wav ï p synth whitenoise vol 0.02) output.wav

Contaminating to original dataset with white noise proves that the system performance can degrade by the disturbances of noise. Due to performance of this dataset, the future speaker recognition researchers will need to investigate to be noise robustness in system.

4.4 Statistics of Speech Datasets

The speech datasets used in this work comprise with 209 total speakers consisting of 89 male speakers and 120 female speakers with the age of ranging from 20 years to 70 years old. The size of original speech dataset is over 96 hours containing 58,014 utterances extending from 8 hours [P1, P2], and 57 hours [P3]. The scrutinized speech dataset is obtained by applying with the scrutinized methods. Its size extending from the dataset of 96 hours has the size over 120 hours with the same utterances as original dataset. The white noise-added dataset is created by using the original dataset. The detailed statistics of speech datasets and total number of utterances used for training, validation and test cases used in this work is described in Table 4.1.

Table 4.1 Statistics of Speech Datasets

Datasets	White noise (hh:mm:ss)	Original (hh:mm:ss)	Scrutinized (hh:mm:ss)	No. of Utterances
Training	86:12:26	85:53:13	107:02:16	51,976
Validation	07:37:28	07:17:37	09:05:28	4,256
OpenTestSet	03:38:48	03:28:42	04:20:12	1,782
ClosedTestSet	03:21:21	03:11:13	03:58:22	1,782
Total	97:28:42	96:39:32	120:27:56	58,014

4.5 Summary

This chapter presents building the Burmese speech dataset for using in speaker identification. It describes collecting, preparing and segmenting the speech data obtained from Web, preparing the transcriptions of daily conversational data for recording. The settings of recording platform are prepared in order to get the formatted speech segments. The scrutinized and white noise-added datasets are also created. The number of speakers and their age range contained in the speech dataset are represented and finally expressing the information of speech datasets used in this work.

CHAPTER 5
THE PROPOSED SYSTEM ARCHITECTURE

Speaker identification is the use of a machine to identify someone's identity based on a spoken phrase. It is way to make automatic recognizing of the person on the basis of specific information that comes from speech signals. There are many perspectives approaching to speaker identification system from the aspects of data point of view, and state-of-the-art technologies to enhance the system performance. This work emphasizes from the data point of view to boost the performance of the speaker models. The basic structure of the speaker recognition system, including verification and identification of speakers as well as their diarization, is shown in this chapter. It also presents the design and implementation processes of proposed system architecture together with clear understanding of pictorial representation.

5.1 Basic Structure of Speaker Recognition System

The three different tasks: speaker verification, speaker identification, and speaker diarization exist in speaker recognition. The detailed components contained in speaker recognition were explained in section 2.2 of Chapter 2. Speaker verification is the process of verifying a speaker's claimed identity based on their already registered voices. It verifies that a given speaker is one who claims to be and is one-to-one matching process. If it matches the set threshold, then the identity claim of the user is accepted otherwise rejected. Finally, the verification result (accept or reject) produces to the user outputs by the system. The process of speaker verification system is depicted in Figure 5.1.

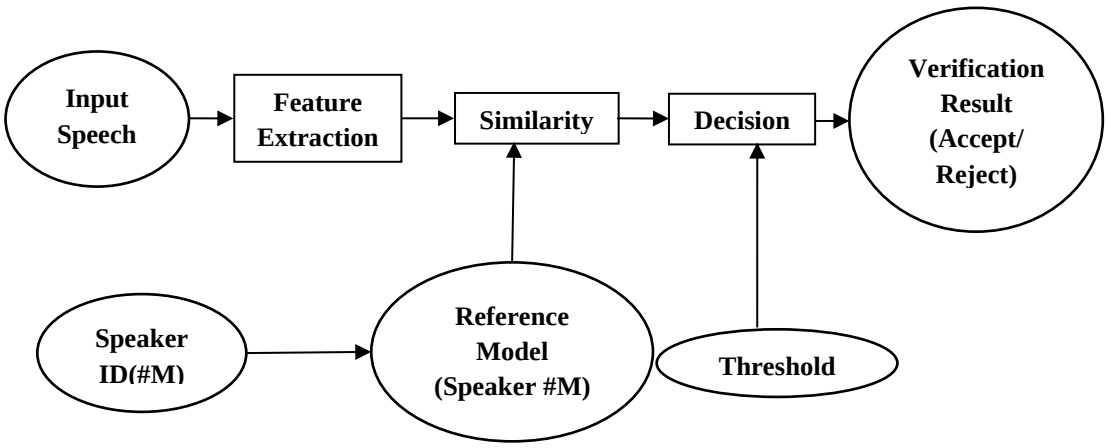


Figure 5.1 Basic Structure of Speaker Verification System

Speaker identification is a procedure that determines one's identity by machine. It detects a particular speaker from a known population by identifying whether a speaker's voice matches or not with any member of several registered voices. When the user provides the test speech utterance to the system, it identifies the best matched user by comparing the features of given speech utterance with those of the stored in the reference models ρĥϑ ĥü which contain the most likely speakers, could have given that speech utterance. Speaker identification performs one-to-many matching and finally gives the speaker identity that has a maximum score. The output speaker's identity is recognized as target speaker or impostor by determining with a predefined threshold in open set identification. If the output score is exceeded the threshold, the speaker who has this score is identified as the target speaker. If not, the speaker is identified as the impostor. The process of speaker identification system is depicted in Figure 5.2.

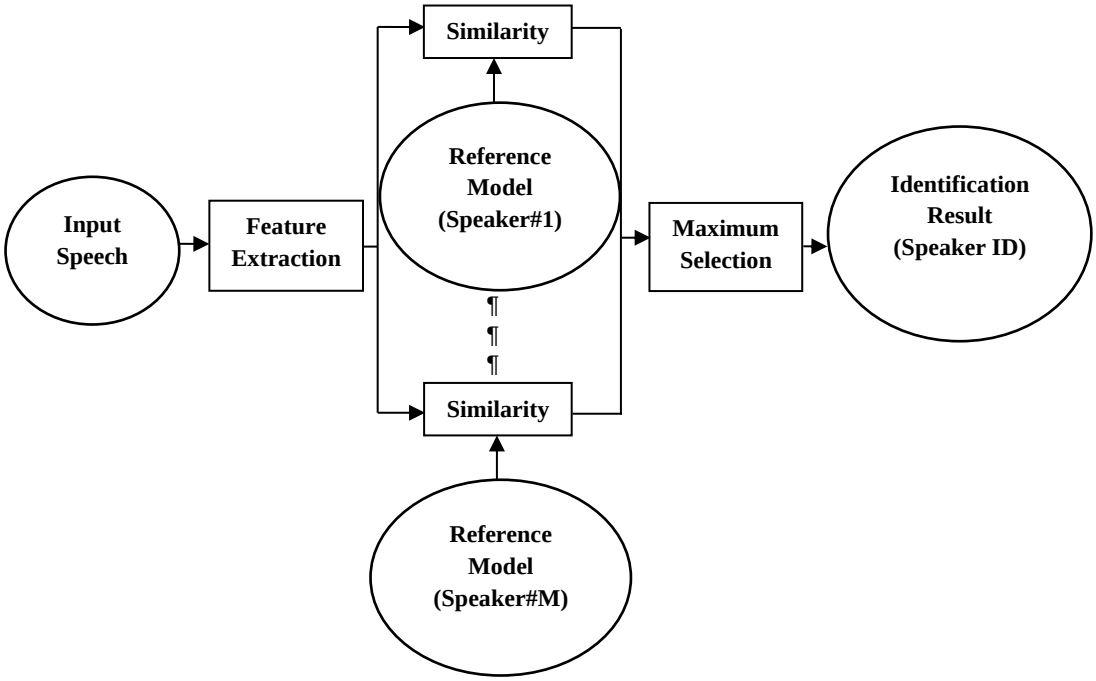


Figure 5.2 Basic Structure of Speaker Identification System

Speaker diarization is the process of determining "who spoke when?" in the whole audio conversation that contains the unknown amount of speech and number of speakers. It separates the speech utterances contained in the whole audio conversation into segments, labels the individual speakers corresponding to these segmented utterances correctly and identifies the different speakers from these segmented utterances. For clustering the speakers, the probabilistic approach like Gaussian

Mixture Model (GMM) is used to extract the feature vectors from speech and describe how many people are talking in the conversation. There are two types of speaker diarization namely online and offline speaker diarization. Online speaker diarization is also known as real time speaker diarization. Offline speaker diarization is based on the finished conversations like recordings and produces better outputs than online speaker diarization. The models for speaker diarization can be trained in two ways: supervised or unsupervised learning. In supervised approach, some or all of the individual speakers in parts of the stream conversation are already tagged and it leads to the lower error rate. But supervised training can only take place on offline recordings. Conversely, unsupervised learning leads to lose on a completed unlabeled conversation and increase the error rates because it does not know in advance who contain in the audio stream. The process of speaker diarization system is depicted in Figure 5.3.

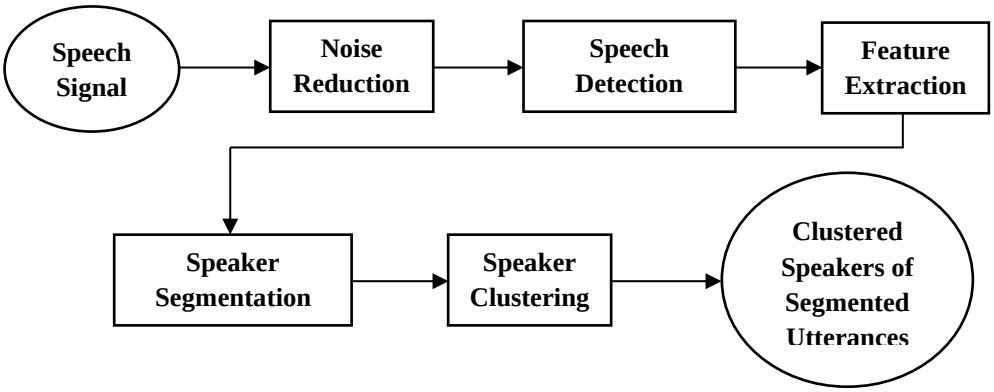


Figure 5.3 Basic Structure of Speaker Diarization System

5.2 Design and Implementation of Proposed System Architecture

The process of precisely identifying speakers through analysis of their speech features is known as speaker identification. Unlike speech recognition and pattern recognition problems, voice biometrics refers to the identification of an individual based on the characteristics of their voices. This section describes the design and implementation of proposed speaker identification system architecture with pictorial representation as shown in Figure 5.4.

As shown in Figure, there are two phases in speaker identification system: training and testing phase. There are also two parts: front-end and backend analysis in both phases. Data preprocessing, data augmentation and feature extraction are front-

end analysis. Constructing speaker models in training phase and identification process in testing phase are backend analysis.

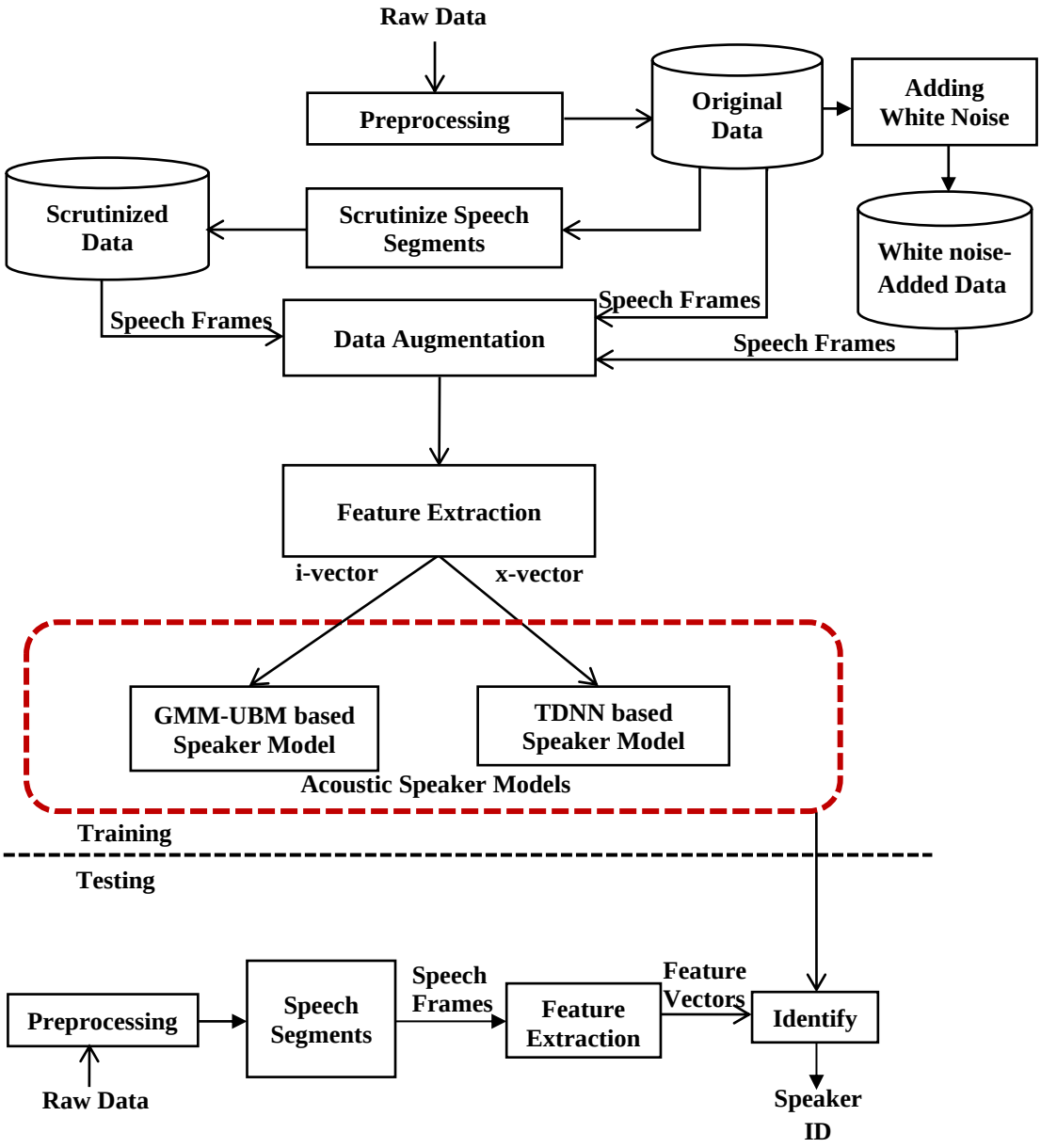


Figure 5.4 Proposed Architecture of Burmese Speaker Identification System

In data preprocessing, the original clean data are obtained from the raw data come from two main sources: Web-based data and recorded data with microphone and telephone. As part of Web-based data, the raw data are mainly taken from online sources in the .mp4 format. These .mp4 formatted files are converted to .wav form of

mono data in 16-bit PCM of 16kHz with the use of "***ffmpeg***" command by writing script in bash programming language. These converted .wav formatted data are segmented with ***Audacity*** tool to obtain the utterance level segments ranging from 3 seconds to 45 seconds depending on the spoken utterance duration.

For recorded data, daily conversational dialogue is recorded by internship students from three academic years and Lab members from Natural Language Processing Lab., University of Computer Studies, Yangon, Myanmar. The sentence level utterances which have at least 200 utterances each are recorded with the prepared setting with the format of 16-bit PCM of 16 kHz in mono data. At the end of data preprocessing, the original clean data long over 96 hours are obtained by combining Web data and recorded data.

Every segment of the original dataset is subjected to a tempo factor warp of 0.2 times (20%) and intensity amplified to 10 dB in order to get the scrutinized data. This makes enhancing the speech quality for effective performance, finding out the more speaker specific information and improving the system performance with reducing the error rate and raising the detecting accuracy. By analyzing the data, the scrutinized dataset with the size of 120 hours is obtained. As a consequence, slowing down the speaking rate (tempo factor down to 0.2 times) can cause the elongated duration on original utterance. Moreover, the original dataset is used to create the white noise-added dataset for proving that any kind of noise disturbances the system performance.

In the process of data augmentation, the training data in three types of datasets are individually contaminated to increase the diversity and volume of data. The data are augmented with noise and reverberation as an inexpensive method to multiply the quantity of training data for the aim of constructing effective speaker models. It can find out more speaker specific information and enhance the robustness of the speaker models in original and scrutinized datasets. Moreover, MUSAN's additive noises and simulated small Room Impulse Responses (RIRs) ranging the room area from 1 m to 10 m are artificially added to the training data. MUSAN dataset is freely downloaded from *OpenSLR* supported by *National Science Foundation Graduate Research Fellowship*. By augmenting with different techniques, the training data size increases in volume but only the amount equal to the original training data size is randomly taken as the augmented data. Therefore, the training data size of original, scrutinized

and white noise-added datasets gets doubling the size of the existing dataset when these augmented data are combined individually to each corresponding training data.

Feature extraction is also important in every speech related system. It converts the speech waveform into a set of feature vectors for further analysis by extracting the speaker specific information from the corresponding speech frames and investigates phonological characteristics that are resilient and discriminative improving recognition rate. This work uses 39-dimensional Mel Frequency Cepstral Coefficients (MFCCs) extracting every 10 ms in the frame size for 25 ms long by using the Hamming window. Delta feature is also used to improve accuracy and robustness in identification. Low level spectral feature like MFCC is easier to extract and more potent than high-level feature such as pitch involving more speaker related information. But high-level feature extraction process is more complicated and time consuming than low level features.

The acoustic speaker models based on GMM-UBM and TDNN are constructed as back-end analysis of speaker identification system. These are built based on three different training data: original, scrutinized and white noise-added datasets in training phase to prove that the benefits of scrutinizing methods and the drawbacks of the disturbance of noise. GMM-UBM is the parametric model which gives the probability distribution of feature vectors extracted from the different speakers and the background model is generated by using speech samples from all of these different speakers. The model corresponding to each of the individual speaker is obtained by adapting the parameters with the use of *Maximum A Posteriori (MAP)*. GMM parameters are trained by using Expectation-Maximization (EM) algorithm. In UBM training, a diagonal covariance model is firstly trained. Iterative training of EM is run with fixed means and mixture weights to obtain full covariance model. The speaker models are built by changing the parameters like the number of Gausian components and dimensions.

Time delay neural network (TDNN) uses a feed forward architecture being proven to be powerful in handling the context information of speech signal and modeling the phonetic content directly. The network processes the input from the narrow contexts to the speech signal in the first layer. The deeper layers will handle the input by splicing the output of the hidden activations from the preceding layer in order to learn wider temporal dependencies. Subsampling technique is applied to save time consuming in model training with the advantages of decreasing the numbers of

parameters and increasing the computational efficiency. In order to teach frame-level speaker features that will appropriately convey the data to the model, this approach employs five time delay layers. The speaker models are built by using different slicing parameters to investigate which network context impacts the model efficiency.

After the acoustic speaker models have been constructed, the recognizing accuracy of the speaker models is assessed in the testing phase. The input test speech is firstly preprocessed and then extracted the MFCC feature corresponding to the test utterance by using the same processes as training phase. For recognizing the speaker identity, the test speech feature vector is classified with the speaker models constructed in the training phase by computing the log likelihood score for the test speech whether the two vectors (target and test) are or are not generated by the same speaker. Probabilistic linear discriminant analysis (PLDA) is used for scoring in speaker identification based on feature vectors.

5.3 Summary

This chapter presents what is speaker identification system, the basic structures of speaker recognition system: speaker verification, speaker identification and speaker diarization with pictorial representations. The design and implementation of proposed architecture of this work are explained in detail together with clear understanding of pictorial representation.

CHAPTER 6
PERFORMANCE ANALYSIS FOR SPEAKER IDENTIFICATION

This chapter presents the experimental setup regarding with building acoustic models, the promising results derived from assessing the performance of the acoustic speaker models and showing the improvement of recognizing quality by scrutinized dataset comparing with original and white noise-added datasets on Burmese speaker identification. Building or training the acoustic models is one of the important phases of backend analysis in speaker recognition system employed after the feature extraction step. This is the process of making the system know the speakers and deals with collecting data from the utterances of people to be identified. Only the speaker models are constructed, the performance of speaker identification system can be assessed by comparing the incoming test sample with these speaker models. If the good speaker model is built, the rest of the processes in speaker identification become extremely easy. The evaluation of automatic speaker identification performance is done according to changing the number of Gaussian components and feature vectors' dimensions on GMM-UBM and tuning the network input contexts parameters on TDNN with different datasets.

6.1 Building Speaker Models

In the world of science, human mimics always understood by computer. The idea for making speaker identification system is to be convenient for humans to interact with the computer by speech or vocalization rather than other instructions. Individualized voice recognition alone is a feature that is still relatively unfinished. Accents and dialects are vast and varied, making it a continual challenge to perfect the technology. There are many speaker identification research carrying out by many researchers for their respective languages and the performance of these systems has been enhanced by investigating the particular properties of the target language and applying the new architecture.

The speaker models are constructed by using the corresponding extracted feature vectors and are stored these models in the training database with corresponding speaker ID which is unique. The system recognizes whether the incoming test sample is from the same speaker the acoustic models were trained on.

The two types of training models that can be used for automatic speaker identification are parametric and nonparametric models.

6.1.1 Parametric Models

These models have a particular structure characterized by a set of parameters. By defining the structure, the form of the model has been specified and limited to a specific requirement. This ensures that it makes an efficient use of the data in estimating the model parameters. The other advantage in using parametric model is that the changes in the parameters can be easily determined by the changes in the data [52]. Literature shows that many researchers have implemented parametric models in the text independent speaker recognition system [32, 52-55]. Some of parametric models are Gaussian Mixture Models (GMM), Hidden Markov Model (HMM), and Neural Networks (NN). Text independent speaker recognition with Gaussian Mixture Model was proposed by Reynolds [32]. GMM is most commonly used parametric model for training purposes [52] and the implementation of Neural Networks was proposed by Seddik, Rahmouni and Sayadi in [55]. These two types of parametric models are used to increase robustness and performance of the designed approach in this work.

6.1.2 Nonparametric Models

Nonparametric models differ from the parametric models like the way in which the space is dichotomized. Only the minimal assumptions regarding the probability density functions are made. Dynamic Time Warping (DTW) is an example for nonparametric models. Vector Quantization is used for text independent speaker recognition whereas Dynamic Time Warping is used for text dependent speaker recognition. Vector Quantization was first applied to speaker recognition by [56]. A description and a comparison of VQ model with HMM for text independent speaker recognition system is given by Matsui and Furui [57].

6.2 Building Acoustic Speaker Models

Acoustic modeling is an essential part of automatic speaker identification system. The acoustic models are built on Gaussian Mixture Model-Universal Background Model (GMM-UBM) and Time Delay Neural Network (TDNN) in this

work. If the efficient acoustic modeling approach can be applied, the higher recognition rate can be achieved. The conventional GMM-UBM also shows the significant improvements while Time Delay Neural Network (TDNN) has given the state-of-the-art performance results. TDNN can model the context information at each layer of network and capture long term temporal dependencies. Two different acoustic models (GMM-UBM and TDNN) are implemented to show that changing the number of Gaussian components and dimensions on GMM-UBM and tuning the network input contexts on TDNN layers can affect to the speaker models' performance. And the experiments are implemented on three different training data: original, scrutinized and white noise-added data and show that the quality of data can also affect the recognizing rate.

6.2.1 Experimental Setup

The experimental setup for speech dataset used in this work is deal with in this section in detail. The speech utterances are collected from Web-based news: broadcasting news, delivered speech and talks and daily conversational dialogues were also recorded ourselves. Because the speakers in the data gathered from Web resources are knowledgeable and skilled, it is accurate and has a clear tone. Web-based speech collection from 111 speakers includes both local and international news about politics, health, sports, education, crime, business news, and weather, among other topics. In data preparation, the wave files in the datasets are already formatted with a frequency rate of 16 kHz in 16-bits, mono PCM. The recorded data with microphone and telephone are collected with 31 Lab members and 36 internships students and 31 persons from others. The length of each daily conversational dialogue spoken in restaurants, hotels, parks and traveling is shorter than the Web-based news. The detailed statistics of the training data and test sets are described sharply in Table 6.1.

6.2.2 GMM-UBM based Acoustic Model

This section presents building the acoustic model based on GMM-UBM, the probability distribution model. It is considered as a generative model and focused on representing the total distribution of the speaker data. The parameters are estimated with Maximum likelihood or Maximum A-Posteriori criteria. Competition with other

models comes through likelihood ratio. The probabilistic models like GMM yield better performance results for training both text dependent and text independent speaker recognition applications [32]. Due to the probabilistic property of a GMM, it can also be applied to speaker recognition applications in the presence of different noises increasing the channel robustness. Mixtures of Gaussians can be used to more correctly mimic the acoustic fluctuations caused by environmental noise, accent, pronunciation variation, speaker factor and other factors.

Table 6.1 Detail Statistics of Speech Datasets

<table>
<tr><th colspan="2">Category</th><th colspan="3">Description</th></tr>
<tr><td colspan="2">Collecting speech</td><td colspan="3">News, Talks, Delivered Speech, Daily Conversational Data Recorded with Telephone and Microphone</td></tr>
<tr><td colspan="2">Sampling</td><td colspan="3">16 kHz, 16-bits, mono PCM</td></tr>
<tr><td colspan="2">Utterance’s length</td><td colspan="3">Ranging from 3 seconds to 45 seconds each</td></tr>
<tr><td colspan="2">Dataset Size</td><td>97 hrs (white noise)</td><td>96 hrs (original)</td><td>120 hrs (scrutinized)</td></tr>
<tr><td rowspan="2">Training</td><td>hr:min:sec</td><td>86:12:26</td><td>85:53:13</td><td>107:02:16</td></tr>
<tr><td>Utts.</td><td colspan="3">51,976</td></tr>
<tr><td rowspan="2">Validation</td><td>hr:min:sec</td><td>07:37:28</td><td>07:17:37</td><td>09:05:28</td></tr>
<tr><td>Utts.</td><td colspan="3">4,256</td></tr>
<tr><td rowspan="2">OpenTestSet</td><td>hr:min:sec</td><td>03:38:48</td><td>03:28:42</td><td>04:20:12</td></tr>
<tr><td>Utts.</td><td colspan="3">1,782</td></tr>
<tr><td rowspan="2">ClosedTestSet</td><td>hr:min:sec</td><td>03:21:21</td><td>03:11:13</td><td>03:58:22</td></tr>
<tr><td>Utts.</td><td colspan="3">1,782</td></tr>
<tr><td rowspan="2">Speakers
(20 ~ 70 yrs.)</td><td>Male</td><td colspan="2">89</td><td rowspan="2">209</td></tr>
<tr><td>Female</td><td colspan="2">120</td></tr>
</table>

Because Gaussian components can express some general speaker dependent spectral forms, Gaussian mixture modeling (GMM) is a classic parametric method

that works best for modeling speaker identities. Since the Gaussian classifier uses a similar technique to the long-term average of spectral data for describing a speaker's average vocal tract shape, it has been successfully used in various text independent speaker identification applications [58].

6.2.2.1 Evaluation with the Number of Gaussian Components and i-Vector Dimensions

Gaussian Mixture Model (GMM) is a probabilistic model which is signified as a biased amount of Gaussian element densities and used to model the distribution of the acoustic characteristics of speech. Universal background model (UBM) is the GMM trained on a large background set in which speaker and channel variability are adequately represented. The number of Gaussian components and i-vector dimensions used in building GMM-UBM based acoustic models can reduce the error rate and improve the recognizing rate. Experiments are implemented to analyze the best parameters of Gaussian components and i-vector dimensions to enhance the system performance. Table 6.2 shows the different parameters used in building GMM-UBM based acoustic models.

Table 6.2 Parameters used in GMM-UBM

GMM-UBM Models	Number of Gaussian Components	Number of i-Vector Dimensions
$Model_{I}$	400	200
$Model_{II}$	300	150
$Model_{III}$	250	125
$Model_{IV}$	200	100
$Model_{V}$	100	50

6.2.2.2 Experimental Results

The evaluation results on different speaker models of GMM-UBM with data augmentation and without data augmentation methods are represented in this section to prove that data augmentation methods can also enhance the system performance. Table 6.3 and Figure 6.1 represents the equal error rate (EER) of different GMM-

UBM based speaker models with (withAug) and without (noAug) data augmentation methods. According to the outcomes as shown in Table 6.3 and Figure 6.1, increasing the number of Gaussian Components and i-vector dimensions reduce the error rate and lead to better detecting accuracy together with and without data augmentation methods.

Table 6.3 EER (%) on GMM-UBM based Acoustic Models with and without Data Augmentation

GMM-UBM Models	Equal Error Rate (%)					
	white noise		original		scrutinized	
	noAug	withAug	noAug	withAug	noAug	withAug
$Model_I$	1.433	1.316	0.6347	0.6344	0.6109	***0.5639***
$Model_{II}$	1.598	1.551	0.7284	0.6579	0.6344	0.6109
$Model_{III}$	1.715	1.48	0.7989	0.7281	0.7804	0.7049
$Model_{IV}$	2.209	1.668	0.9868	0.8224	0.8929	0.7754
$Model_V$	4.018	2.961	2.138	1.809	1.903	1.527

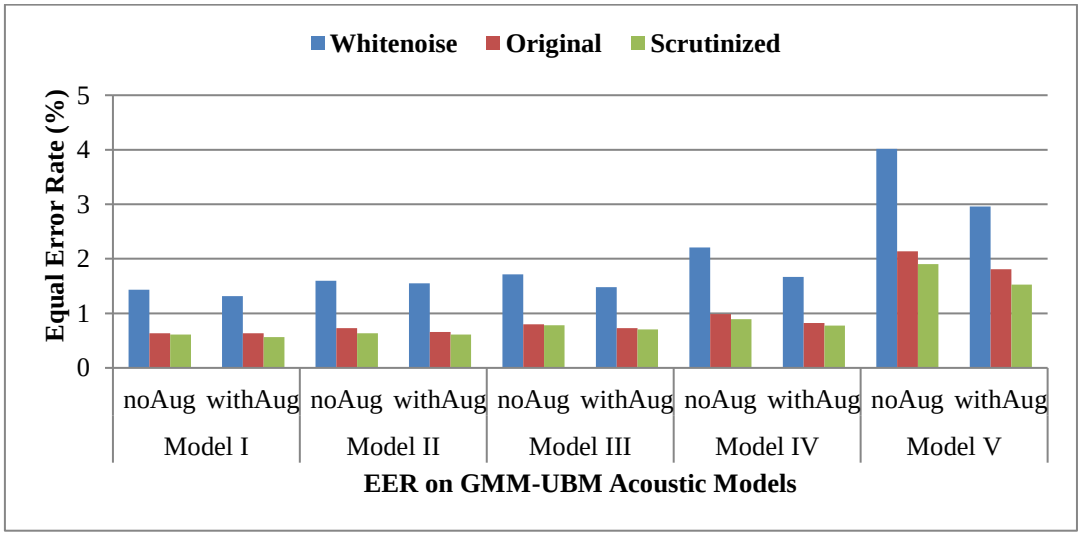


Figure 6.1 EER (%) on GMM-UBM based Acoustic Models with and without Data Augmentation

Table 6.4 and Figure 6.2 represents the detecting accuracy in percentage (recognizing rate of speaker models) of different GMM-UBM based speaker models

with (withAug) and without (noAug) data augmentation methods in open test set evaluation.

Table 6.4 Detecting Accuracy (%) of GMM-UBM based Acoustic Models with and without Data Augmentation on OpenTestSet

GMM-UBM Models	Detecting Accuracy (%)					
	white noise		original		scrutinized	
	noAug	withAug	noAug	withAug	noAug	withAug
$Model_{I}$	90.00	92.98	97.08	97.41	97.69	***98.03***
$Model_{II}$	88.37	91.18	96.29	97.3	96.57	97.52
$Model_{III}$	88.69	90.00	95.84	96.51	96.29	96.74
$Model_{IV}$	84.83	89.33	94.21	95.95	95.84	96.29
$Model_{V}$	77.82	80.57	87.98	90.9	90.34	91.85

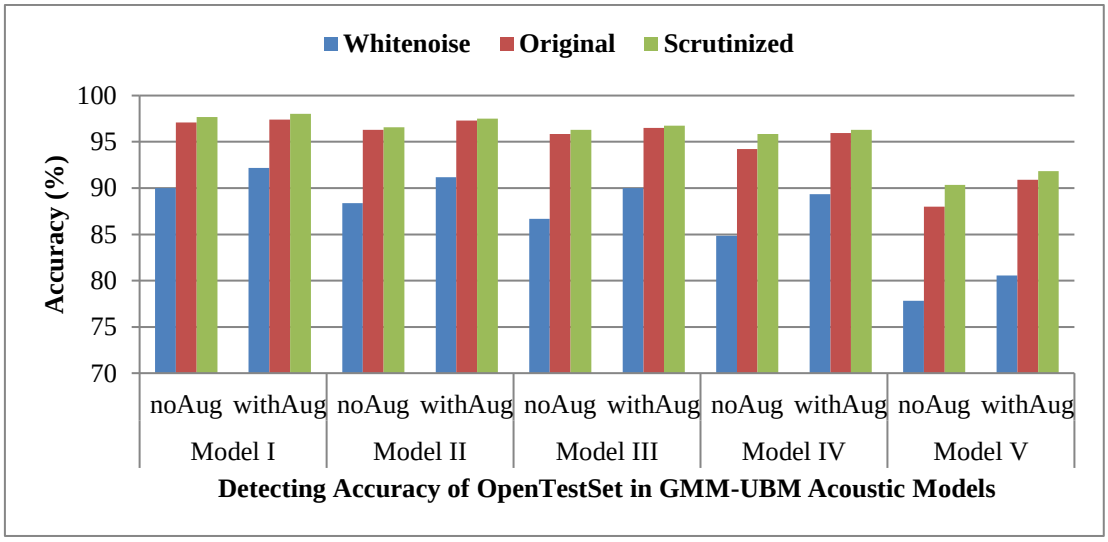


Figure 6.2 Detecting Accuracy (%) of GMM-UBM based Acoustic Models with and without Data Augmentation on OpenTestSet

Table 6.5 and Figure 6.3 represent the detecting accuracy in percentage (recognizing rate of speaker models) of different GMM-UBM based speaker models with (withAug) and without (noAug) data augmentation methods in closed test set evaluation.

Table 6.5 Detecting Accuracy (%) of GMM-UBM based Acoustic Models with and without Data Augmentation on ClosedTestSet

GMM-UBM Models	Detecting Accuracy (%)					
	white noise		original		scrutinized	
	noAug	withAug	noAug	withAug	noAug	withAug
$Model_{I}$	91.35	92.98	97.36	97.86	98.61	***98.81***
$Model_{II}$	89.89	92.41	97.08	97.52	97.13	97.69
$Model_{III}$	89.33	91.01	96.06	96.68	97.02	97.3
$Model_{IV}$	88	88.04	95.33	96.79	96.01	97.13
$Model_{V}$	78.6	82.42	89.21	91.46	90.67	92.19

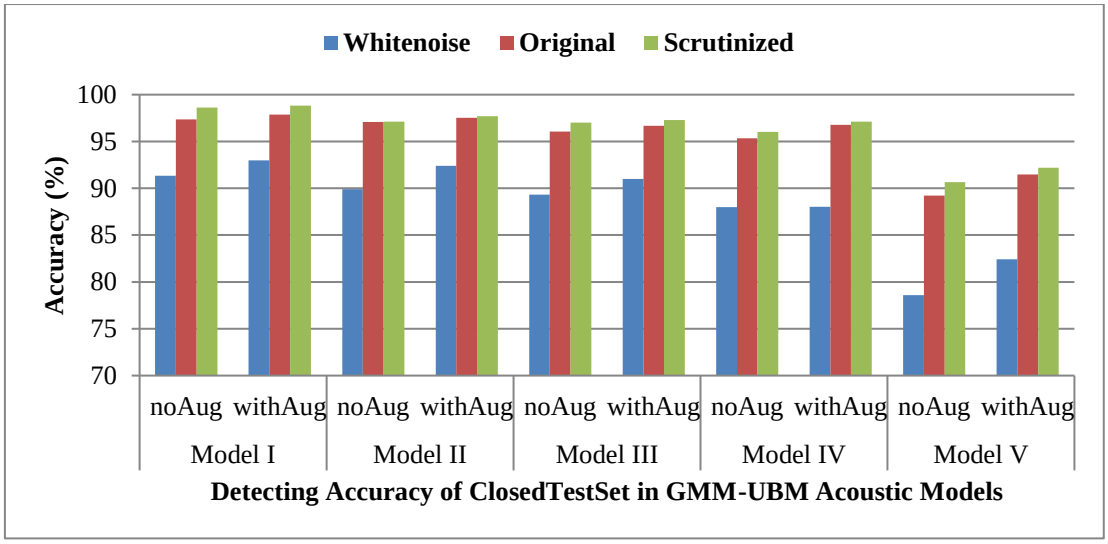


Figure 6.3 Detecting Accuracy (%) of GMM-UBM based Acoustic Models with and without Data Augmentation on ClosedTestSet

6.2.3 Time Delay Neural Network based Acoustic Model

Neural Networks require many speech data in convergence of model training. As a result, it consumes a lot of time in model training. This work explores the ways to improve the modeling capabilities of Time delay neural network (TDNN). It is designed to express a relation among inputs in time and utilized to model long-term dependencies. It also has strong probability in context modeling. TDNN consists of frame level layers, statistics pooling that estimates mean and standard deviation over input frame features, segment level layers and a final softmax layer where outputs correspond to speaker probabilities explained in detail at Chapter 3.

6.2.3.1 Evaluation with Different Network Contexts

TDNN architecture is tied across time steps to decrease parameters and learn shift invariant feature transforms and to increase the computational efficiency. Splicing continuous windows of frames into typical TDNNs results in overlap and redundancy, which makes model training time-consuming. In order to overcome the performance of the original data, this study employs the subsampling strategy to increase efficiency while permitting the cracks between feature frames at each layer and to shorten the training time complexity of the model. Table 6.6 shows the different parameters tuning of layer-wise network context with subsampling technique for use in building TDNN based acoustic models. In this table, the splicing configuration {-2, 2} means that splice the input at present period subtract 2 and the current time step add 2.

Table 6.6 Layer-wise Context Parameter Tuning Settings of TDNN

TDNN Models	Network Context	Layer-wise Context				
		1	2	3	4	5
$TDNN_{I}$	[-7, 7]	[-2, 2]	{-2, 2}	{-3, 3}	{0}	{0}
$TDNN_{II}$	[-8, 8]	[-2, 2]	{-1, 1}	{-2, 2}	{-3, 3}	{0}
$TDNN_{III}$	[-9, 7]	[-2, 2]	{-2, 2}	{-5, 3}	{0}	{0}
$TDNN_{IV}$	[-10, 6]	[-2, 2]	{-2, 2}	{-6, 2}	{0}	{0}
$TDNN_{V}$	[-10, 8]	[-2, 2]	{-1, 1}	{-2, 2}	{-5, 3}	{0}
$TDNN_{VI}$	[-11, 6]	[-2, 2]	{-2, 2}	{-7, 2}	{0}	{0}
$TDNN_{VII}$	[-11, 7]	[-2, 2]	{-1, 1}	{-2, 2}	{-6, 2}	{0}
$TDNN_{VIII}$	[-12, 7]	[-2, 2]	{-1, 1}	{-2, 2}	{-7, 2}	{0}

6.2.3.2 Experimental Results

The experimental results on different speaker models of TDNN with data augmentation and without data augmentation methods are represented in this section to show that data augmentation methods can also upgrade the system performance. Table 6.7 and Figure 6.4 represents the equal error rate (EER) of different TDNN based speaker models with (withAug) and without (noAug) data augmentation

methods. According to the results as shown in Table 6.7 and Figure 6.4, the slicing parameters for 5-time delay layers: {t-2, t-1, t, t+1, t+2}, {t-1, t, t+1}, {t-2, t, t+2}, {t-3, t, t+3}, {t} give the optimal result in EER and lead to better detecting accuracy together with and without data augmentation methods.

Table 6.7 EER (%) on TDNN based Acoustic Models using Subsampling Technique with and without Data Augmentation

TDNN Models	**Equal Error Rate (%)**					
	white noise		original		scrutinized	
	noAug	withAug	noAug	withAug	noAug	withAug
$TDNN_{I}$	2.198	1.025	1.292	0.9164	1.245	0.8459
$TDNN_{II}$	2.151	1.626	1.222	0.8929	1.198	***0.7989***
$TDNN_{III}$	2.292	1.955	1.339	0.9398	1.316	0.8134
$TDNN_{IV}$	2.339	1.931	1.392	0.9164	1.292	0.8271
$TDNN_{V}$	2.222	1.696	1.269	0.9263	1.241	0.8929
$TDNN_{VI}$	2.292	1.649	1.363	0.9868	1.316	0.8224
$TDNN_{VII}$	2.198	1.767	1.682	0.9633	1.269	0.8929
$TDNN_{VIII}$	2.175	1.814	1.523	0.9868	1.245	0.9164

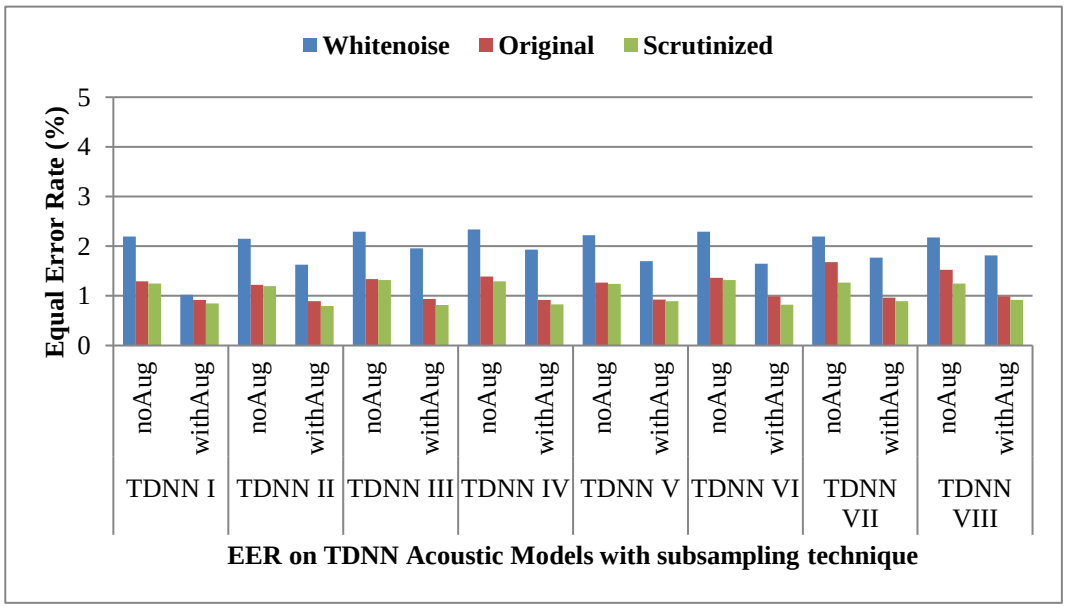


Figure 6.4 EER (%) on TDNN based Acoustic Models using Subsampling Technique with and without Data Augmentation

Table 6.8 and Figure 6.5 represent the detecting accuracy in percentage (recognizing rate of speaker models) of different TDNN based speaker models with (withAug) and without (noAug) data augmentation methods in open test set evaluation by using subsampling technique.

Table 6.8 Detecting Accuracy (%) of TDNN based Acoustic Models with and without Data Augmentation on OpenTestSet using Subsampling Technique

TDNN Models	**Detecting Accuracy (%)**					
	white noise		original		scrutinized	
	noAug	withAug	noAug	withAug	noAug	withAug
$TDNN_{I}$	87.59	95.95	96.06	98.25	96.4	98.54
$TDNN_{II}$	87.59	95.9	96.91	98.03	97.13	***98.88***
$TDNN_{III}$	87.53	94.6	96.29	97.92	96.63	98.48
$TDNN_{IV}$	86.35	95.28	96.7	97.97	96.35	98.59
$TDNN_{V}$	87.7	95.56	95.9	97.75	96.85	98.37
$TDNN_{VI}$	85.73	95.5	96.35	97.69	96.85	98.59
$TDNN_{VII}$	88.04	95.5	96.01	97.86	96.57	98.7
$TDNN_{VIII}$	88.15	94.38	96.57	97.97	96.63	98.59

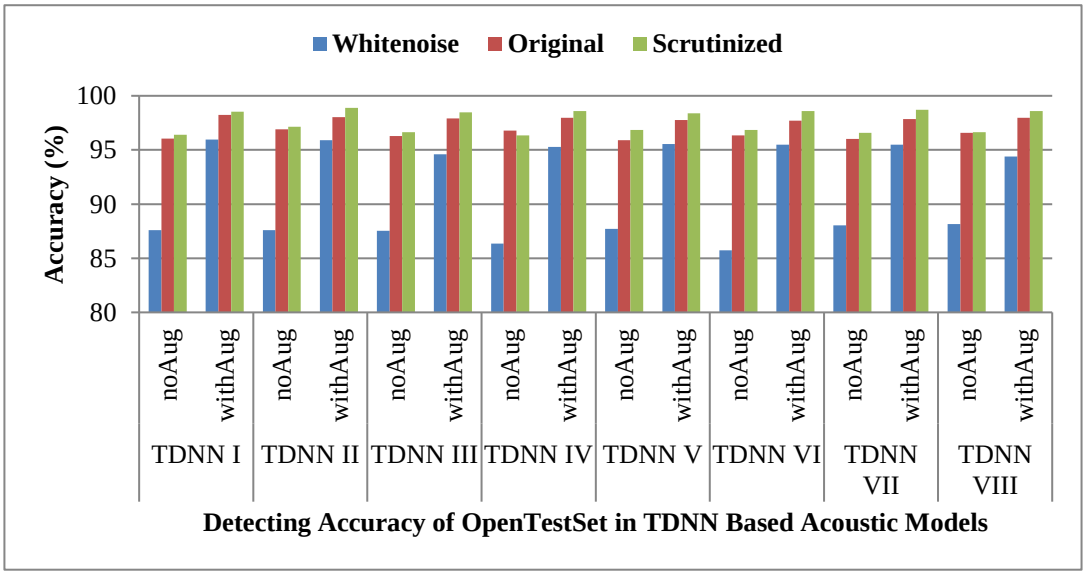


Figure 6.5 Detecting Accuracy (%) of TDNN based Acoustic Models with and without Data Augmentation on OpenTestSet using Subsampling Technique

Table 6.9 and Figure 6.6 represent the detecting accuracy in percentage (recognizing rate of speaker models) of different TDNN based speaker models with (withAug) and without (noAug) data augmentation methods in closed test set evaluation by using subsampling technique.

Table 6.9 Detecting Accuracy (%) of TDNN based Acoustic Models with and without Data Augmentation on ClosedTestSet using Subsampling Technique

TDNN Models	Detecting Accuracy (%)					
	white noise		original		scrutinized	
	noAug	withAug	noAug	withAug	noAug	withAug
$TDNN_{I}$	92.08	97.02	96.51	98.78	97.08	98.82
$TDNN_{II}$	92.08	96.12	97.02	98.59	97.86	***98.91***
$TDNN_{III}$	90.51	95.84	96.96	98.42	97.52	98.61
$TDNN_{IV}$	90.17	96.46	96.86	98.48	97.02	98.82
$TDNN_{V}$	92.19	96.57	97.02	98.37	97.24	98.48
$TDNN_{VI}$	89.55	96.12	96.57	98.69	97.52	98.82
$TDNN_{VII}$	91.8	96.73	97.13	98.59	97.47	98.87
$TDNN_{VIII}$	92.53	96.06	97.02	98.42	97.47	98.72

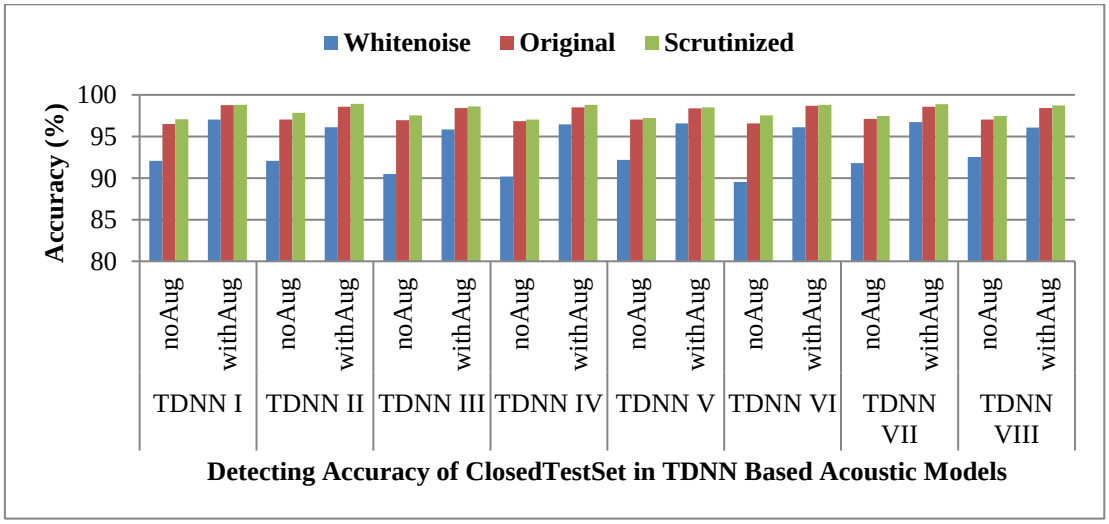


Figure 6.6 Detecting Accuracy (%) of TDNN based Acoustic Models with and without Data Augmentation on ClosedTestSet using Subsampling Technique

6.3 Discussion on Different Acoustic Models and Test Cases

Based on the experimental results of three acoustic models on different training datasets, some analysis and discussion are presented in this section. According to the aspects of training datasets, adding white noise to the original data does not give the effectiveness in building the acoustic models. It can cause not only hindering the clarity of sound upon each speech utterances but also degrading the recognizing performance without knowing clearly who the speakers are. However, the dataset created with proposed scrutinized methods leads to better the recognizing performance than the original dataset. Moreover, the pace and clarity of sound to all utterances are smoothed and has clear understanding of what they are saying. It is evident that the suggested data inspection method produces findings that are equivalent for each speaker type, each of which has a lower error rate. This may also give the benefits to the automatic speech recognition.

By discussing from the aspects of data augmentation, it makes the training powerful and increases the diversity of training data having doubling the size of original dataset. If the experimental results of speaker models using data augmentation techniques compares with the experimental results of speaker models without data augmentation techniques, the models comprising with data augmentation techniques give better results than that of comprising without data augmentation techniques. Therefore, the benefits not only getting the variety of training data but also getting system robustness obtain from augmenting the training data.

On the other hand of the aspects of speaker models, the different parameter tunings used in acoustic models' trainings give the effectiveness of not only reducing the error rate but only increasing the identifying rate. The experimental results of speaker models using data augmentation are discussed and analyzed in this section.

In GMM-UBM based acoustic models using data augmentation, 400 Gaussian components and 200 i-vector dimensions reduce the identifying error rate and increase the detecting accuracy of every speaker's identity in both test cases (OpenTestSet and ClosedTestSet). This is because it is a classical parametric method expressing the speaker related data in total distribution with the capability of representing the spectral shapes in Gaussian components in which speaker and channel variability are adequately represented. Therefore, building the speaker

acoustic models with 400 Gaussian components and 200 i-vector dimensions upgrade the system performance leading to reducing the error rate and enhancing the accuracy on all of training datasets.

According to the experiments, increasing the number of Gaussian components and i-vector dimensions causes the speaker models leading to better performance. By comparing the relative improvements between original and scrutinized datasets, the speaker models using data augmentation constructed with 400 Gaussian components and 200 i-vector dimensions give the best relative improvement up to over 12 % although the improvements on each respective model increase. The results of equal error rate of speaker models with 100 Gaussian components and 50 i-vector dimensions as shown in Table 6.3 are the highest on all of speaker models although these yield the highest relative improvement among different parameter tunings.

In TDNN based acoustic models using data augmentation, the speaker models with [-8, 8] layer-wise network input context decreases the identifying error rate and improves the detecting accuracy of every speaker's identity in both test cases (OpenTestSet and ClosedTestSet). This is because the information from relatively narrow contexts is processed by the initial transforms of a TDNN and the hidden activations from a wider context are processed by the deeper layers. In a typical TDNN, hidden activations are computed at all-time steps in which large overlap between input contexts computed at adjacent time frames will cause redundant computation. Minimum overlaps between input contexts can be achieved through the carefully designed hierarchical structure. Instead of simply splicing together continuous temporal window of frames at each layer of network, subsampling technique is used in this work. It allows gaps between the frames in a window and thus substantially reduces the overall computation. And this process helps save computation, reduce the model size and speed up the training time significantly.

According to the experiments, the network contexts of [t-7, t+7], [t-9, t+7], [t-10, t+6] and [t-11, t+6] containing two fully connected layers in upper layers 4 and 5 obviously decreases the error rate value with relative improvement up to over 19 % when comparing the results of the original data with scrutinized data. The network contexts of [t-8, t+8], [t-10, t+8], [t-11, t+7] and [t-12, t+7] which contains only one fully connected layer before producing the output decrease the error rate with relative

improvement over 7 % in [t-11, t+7], [t-12, t+7] and only nearly 4 % in [t-10, t+8]. This is because the hidden layer 4 of [t-10, t+8], [t-11, t+7] and [t-12, t+7] takes the time step of the upper and lower bound very far causing loss the specific information of speakers. Among them, the reason of why the network context of [t-8, t+8] as the optimal network context is chosen is that it reduces the EER not only in scrutinized dataset but also in original dataset obviously. Moreover, as the input, the network context in layer 4 takes the adjacent frames on narrow temporal contexts from the lower layer 3. Therefore, building the speaker acoustic models with [-8, 8] layer-wise network input context enhances the system performance leading to reducing the error rate and increasing the accuracy on both test cases. The error rate of network context [-8, 8] scrutinized dataset is the best rate in comparing with the results of other acoustic models. It can be obviously seen that the error rate of scrutinized dataset decreases on every speaker model whether with or without augmentation methods but the error rate of [-8, 8] context of scrutinized dataset is the best rate in comparing with the results of other network contexts.

In comparing the detecting accuracy of both testsets (OpenTestSet and ClosedTestSet), the accuracy of ClosedTestSet is higher than that of OpenTestSet on all of speaker models. This is because there are no spoken utterances of OpenTestSet in training datasets although the spoken utterances of ClosedTestSet are already contained in training datasets. It leads this dissertation to text independent Burmese speaker identification in open-set case.

6.4 Summary

This chapter presents the experimental setup regarding with building acoustic models, the discussion and analysis of promising results derived from assessing the performance of the acoustic speaker models and showing the improvement of recognizing quality by scrutinized dataset comparing with original and white noise-added dataset on Burmese speaker identification.

CHAPTER 7
CONCLUSION AND FUTURE WORKS

Summarization of the dissertation, its advantages and limitations of proposed system are described and future works will be discussed in this chapter.

7.1 Dissertation Summary

There are many state-of-the-art technologies applied to automatic speaker recognition. It can be done by traditional GMM-UBM and more advanced by Neural Networks especially TDNN. From the aspects of training data, data augmenting and scrutinizing methods are investigated and analyzed for more performance in this dissertation.

The objective of this research is to build the Burmese speech dataset first for applying to speaker identification, to augment with MUSAN dataset for sufficient amount of training data and to investigate the speech quality by using scrutinized techniques for the capability of identifying the speakers. Over 96 hours of speech dataset are created as one of main contributions. The scrutinized speech dataset (over 120 hours) is built for better performance by using these over 96 hours of original collected speech data because getting and building high quality speech dataset is important for speaker recognition research especially in low-resourced tonal language like Burmese. It is created by using two data scrutinizing methods: increasing the speech intensity in SNRs to 10 dB and downing the tempo factor 0.2 times without affecting the pitch of utterances. The dataset with white noise-added is also created for highlighting any kind of noise hiders the performance of systems. Performance has been assessed with GMM-UBM and TDNN recognizers modeled using MFCC feature to prove the robustness of scrutinized training dataset and the weakness of white noise-added training datasets comparing with the original dataset. Moreover, the effectiveness of parameters like number of components, and i-vector dimensions in GMM-UBM and layer-wise context parameter tuning in TDNN used for building the acoustic speaker models was analyzed on corresponding different training models.

This dissertation summarizes how to collect the speech data, the effectiveness of scrutinized data for capability of identifying the speakers and the impacts of

reducing error rate and increasing detecting accuracy on various speaker models built with different training data. The exploration of TDNN is the first work applying in speaker identification especially for Burmese language.

7.2 Advantages and Limitations

Speaker identification is still becoming as the increased attention from researchers in the domain of speech processing concerning information security for many years. According to the research findings and experimental outcomes, the proposed augmenting and scrutinizing methods enhance the system performance. Scrutinizing speech datasets will be very beneficial for future research on speaker identification and will allow for the addition of more speech data. This can help to provide the safety access to call centers in financial sectors, some access control in the industrial sector and to improve in biometrics applications like fingerprint, face, iris, palm and vein, and heartbeat. Scrutinizing data helps to obtain both numerous and high-quality data for low-resourced languages like Burmese. Because the stability of the identification process is insufficient, improved recognition also depends on the duration, speech frequency range, recording environment, accent, and physical state of the speakers. As a drawback of this work, the recording environment and type of headphone is important for testing in real time. If the quality of headphone is not good and the recording environment is noisy, the incoming test utterance cannot produce the speaker identity correctly.

7.3 Future Works

The results show that, with corresponding relative improvements, the performance of models using the scrutinized speech dataset beats that of models using the original collected speech dataset in acoustic speaker modeling. By further optimization, these examining techniques will be used to other research, such as the study of speaker identification. The hybrid system of automatic speech recognition and speaker recognition will be implemented as extensions of the identified words what the speaker says. End to end learning approach will be pursued in speech recognition as future work for more improving the performance of speaker identification.

LIST OF ACRONYMS

Acc	Detecting Accuracy
ASR	Automatic Speech Recognition
BBC	British Broadcasting Corporation
CER	Crossover Error Rate
dB	Decibel
DCT	Discrete Cosine Transform
DET	Detection Error Tradeoff
DFT	Discrete Fourier Transform
DNNs	Deep Neural Networks
DVB	Democratic Voice of Burma
EER	Equal Error Rate
EM	Expectation-Maximization
FA	False Acceptance
FAR	False Acceptance Rate
FFT	Fast Fourier Transform
FR	False Rejection
FRR	False Rejection Rate
GMMs	Gaussian Mixture Models
GMM-UBM	Gaussian Mixture Model-Universal Background Model
GPUs	Global Processing Units
HCI	Human-Computer Interaction
HMMs	Hidden Markov Models
HZCRR	High Zero Crossing Rate Ratio
JFA	Joint Factor Analysis

kHz	Kilo Hertz
LDA	Linear Discriminant Analysis
LSTER	Low Short Time Energy Ratio
LVCSR	*Large Vocabulary Continuous Speech Recognition*
MAP	Maximum A Posteriori
MFCC	Mel Frequency Cepstral Coefficient
minDCFs	Minimum Detection Cost Functions
MLP	Multi-Layer Perceptron
MRTV	Myanmar Radio and Television
MUSAN	Music, Speech and Noise Corpus
NIST	National Institute of Standards and Technology
NTT	Nippon Telegraph and Telephone
OpenSLR	Open Speech and Language Resources
PCM	Pulse Code Modulation
pdf	Probability Density Function
PLDA	Probabilistic Linear Discriminant Analysis
PLP	Perceptual Linear Prediction
ReLUs	Rectified Linear Units
RFA	Radio Free Asia
RIRs	Room Impulse Responses
RVB	reverberation
RWCP-SSD	Real World Computing Partnership-Sound Scene Database
SNRs	Signal to Noise Ratios
SP	speed perturbation
SRE	Speaker Recognition Evaluation

STFT	Short-Time Fourier Transform
sup-GMM	Supervised GMM
SVM	Support Vector Machine
TAR	True Acceptance Rate
TDNNs	Time Delay Neural Networks
TTS	Total Test Speech Samples
TV	Television
UBM	Universal Background Model
VAD	Voice Activity Detection
VOA	Voice of America
VQ	Vector Quantization
WDS	Wrong Detected Samples
WER	word error rate
YOHO	YOHO Speaker Verification Dataset
ZCR	Zero Crossing Rate

Author's Publications

[P1] Win Lai Lai Phyu, Win Pa Pa, "Text Independent Speaker Identification for Myanmar Speech", The 11th International Conference on Future Computer and Communication (*ICFCC*), pp. 86-89, 27-28 February, 2019. Yangon, Myanmar.

[P2] Win Lai Lai Phyu, Win Pa Pa, "Building Speaker Identification Dataset for Noisy Conditions", In *2020 IEEE Conference on Computer Applications (ICCA)*, The 18th International Conference on Computer Applications (IEEE ICCA), pp. 182-188, 27-29 February, 2020. Yangon, Myanmar.

[P3] Win Lai Lai Phyu, Win Pa Pa, "Data Augmentation for Burmese Speaker Identification on Time Delay Neural Network", The 23th International Committee for the Co-ordination and Standardization of Speech Databases and Assessment Techniques (*O-COCOSDA*), pp. 183-186, 5-7 November, 2020, Yangon, Myanmar.

[P4] Win Lai Lai Phyu, Hay Mar Soe Naing, Win Pa Pa, "Improving the Performance of Low-resourced Speaker Identification with Data Preprocessing", Journal of ICT Research and Applications (*JICTRA*), Volume 17, Issue 3, pp. 275-291, December 2023.

Bibliography

[1] Kanagasundaram, A., Vogt, R., Dean, D., Sridharan, S., Mason, M., "i-vector Based Speaker Recognition on Short Utterance", *Proceedings of the 12th Annual Conference of the International Speech Communication Association, INTERSPEECH 2011*, Florence, Italy, pp. 2341-2344, 28-31 August 2011.

[2] Senoussaoui, M., Kenny, P., Dehak, N., Dumouchel, P., "An i-vector Extractor Suitable for Speaker Recognition with both Microphone and Telephone Speech", *Odyssey 2010*, Brno, pp. 6, 28 June 2010.

[3] Kanagasundaram, A., Vogt, R., Dean, D., Sridharan, S., "PLDA based Speaker Recognition on Short Utterances", *Proceedings of The Speaker and Language Recognition Workshop, Odyssey 2012*, Singapore, pp. 28-33, 25-28 June 2012.

[4] Nayana, P. K., Mathew, D., Thomas, A., "Comparison of Text Independent Speaker Identification Systems using GMM and i-Vector Methods", *ICACC-2017*, Cochin, India, pp. 47-54, 22-24 August 2017.

[5] Mehendale, A. S., and Dixit, M. R., "Speaker Identification", *Signal & Image Processing: An International Journal (SIPIJ),* Vol.2, No.2, June 2011.

[6] Cheng, J. M., and Wang, H. C., "A Method of Estimating the Equal Error Rate for Automatic Speaker Identification", *International Symposium on Chinese Spoken Language Processing (ISCSLP 2004)*, Hong Kong, pp. 285-288, 15-18 December 2004, 0-7803-8678-7/04/$20.00 © 2004 IEEE.

[7] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., Lang, K. J., "Phoneme Recognition Using Time Delay Neural Networks", *IEEE Transcations on Acoustics, Speech and Signal Processing*, Vol. 37, No. 3, pp. 328-339, March 1989.

[8] Imam, S. A., Bansal, P., Singh, V., "Review: Speaker Recognition Using Automated Systems", *AGU International Journal of Engineering and Technology, AGUIJET 2017*, Vol. No. 5, pp. 31-39, Jul 2017.

[9] Poddar, A., Sahidullah, M., Saha, G., "Performance Comparison of Duration Variability", *Annual IEEE India Conference (INDICON)*, IEEE 2015, pp. 1-6, 17 Dec 2015.

[10] Qin Jin, Thomas Fang Zheng, “Overview of Front-end Features for Robust Speaker Recognition”, APSIPA ASC 2011, Xi‟an, China.

[11] R.ARUL JOTHI M.E, “Analysis of Suitable Extraction Methods and Classifiers for Speaker Identification”, IRJET 2017, Volume: 04 Issue: 03, Mar 2017.

[12] Rania Chakroun, Leila Beltaifa Zouari, Mondher Frikha, “An Improved Approach for Text-Independent Speaker Recognition”, IJACSA 2016, Vol. 7, No. 8, 2016.

[13] Zhenhao Ge, Sudhendu R. Sharma, Mark J.T. Smith, “PCA/LDA Approach for Text-Independent Speaker Recognition”, 25 Feb 2016.

[14] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L. Seltzer, Sanjeev Khudanpur, “A Study on Data Augmentation of Reverberant Speech for Robust Speech Recognition”, ICASSP 2017, IEEE International Conference on. IEEE, 2017, pp. 5220–5224

[15] David Snyder, Daniel Garcia-Romero, Daniel Povey, “Time Delay Deep Neural Network-Based Universal Background Models for Speaker Recognition,”*ASRU,* 2011.

[16] Alvin F Martin, Craig S Greenberg, “The NIST 2010 Speaker Recognition Evalutaion”, *INTERSPEECH 2010*, Makuhari, Chiba, Japan, 26-30 September 2010.

[17] Mousmita Sarma, Kandarpa Kumar Sarma Nagendra Kumar Goel, “Language Recognition using Time Delay Neural Network”, 13 April 2018.

[18] V. Peddinti, D. Povey, S. Khudanpur, “A Time Delay Neural Network Architecture for Efficient Modeling of Long Temporal Contexts”, Proceedings of Interspeech, 3214-3218, 2015.

[19] H. Park, D. Lee, M. Lim, Y. Kang, J. Oh and J. H. Kim, “A Fast-Converged Acoustic Modeling for Korean Speech Recognition: A Preliminary Study on Time Delay Neural Network”, To cite this article, Boji Liu et.al, 2019 J.Phys : Conf. Ser. 1229 012078.

[20] Padmanabhan Rajan, Anton Afanasyev, Ville Hautamaki, Tomi Kinnunen, “From single to multiple enrollment i-vectors: practical PLDA scoring variants for spaeaker verification”, Digital Signal Processing, Volume 31,

pp. 93-101, Aug 2014.

[21] Snyder, D., Chen, G., Povey, D., "MUSAN: A Music, Speech, and Noise Corpus", *arXiv preprint arXiv:1510.08484*, October 28, 2015.

[22] Kurzekar, P. K., Deshmukh, R. R., Waghmare, V. B., and Shrishrimal, P. P., "Continuous Speech Recognition System A Review", *Asian Journal of Computer Science and Information Technology*, Vol. 4, No. 6, pp. 62-66, Jun 2014.

[23] Kurzekar, P. K., Deshmukh, R. R., Waghmare, V. B., and Shrishrimal, P. P., "A Comparative Study of Feature Extraction Techniques for Speech Recognition System", *International Journal of Innovative Research in Science, Engineering and Technology*, Vol. 3, Issue 12, pp. 18006-18016, Dec 2014.

[24] Davis S., and Mermelstein, P., "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences", *IEEE Transactions on Acoustics, Speech and Signal Processing (ASSP),* 28 (4), 357-366, Aug 1980.

[25] Proakis, J. G., And Manolakis, D., "Digital Signal Processing, Principles, Algorithms and Applications", Second ed., Macmillan Publishing Company, New York, 1992.

[26] Shen, J. L., Hwang, W. L., and Lee, L. S., "Robust Speech Recognition Features based on Temporal Trajectory Filtering of Frequency Band Spectrum", *Proceeding of Fourth International Conference on Spoken Language Processing, ICSLP'96*, IEEE, Vol. 2, pp. 881-884, 3 Oct 1996.

[27] Farahani, F., Georgiou, P. G., and Narayanan, S. S., "Speaker Identification using Supra-segmental Pitch Pattern Dynamics," *IEEE International Conference on Acoustics, Speech, and Signal Processing, Proceedings of ICASSP'04*, , Vol. 1, pp. I-89, 17 May 2004.

[28] Campbell, J. P., "Testing with the YOHO CD-ROM Voice Verification Corpus", *International Conference on Acoustics, Speech, and Signal Processing, ICASSP'95*, Vol. 1, pp. 341-344, 9 May 1995.

[29] Teixeira, C., Trancoso, I., Serralheiro, A., "Accent Identification", *Proceeding of Fourth International Conference on Spoken Language Processing, ICSLP'96*, IEEE, Vol. 3, pp. 1784-1787, 3 Oct 1996.

[30] Kat, L. W., and Fung, P., “Fast Accent Identification and Accented Speech Recognition”, *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP’99* (Cat. No. 99CH36258), Vol. 1, pp. 221-224, 15 Mar 1999.

[31] Chen, T., Huang, C., Chang, E., and Wang, J., “Automatic Accent Identification using Gaussian Mixture Models”, *IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU’01*, pp. 343-346, 9 Dec 2001.

[32] Reynolds, D. A., Rose, R. C., “Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models”, *IEEE Transactions on Speech and Audio Processing*, Vol. 3, No. 1, pp. 72-83, January 1995.

[33] Hazen, T. J., Jones, D. A., Park, A., Kukolich, L. C., and Reynolds, D. A., “Integration of Speaker Recognition into Conversational Spoken Dialogue Systems”, *Eighth European Conference on Speech Communication and Technology*, *Eurospeech*, pp. 1961-1964, 2003.

[34] Chan, M. V., Feng, X., Heinen, J. A., and Niederjohn, R. J., “Classification of Speech Accents with Neural Networks”, *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, Vol. 7, pp. 4483-4486, 28 Jun 1994.

[35] Deshpande, S., Chikkerur, S., and Govindaraju, V., “Accent Classification in Speech”, *Fourth IEEE Workshop on Automatic Identification Advanced Technologies (AutoID'05)*, pp. 139-143, 17 Oct 2005.

[36] Tanabian, M. M., Tierney, P., and Azami, B. Z., “Automatic Speaker Recognition with Formant Trajectory Tracking using CART and Neural Networks”, *Canadian Conference on Electrical and Computer Engineering*, IEEE, pp. 1225-1228, Canada, 1 May 2005.

[37] Gray, S., and Hansen, J. H., “An Integrated Approach to the Detection and Classification of Accents/Dialects for a Spoken Document Retrieval System”, *IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 35-40, 27 Nov 2005.

[38] Bartkova, K., and Jouvet, D., “Using Multilingual Units for Improved Modeling of Pronunciation Variants”, Acoustics, *IEEE International*

Conference on Speech and Signal Processing, ICASSP 2006, Vol. 5, pp. 1037-1040, 14 May 2006.

[39] Angkititrakul, P., and Hansen, J. H., "Advances in Phone-Based Modeling for Automatic Accent Classification", *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 14, No. 2, pp. 634-646, 21 Feb 2006.

[40] Pruzansky, S., and Mathews, M. V., "Talker Recognition Procedure based on Analysis of Variance", *The Journal of the Acoustical Society of America*, 36(11), pp. 2041-2047, Nov 1964.

[41] Li, K.P., Dammann, J.E. and Chapman, W.D., "Experimental Studies in Speaker Verification using an Adaptive System", *The Journal of the Acoustical Society of America*, 40(5), pp. 966-978, Nov 1966.

[42] Furui, S., "Cepstral Analysis Technique for Automatic Speaker Verification", *IEEE Transactions on Acoustics, Speech, Signal Processing 29, ASSP-29*, pp. 254-272, Apr 1981.

[43] Matsui, T., and Furui, S., "Concatenated Phoneme Models for Text-variable Speaker Recognition", In *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, Vol. 2, pp. 391-394, 27 Apr 1993.

[44] Doddington, G., "Speaker Recognition based on Idiolectal Differences between Speakers", In *Seventh European Conference on Speech Communication and Technology, Eurospeech*, pp. 2521-2524, 2001.

[45] Peddinti, V., Chen, G., Povey, D., Khudanpur, S., "Reverberation Robust Acoustic Modeling using i-vectors with Time Delay Neural Networks", In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[46] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian,Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K., "The Kaldi Speech Recognition Toolkit*", In IEEE 2011 workshop on automatic speech recognition and understanding 2011, ASRU2011*, 2011.

[47] Price, J. R., Gee, T. F., "Face Recognition using Direct, Weighted Linear Discriminant Analysis and Modular Subspaces", *Pattern Recognition*, Vol.

38, No. 2, pp. 209-219, 1 Feb 2005.

[48] Senoussaoui, M., Kenny, P., Brummer, N., Villiers, E. D., Dumouchel, P., "Mixture of PLDA Models in i-vector Space for Gender Independent Speaker Recognition", *In Twelfth Annual Conference of the International Speech Communication Association,* pp. 25-28, 2011.

[49] Kenny, P., "Bayesian Speaker Verification with Heavy Tailed Priors", *Proceedings of Odyssey Speaker and Language Recognition Workshop 2010, Odyssey2010,* Brno, Czech Republic, 2010.

[50] Burget, L., Plchot, O., Cumani, S., Glembek, O., Matejka, P., Brummer, N., "Discriminatively Trained Probalistic Linear Discriminant Analysis for Speaker Verification", *In 2011 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2011,* pp. 4832-4835, 22 May 2011.

[51] Jin Q. and Waibel A., "Application of LDA to Speaker Recognition", *Sixth International Conference on Spoken Language Processing,* pp. 250-253, Oct 2000.

[52] Gish, H., and Schmidt, M., "Text-independent Speaker Identification", *IEEE Signal Processing Magazine,* Vol. 11, No. 4, pp. 18-32, October 1994.

[53] Poritz, A., "Linear Predictive Hidden Markov Models and the Speech Signal", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'82),* Vol. 7, pp. 1291-1294, 3 May 1982.

[54] Tishby, N. Z., "On the Application of Mixture AR Hidden Markov Models to Text Independent Speaker Recognition", *IEEE Transactions on Signal Processing,* Vol. 39, No. 3, pp. 563-570, Mar 1991.

[55] Seddik, H., Rahmouni, A., and Sayadi, M., "Text Independent Speaker Recognition using the Mel Frequency Cepstral Coefficients and a Neural Network Classifier", In *First International Symposium on Control, Communications and Signal Processing,* IEEE, pp. 631-634, 21 Mar 2004.

[56] Soong, F. K., Rosenberg, A. E., Rabiner, L. R., and Juang, B. H., "A Vector Quantization Approach to Speaker Recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'85), AT&T technical journal,* Vol. 66, No. 2, pp. 14-

26, March 1987.

[57] Matsui, T., and Furui, S., "Comparison of Text-independent Speaker Recognition Methods using VQ-distortion and Discrete/Continuous HMMs", *IEEE Transactions on Speech and Audio Processing*, Vol. 2, No. 3, pp. 456-459, Jul 1994.

[58] Gish, H., Krasner, M., Russell, W., and Wolf, J., "Methods and Experiments for Text-independent Speaker Recognition over Telephone Channels", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'86)*, Vol. 11, pp. 865-868, 7 Apr 1986.

[59] Majetniak, A., Tan, Z. H., "Speaker Recognition using Universal Background Model on YOHO Database", *Aalborg University, The Faculties of Engineering, Science and Medicine Department of Electronic Systems*, pp. 1-61, 31 May 2011.

[60] Zheng, R., Zhang, S., Xu, B., "Text Independent Speaker Identification using GMM-UBM and Frame Level Likelihood Normalization", *IEEE International Symposium on Chinese Spoken Language Processing (ISCSL 2004)*, Hong Kong, pp. 289-292, 15-18 Dec 2004.

[61] Al Marashli, A., Al Dakkak, O., "Automatic, Text-Independent, Speaker Identification and Verification System using Mel Cepstrum and GMM", *The 3rd International Conference on Information and Communication Technologies: From Theory of Applications (ICTTA 2008)*, IEEE, pp. 1-6, 7 Apr 2008.

[62] McLaren, M., Lei, Y., and Ferrer, L., "Advances in Deep Neural Network Approaches to Speaker Recognition", *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015*, pp. 4814-4818, 19 Apr 2015.

[63] Zhang, X., Trmal, J., Povey, D., and Khudanpur, S., "Improving Deep Neural Network Acoustic Models using Generalized Maxout Networks", *2014 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014*, pp. 215-219, 4 May 2014.

[64] Cumani, S., Plchot, O., Laface, P., "Probabilistic Linear Discriminant Analysis of i-Vector Posterior Distributions", *2013 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP2013*, pp.

7644-7648, 26 May 2013.

[65] Bousquet, P. M., Larcher, A., Matrouf, D., Bonastre, J. F., Plchot, O., "Variance-Spectra based Normalization for i-Vector Standard and Probabilistic Linear Discriminant Analysis", *Proceeding of Odyssey 2012: The Speaker and Language Recognition Workshop*, pp. 157-164, 25 Jun 2012.

[66] Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., and Outllet, P., "Front-End Factor Analaysis for Speaker Verification", *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 19, No. 4, pp. 788-798, 9 Aug 2010.

[67] Wang, Y., Xu, H., Ou, Z., "Joint Bayesian Gaussian Discriminant Analysis for Speaker Verification", In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP2017*), pp. 5390-5394, 5 Mar 2017. https://arxiv.org/abs/1612.04056, 2017.

[68] Snyder, D., Garcia-Romero, D., Povey, D., Khudanpur, S., "Deep Neural Network Embeddings for Text Independent Speaker Verification", *Proceedings of Interspeech 2017*, pp. 999-1003, 20 Aug 2017.

[69] Snyder, D., Garcia-Romero, D., Sell, G., Povey, and D., Khudanpur, S., "X-vectors: Robust DNN Embeddings for Speaker Recognition", In *2018 IEEE international conference on acoustics, speech and signal processing, ICASSP2018*, pp. 5329-5333, 15 Apr 2018.

[70] Snyder, D., "NIST SRE 2016 Xvector Recipe", https://david-ryan-snyder.github.io/2017/10/04/model_sre16_v2.html, 2017.

[71] Povey, D., Zhang, X., Khudanpur, S., "Parallel Training of Deep Neural Networks with Natural Gradient and Parameter Averaging", *arXiv preprint arXiv:1410.7455, CoRR*, Vol. abs/1410.7455, Oct 2014. [Online]. Available: http://arxiv.org/abs/1410.7455.

[72] Ioffe, S., "Probabilistic Linear Discriminant Analysis", *European Conference on Computer Vision, Springer*, Berlin, Heidelberg, pp. 531-542, 7 May 2006.

[73] Kabir, M. M., Mridha, M. F., Shin, J., Jahan, I., and Ohi, A. Q., "A Survey of Speaker Recognition: Fundamental Theories, Recognition Methods and Opportunities", *IEEE Access*, 27 May 2021.

[74] Meng, A., Ahrendt, P., and Larsen, J., "Improving Music Genre Classification by Short Time Feature Integration", *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'05*, Vol. 5, pp. v-497-v-500, 23 Mar 2005.

[75] Hernawan. S., "Speaker Diarization: Its Developments, Applications and Challenges", In *Proceedings of The 1st International Conference on Information Systems For Business Competitiveness (ICISBC)*, 19 Sep 2012.

[76] Mon, A. N., Pa, W. P., Thu, Y. K., "Building HMM-SGMM Continuous Automatic Speech Recognition on Myanmar Web News", *International Conference on Computer Applications, ICCA 2017*, Yangon, Myanmar, 16-17 Feb 2017.

[77] Ming, J., Hazen, T. J., Glas, J. R., and Reynolds, D. A., "Robust Speaker Recognition in Noisy Conditions", IEEE Transactions on Audio, Speech, and Language Processing, Vol. 15, No. 5, pp. 1711-1723, 18 Jun 2007.

APPENDIX

APPENDIX: Developing GMM-UBM based Automatic Speaker Identification

KaldiASR toolkit is used to implement the Burmese speaker identification. Data preparation, feature extraction, building acoustic speaker models, creating trial files, and identification process by using this toolkit are presented in this appendix.

1. Data Preparation

Data preparation is a preliminary and necessary step to set up the Burmese speaker system with the speech corpus. This section describes details how to prepare the data for training, development and test sets for Burmese language. Prepared data are stored in the "data" directory containing information about the specific of the audio files.

1.1 Data Preparation ("data" directory)

In the "data" part for training, validation and test sets data, audio files' path related to utterances (wav.scp), speaker and utterance mappings: utt2spk and spk2utt are necessary to prepare manually.

(a) wav.scp

This file connects every utterance (sentence said by one person during particular recording session) with an audio file related to this utterance.

Pattern: ỢÕ Ô Ô Å Ò ǺƠ Î Ã ÅỢĘ)ÕȘĖÌ ÂỢÔĖẸÁ Õ ÄẸÉĖỚÌ Å

```
v2# head -5 data/train/wav.scp
mm-winlailaiphyu_10001 /root/kaldi/data/train/mm-winlailaiphyu_10001.wav
mm-winlailaiphyu_10002 /root/kaldi/data/train/mm-winlailaiphyu_10002.wav
mm-winlailaiphyu_10003 /root/kaldi/data/train/mm-winlailaiphyu_10003.wav
mm-winlailaiphyu_10004 /root/kaldi/data/train/mm-winlailaiphyu_10004.wav
mm-winlailaiphyu_10005 /root/kaldi/data/train/mm-winlailaiphyu_10005.wav
```

(b) utt2spk

The "utt2spk" file tells the system which utterance belongs to a particular speaker.

Pattern: ?Õ Ô Ô Å Ò Ý Î Ã Å?Ó) Ð$Å Á ËÓÅ Ò) $

```
v2# head -5 data/train/utt2spk
mm-winlailaiphyu_10001 mm-winlailaiphyu
mm-winlailaiphyu_10002 mm-winlailaiphyu
mm-winlailaiphyu_10003 mm-winlailaiphyu
mm-winlailaiphyu_10004 mm-winlailaiphyu
mm-winlailaiphyu_10005 mm-winlailaiphyu
```

(c) spk2utt

The "spk2utt" file tells the system which utterances speak by a particular speaker. It can be produced with the following command by using the "utt2spk" file.

```
utils/utt2spk_to_spk2utt.pl data/train/utt2spk > data/train/spk2utt
```

Pattern: ?Ó Ð Å Á ËÓÅ Ò) $?Õ Ô Ô Å Ò Ý Î Ã Å) $

```
v2# head -1 data/train/spk2utt
mm-winlailaiphyu mm-winlailaiphyu_10001 mm-winlailaiphyu_10002 mm-winlailaiphyu_10003 mm-winlailaiphyu_10004 mm-winlailaiphyu_10005
```

2. Data Augmentation

Data augmentation is used for the purpose of increasing the training data size by creating new and different data from existing data artificially. Additive noise and reverberation to original training data are employed. Firstly, convolving simulated room impulse responses (RIRs) with audio is applied to add the echo to the speech data artificially by using the following command.

```
steps/data/reverberate_data_dir.py data/train data/train_reverb
```

The data files under "data/train" are used to produce the output of the audio files' path related to utterances (wav.scp), speaker and utterance mappings: utt2spk and spk2utt.

(a) wav.scp

This file connects every utterance (sentence said by one person during particular recording session) with an audio file related to this utterance.

Pattern: ꝺÕ Ô Ô Å Ò Ǿ Î Ã Å ꝺ) Õ ꝺ Ì Â ꝺ Ë Á Õ Ä ꝺ É Ǿ Ì Å

```
v2# head -3 data/train_reverb/wav.scp
mm-winlailaiphyu_10001-reverb cat /root/kaldi/data/train/mm-winlailaiphyu_10001.wav | wav-reverberate --shift-output=true --impulse-response="sox RIRS_NOISES/simulated_rirs/mediumroom/Room072/Room072-00084.wav -r 16000 -t wav - |" - - |
mm-winlailaiphyu_10002-reverb cat /root/kaldi/data/train/mm-winlailaiphyu_10002.wav | wav-reverberate --shift-output=true --impulse-response="sox RIRS_NOISES/simulated_rirs/mediumroom/Room036/Room036-00041.wav -r 16000 -t wav - |" - - |
mm-winlailaiphyu_10003-reverb cat /root/kaldi/data/train/mm-winlailaiphyu_10003.wav | wav-reverberate --shift-output=true --impulse-response="sox RIRS_NOISES/simulated_rirs/mediumroom/Room091/Room091-00020.wav -r 16000 -t wav - |" - - |
```

(b) utt2spk

The “utt2spk” file tells the system which utterance belongs to a particular speaker.

Pattern: ꝺÕ Ô Ô Å Ò Ǿ Î Ã Å ꝺÓ) Ð $ Å Á Ë Ǿ Å Ò) $

```
v2# head -5 data/train_reverb/utt2spk
mm-winlailaiphyu_10001-reverb mm-winlailaiphyu
mm-winlailaiphyu_10002-reverb mm-winlailaiphyu
mm-winlailaiphyu_10003-reverb mm-winlailaiphyu
mm-winlailaiphyu_10004-reverb mm-winlailaiphyu
mm-winlailaiphyu_10005-reverb mm-winlailaiphyu
```

(c) spk2utt

The “spk2utt” file tells the system which utterances speak by a particular speaker. It can be produced with the following command by using the “utt2spk” file.

```
utils/utt2spk_to_spk2utt.pl data/train_reverb/utt2spk > data/ train_reverb /spk2utt
```

Pattern: ꝺÓ Ð Å Á Ë Ǿ Å Ò) $ ꝺÕ Ô Ô Å Ò Ǿ Î Ã Å) $

```
v2# head -1 data/train_reverb/spk2utt
mm-winlailaiphyu mm-winlailaiphyu_10001-reverb mm-winlailaiphyu_10002-reverb mm-winlailaiphyu_10003-reverb mm-winlailaiphyu_10004-reverb mm-winlailaiphyu_10005-reverb
```

MUSAN: Music, Speech and Noise corpus is used for augmenting the existing training data consisting of over 900 noises, 60 hours of speech and 42 hours of music by using the following command to obtain the MUSAN corpus which consists of music, speech and noise suitable for augmentation.

```
steps/data/make_musan.sh path_to_MUSAN_corpus data
```

MUSAN corpus is used to produce the outputs of the audio files' path related to utterances (wav.scp), speaker and utterance mappings: utt2spk and spk2utt. The data folders: speech, noise, and music under MUSAN corpus are used to produce the corresponding output folders: "data/musan_noise", "data/musan_speech", and "data/musan_music". The output under "data/musan_noise" is represented as an example output.

(a) wav.scp

Pattern: ộÕ Ô Ô Å Ò Ớ Î Ã ÅệĘ)Õ$ĐÌ Â ỘÔĖÏÁ Õ Ä ẸÉ ĐỚÌ Å

```
v2# head -5 data/musan_noise/wav.scp
noise-free-sound-0000 /root/kaldi/egs/sitw/musan/noise/free-sound/noise-free-sound-0000.wav
noise-free-sound-0001 /root/kaldi/egs/sitw/musan/noise/free-sound/noise-free-sound-0001.wav
noise-free-sound-0002 /root/kaldi/egs/sitw/musan/noise/free-sound/noise-free-sound-0002.wav
noise-free-sound-0003 /root/kaldi/egs/sitw/musan/noise/free-sound/noise-free-sound-0003.wav
noise-free-sound-0004 /root/kaldi/egs/sitw/musan/noise/free-sound/noise-free-sound-0004.wav
```

(b) utt2spk and spk2utt

The "utt2spk" and "spk2utt" are the same files because the data files are the individually separated files.

Pattern: ộÕ Ô Ô Å Ò Ớ Î Ã ÅộÓ)Đ$Å Á ĖỚÅ Ò) $

```
v2# head -5 data/musan_noise/utt2spk
v2# head -5 data/musan_noise/spk2utt

noise-free-sound-0000 noise-free-sound-0000
noise-free-sound-0001 noise-free-sound-0001
noise-free-sound-0002 noise-free-sound-0002
noise-free-sound-0003 noise-free-sound-0003
noise-free-sound-0004 noise-free-sound-0004
```

After that, the data folders: “data/musan_noise”, “data/musan_speech”, and “data/musan_music” are used to produce the corresponding augmented output folders: “data/train_noise”, “data/ train _music”, and “data/ train _babble” with these commands.

```
steps/data/augment_data_dir.py “data/musan_noise” data/train data/train_noise
```

```
steps/data/augment_data_dir.py “data/musan_music” data/train data/train_music
```

```
steps/data/augment_data_dir.py “data/musan_speech” data/train data/train_babble
```

The output under “data/train_noise” is represented as an example output.

(a) wav.scp

Pattern: ộÕ Ô Ô Å Ò Ǿ Î Ã ÅộÈ)ÕẸÌ Â ộÕẸÅ Õ ÄẠÉ ÉǾÌ Å

```
v2# head -3 data/train_noise/wav.scp
mm-winlailaiphyu_10001-noise wav-reverberate --shift-output=true --additive-signals='/root/kaldi/egs/sitw/musan/noise/free-sound/noise-free-sound-0643.wav' --start-times='0' --snrs='15' / home/winlai/Data/train/mm-winlailaiphyu_10001.wav - |
mm-winlailaiphyu_10002-noise wav-reverberate --shift-output=true --additive-signals='/root/kaldi/egs/sitw/musan/noise/sound-bible/noise-sound-bible-0036.wav' --start-times='0' --snrs='5' /home/winlai/Data/train/mm-winlailaiphyu_10002.wav - |
mm-winlailaiphyu_10003-noise wav-reverberate --shift-output=true --additive-signals='/root/kaldi/egs/sitw/musan/noise/free-sound/noise-free-sound-0734.wav' --start-times='0' --snrs='0' /home/winlai/Data/train/mm-winlailaiphyu_10003.wav - |
```

(b) utt2spk

The “utt2spk” file tells the system which utterance belongs to a particular speaker.

Pattern: ộÕ Ô Ô Å Ò Ǿ Î Ã ÅộÓ)Ð$Å Á ËǾÅ Ò) $

```
v2# head -5 data/train_ noise/utt2spk
mm-winlailaiphyu_10001-noise mm-winlailaiphyu
mm-winlailaiphyu_10002-noise mm-winlailaiphyu
mm-winlailaiphyu_10003-noise mm-winlailaiphyu
mm-winlailaiphyu_10004-noise mm-winlailaiphyu
mm-winlailaiphyu_10005-noise mm-winlailaiphyu
```

(c) spk2utt

The “spk2utt” file tells the system which utterances speak by a particular speaker. It can be produced with the following command by using the “utt2spk” file.

```
utils/utt2spk_to_spk2utt.pl data/train_reverb/utt2spk > data/ train_reverb /spk2utt
```

Pattern: ôÓ Ð Å Á ËØÅ Ò) $ÕÔ Ô Å Ò ÁÝ Î Ã Å) $

```
v2# head -1 data/train_reverb/spk2utt
mm-winlailaiphyu mm-winlailaiphyu_10001-noise mm-winlailaiphyu_10002-noise mm-winlailaiphyu_10003-noise mm-winlailaiphyu_10004-noise mm-winlailaiphyu_10005-noise
```

The training data naming “data/train_reverb”, “data/train_noise”, “data/train_music”, and “data/train_babble” has four times than that of original training data. These are obtained from reverberating the training data and augmenting the training data with MUSAN corpus. And then, these four augmented data files are combined into the folder name “data/aug” by using the following command.

```
utils/combine_data.sh data/aug data/train_reverb data/train_noise data/train_music data/train_babble
```

3. Feature Extraction

Feature extraction step is performed after preparing the data. In this work, MFCC feature extraction technique is used. The “feats.scp” file is made by the following command. This command is only for the MFCC feature extraction method. It will create “feats.scp” in “data/train/feats.scp” with corresponding archives in a folder called “mfccvad” and written log files to exp/make_mfcc.

```
steps/make_mfcc.sh data/train exp/make_mfcc/ ./mfccvad
```

The data files under “data/aug” are also used to extract the feature like extracting the feature and detecting silent and non-silent speech frames from the data under “data/train”. After that, these data under the folders “data/train” and “data/aug” are combined using the following command to form “data/combinetrain”.

```
utils/combine_data.sh data/combinetrain data/train data/aug
```

For voice activity detection, the “vad.scp” file is made by the following command. It will create “vad.scp” in “data/train/vad.scp” with corresponding archives in a folder called “mfccvad” and written log files to exp/make_vad. For “data/aug”, only “data/train/vad.scp” file is used without creating the separated “vad.scp” file.

```
sid/compute_vad_decision.sh data/train exp/make_vad/ ./mfccvad
```

The example output of MFCC feature format in Kaldi is as follows.

Pattern: ỘÕ Ô Ô Å Ò ÁÓ Î Ã ÅỘÅ)ØŞÔ Å ẬÆÄÉÅÌ ÄÅ Ï ÁẸ Í&Å#ÆĘ Å Á ÔÓÕ Ò Å Ó

```
v2# head -5 data/combinetrain/feats.scp
mm-winlailaiphyu_10001  /root/kaldi/egs/sitw/v2/mfccvad/raw_mfcc_combinetrain.1.ark:
955828779

mm-winlailaiphyu_10002  /root/kaldi/egs/sitw/v2/mfccvad/raw_mfcc_combinetrain.1.ark:
955833143

mm-winlailaiphyu_10003  /root/kaldi/egs/sitw/v2/mfccvad/raw_mfcc_combinetrain.1.ark:
955843897

mm-winlailaiphyu_10004   /root/kaldi/egs/sitw/v2/mfccvad/raw_mfcc_combinetrain.1.ark:
955851171

mm-winlailaiphyu_10005         /root/kaldi/egs/sitw/v2/mfccvad/raw_mfcc_combinetrain.1.ark:
955868615
```

The example output of VAD value in Kaldi format is as follows.

Pattern: ỘÕ Ô Ô Å Ò ÁÓ Î Ã ÅỘÅ)ØŞÔ Å ẬÆÄÉÅÌ ÄÅ Ï ÁĢ Í!ÅŞÆ Å Á ÔÓÕ Ò Å Ó

```
v2# head -5 data/combinetrain /vad.scp
mm-winlailaiphyu_10001  /root/kaldi/egs/sitw/v2/mfccvad/vad_combinetrain.1.ark: 127133025

mm-winlailaiphyu_10002  /root/kaldi/egs/sitw/v2/mfccvad/ vad_combinetrain.1.ark: 127133602

mm-winlailaiphyu_10003 /root/kaldi/egs/sitw/v2/mfccvad/ vad_combinetrain.1.ark: 127135031

mm-winlailaiphyu_10004 /root/kaldi/egs/sitw/v2/mfccvad/ vad_combinetrain.1.ark: 127135996

mm-winlailaiphyu_10005 /root/kaldi/egs/sitw/v2/mfccvad/ vad_combinetrain.1.ark: 127138317
```

4. Training GMM-UBM based Acoustic Model

Gaussian Mixture Model_Universal Background Model (GMM_UBM) based system is trained on top of MFCC features to build the acoustic speaker models with

parameter tuning like changing the number of components and dimensions in this work. It includes the following steps.

4.1 Training Diagonal UBM

Firstly, the data under “data/combinetrain” are used to train the diagonal UBM. The number of components and dimensions can be able to change to produce the diagonal UBM. As an example, 400 Gaussian components and 200 i-vector dimensions are used to produce “final.dubm” by using the following command.

```
sid/train_diag_ubm.sh data/combinetrain 400 exp/diag_ubm_400
```

After running, the output diagonal background model is produced under “exp/diag_ubm_400” as final.dubm.

4.2 Training Full UBM

The diagonal background model, “final.dubm” under “exp/diag_ubm_400” is used for training full UBM by the following command.

```
sid/train_full_ubm.sh data/combinetrain  exp/diag_ubm_400 exp/full_ubm_400
```

The output model full UBM is produced under “exp/full_ubm_400” as final.ubm. This background model is used to construct i-vector extractor.

4.3 Constructing i-Vector Extractor

The data under “data/combinetrain” and “final.ubm” under “exp/full_ubm_400” are used to construct i-vector extractor with this command to produce the extractor “final.ie” storing under “exp/extractor”.

```
sid/train_ivector_extractor.sh exp/full_ubm_400/final.ubm data/combinetrain exp/extractor
```

4.4 Extracting i-Vector

Extracting i-vector corresponding to each utterance is performed by i-vector extractor located in “exp/extractor” to produce “ivector.scp”, “ivector.ark”, “spk_ivector.scp” and “spk_ivector.ark” with the following command. “ivector.ark”, and “spk_ivector.ark” are the archive files.

```
sid/extract_ivectors.sh exp/extractor data/combinetrain exp/ivectors_combinetrain
```

The process of extracting i-vector is stored as the log file under "exp/ivectors_combinetrain/log". The example output of "ivector.scp" is as follows.

Pattern: ộÕ Ô Ô Å Ò Ấ Î Ã Åộì) Ï$Ã Á ̨ÔǼ Ö Å Ą̃ÖÔÁ ÏÌŐÕ Å Ó

```
v2# head -5 exp/ivector_combinetrain/ivector.scp
mm-winlailaiphyu_10001-babble exp/ivectors_combined/ivector.1.ark:176015995
mm-winlailaiphyu_10002 exp/ivectors_combined/ivector.1.ark:176018089
mm-winlailaiphyu_10002-music exp/ivectors_combined/ivector.1.ark:176020164
mm-winlailaiphyu_10003 exp/ivectors_combined/ivector.1.ark:176022233
```

The example output of "spk_ivector.scp" is as follows.

Pattern: ộÓ Ð Å Á ËŐÅộÌÒ Ï) Ã$Á ̨ÔǺ Ä Ą̃ÓÐ Å Ą́ËÖÅÒĄ̃ÖÔÁ ÏÌÓÕ Å

```
v2# head -5 exp/ivector_combinetrain/spk_ivector.scp
mm-winlailaiphyu exp/ivectors_combined/spk_ivector.ark:128618
mm-wintwarhlaing exp/ivectors_combined/spk_ivector.ark:129445
mm-yadanaroo exp/ivectors_combined/spk_ivector.ark:130268
mm-yaminthu exp/ivectors_combined/spk_ivector.ark:131090
mm-yekhaungmyintmg exp/ivectors_combined/spk_ivector.ark:131919
```

The same process as extracting i-vectors of "data/combinetrain" is also used to extract i-vector for validation and testset. The output i-vectors are stored in the corresponding folders "exp/ivectors_dev" and "exp/ivectors_testset" respectively.

5. Training Time Delay Neural Network based Acoustic Model

KaldiASR toolkit is used to implement TDNN based Burmese speaker identification. Time Delay Neural Network (TDNN) based system is used to build the acoustic speaker models with changing different network contexts. Eight different acoustic models are constructed based on network contexts with MFCC feature in this work. It includes the following steps.

5.1 Extracting x-Vector

The required files are prepared for embedding x-vectors of “combinetrain”, “validation”, and “testset” with this command. The required output files are stored in “exp/xvector_nnet_1a/egs” for further use in extracting x-vectors.

```
local/nnet3/xvector/run_xvector.sh data/combinetrain exp/xvector_nnet_1a
exp/xvector_nnet_1a/egs
```

Once trained, the 512-dimensional activations of the penultimate full connected layer are extracted as an x-vector with the following command to produce “xvector.scp”, “spk_xvector.scp” and “spk_xvector.ark”.

```
sid/nnet3/xvector/extract_xvectors.sh exp/xvector_nnet_1a data/combinetrain
exp/xvector_nnet_1a/xvector_combinetrain
```

The process of extracting x-vector is stored as the log file under “exp/xvectors_combinetrain/log”. The example output of “xvector.scp” is as follows.

Pattern: ọÕ Ô Ô Å Ò Ø Ý Î Ã Å ọì) ï$Ã Á ọÔ Æ Ø Ï Å Ã ọÖ Ô Áï ìÖ Õ Å Ó

```
v2# head -5 exp/xvector_nnet_1a/xvector_combinetrain/xvector.scp
mm-winlailaiphyu_10001 exp/xvector_nnet_1a/xvectors_combinetrain/xvector.1.ark:176975619
mm-winlailaiphyu_10002 exp/xvector_nnet_1a/xvectors_combinetrain/xvector.1.ark:176979788
mm-winlailaiphyu_10003 exp/xvector_nnet_1a/xvectors_combinetrain/xvector.1.ark:176983956
mm-winlailaiphyu_10004 exp/xvector_nnet_1a/xvectors_combinetrain/xvector.1.ark:176990213
mm-winlailaiphyu_10005 exp/xvector_nnet_1a/xvectors_combinetrain/xvector.1.ark:176992294
```

The example output of “spk_xvector.scp” is as follows.

Pattern: ọÓ Ð Å Á ËÖ ọì Ï) Ã$Á ọÔ Æ Ë Ï Ã ọÓ Ë Ð Å Á ọÆ Ë Ø Å Ã ọÖ Ô Áï ìÖ Õ Å

```
v2# head -5 exp/xvector_nnet_1a/xvector_combinetrain/spk_xvector.scp
mm-winlailaiphyu exp/xvector_nnet_1a/xvectors_combinetrain/spk_xvector.ark:323306
mm-wintwarhlaing exp/xvector_nnet_1a/xvectors_combinetrain/spk_xvector.ark:325381
mm-yadanaroo exp/xvector_nnet_1a/xvectors_combinetrain/spk_xvector.ark:327452
mm-yaminthu exp/xvector_nnet_1a/xvectors_combinetrain/spk_xvector.ark:329522
mm-yekhaungmyintmg exp/xvector_nnet_1a/xvectors_combinetrain/spk_xvector.ark:331599
```

The same process as extracting x-vectors of “data/combinetrain” is also used to extract x-vector for validation and testset. The output x-vectors are stored in the corresponding folders “exp/xvectors_dev” and “exp/xvectors_testset” respectively.

6. Creating Trial Files

The mapping of speaker and utterance determining whether target speaker or not is created by using “spk_ivector.scp” in “exp/ivectors_combinetrain” and “ivector.scp” in “exp/ivectors_dev” for further use in assessing the model performance in terms of Equal Error Rate (EER). The trial file for testset is also created for assessing the testset performance. The example output of “trial” is as follows:

Pattern: [illegible]

```
v2# head -5 exp/trial
mm-winlailaiphyu mm-winlailaiphyu_10183 target
mm-wintwarhlaing mm-winlailaiphyu_10183 nontarget
mm-yadanaroo mm-winlailaiphyu_10183 nontarget
mm-yaminthu mm-winlailaiphyu_10183 nontarget
mm-yekhaungmyintmg mm-winlailaiphyu_10183 nontarget
```

Creating trial files for TDNN based system are the same task as in GMM-UBM based system.

7. Training Probabilistic Linear Discriminant Analysis (PLDA)

Probabilistic linear discriminant analysis (PLDA) is used to produce the PLDA model for scoring in both GMM-UBM and TDNN.

7.1 PLDA for GMM-UBM based Acoustic Model

The PLDA model is trained by using “data/combinetrain/spk2utt” and “exp/ivectors_combinetrain/ivector.scp” with the command “ivector_compute_plda”. The process of generating PLDA is stored in the log file named “plda.log” under “exp/ivectors_combinetrain/log”.

7.2 PLDA for TDNN based Acoustic Model

Before PLDA model is trained, linear discriminant analysis (LDA) is firstly applied to decrease the dimensionality prior to PLDA. The dimension of LDA is reduced to 128 and it is firstly generated by the command “ivector_compute_lda” with the use of “exp/xvector_nnet_1a/xvectors_combinetrain/xvector.scp” and “data/combinetrain/utt2spk”. The output is produced in the Microsoft access table format named “exp/xvector_nnet_1a/xvectors_combinetrain/transform.mat”. The process of applying LDA is stored as the log file named “lda.log” under “exp/xvector_nnet_1a/xvectors_combinetrain /log”.

The PLDA model is trained with the command “ivector_compute_plda” by using “data/combinetrain/spk2utt”, “exp/xvectors_combinetrain/xvector.scp” and “exp/xvector_nnet_1a/xvectors_combinetrain/transform.mat”. The process of generating PLDA is stored in the log file named “plda.log” under “exp/xvectors_combinetrain/log”.

8. Computing Scores

For assessing the acoustic speaker models’ performance, the corresponding PLDA scores for each speaker is evaluated.

8.1 Computing Scores for GMM-UBM based Acoustic Model

After generating PLDA for use where compute the identification score for each speaker, the acoustic model performance is assessed with the command “ivector_plda_scoring”. The files:“exp/ivectors_validation/num_utts.ark”, “exp/ivectors_combinetrian/plda”, “exp/ivectors_combinetrian/spk_ivector.ark”, “exp/ivectors_validation/ivector.scp” and validation trial file under “exp/” are required to generate the validation scores. The process of generating scores is stored as the log file named “validation_scoring.log” under “exp/scores/log” and the output score is stored in “exp/scores/validation_scores”.

8.2 Computing Scores for TDNN based Acoustic Model

The acoustic model performance is assessed with the command “ivector_plda_scoring” after generating PLDA for computing the speakers’

identification scores. The process of generating scores is stored as the log file named "validation_scoring.log" under "exp/scores/log". The output score is stored in "exp/scores/validation_scores". The files: "exp/ivectors_validation/num_utts.ark", "exp/ivectors_combinetrian/spk_ivector.ark", "exp/ivectors_combinetrian/plda", "exp/ivectors_validation/ivector.scp" and validation trial file under "exp/" are required to generate the validation scores. The example format of score files is as follows:

Pattern: ǫÓ Ð Å Á ËÖÅǫÕÕÔ)ÇÕ¥ǫÐ$Ì ÆÖÆÃ ÏǪÐ Å Ó

```
v2# head -5 exp/validation_scores
mm-winlailaiphyu mm-winlailaiphyu_10181 24.15842
mm-wintwarhlaing mm-winlailaiphyu_10181 -39.32951
mm-yadanaroo mm-winlailaiphyu_10181 -3.227915
mm-yaminthu mm-winlailaiphyu_10181 -35.46188
mm-yekhaungmyintmg mm-winlailaiphyu_10181 -46.10092
```

9. Assessing the Acoustic Model Performance

EER is calculated using "compute_eer" command with the call "local/prepare_for_eer.py" by taking the corresponding trial and score files. The EER performance is evaluated by this command. The less the output EER value, the better the model performance.

```
'compute_eer < (local/prepare_for_eer.py $validationtrials exp/scores/validation_scores)'
```

10. Assessing the Testset Performance

The automatic evaluation of detecting accuracy is used for assessing the testset performance with the following command. The output value is stored in "exp/DetectedTestsetAccuracy". The more the accuracy value, the better the testset accuracy.

```
./calculate_accuracy_for_testset.py exp/scores/testset_scores exp/DetectedTestsetResults
```