PROSODY BASED PHONETIC ENGINE AND SPEAKER CLASSIFICATION FOR PUNJABI LANGUAGE

A thesis

submitted in partial fulfilment of the requirements for the award of degree of

Doctor of Philosophy

Submitted By

RUPINDERDEEP KAUR

(951203008)

Under the supervision of

DR. R. K. SHARMA

(Professor)

DR. PARTEEK KUMAR

(Professor)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY

PATIALA-147004, PUNJAB INDIA.

November, 2021

Certificate

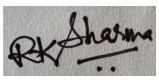
I, *Rupinderdeep Kaur*, Regn. No. 951203008, hereby declare that the thesis entitled "**Prosody Based Phonetic Engine and Speaker Classification for Punjabi Language**" submitted to the Computer Science and Engineering Department at Thapar Institute of Engineering and Technology, Patiala, Punjab, India is an authenticated record of my own work for the award of the degree of "Doctor of Philosophy" under the supervision of Dr. R. K. Sharma and Dr. Parteek Kumar. This report has not been submitted to any other institution for award of any other degree.

Rupinderdeep Kaur
Regn. No. 951203008

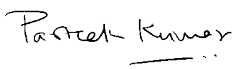
Place: Patiala
Date: 1.11.2021

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Verified by:



Dr. R. K. Sharma
(Professor)



1.11.2021

Dr. Parteek Kumar
(Professor)

Abstract

Speech is the most natural means of communication between humans. It is one of the first skills that we learn. Babies quickly learn how to react to the voice of their mother and they even more quickly learn to produce noise when they are in need. Speech has always been an important way of communication. Even before writing, the spoken words were used to pass the knowledge.

Despite all our novel ways of communication, such as e-mail and chat, speech is still considered to be the best means of communication. So, it is only logical that machine interface designers in their quest for a natural man-machine interface have turned to automatic speech recognition and speech production as one of the most promising interfaces. The system which converts speech signal to text is termed as Automatic Speech Recognition (ASR) system. Phonetic Engine (PE) is the first stage of ASR and it converts speech signal to phonetic symbols. ASR system does this process by capturing speech waveform, extracting the relevant features, capturing the message and reproducing it as text.

The main motivation behind this work is to develop a PE for Punjabi language and explore the possibility of improving its performance by incorporating prosody. Prosody refers to the collection of characteristics that lend naturalness to speech. PE is a transformation tool which utilizes the acoustic phonetic details present in an input speech signal to decompose it into a symbolic form. PE develops a sequence of symbols without considering any language constraints in the form of lexical, syntactic and higher level knowledge source. The choice of symbols should be such that it can capture all the phonetic variations in the speech.

In this research work, a PE is designed and implemented for continuous speech of an Indian language named as Punjabi. Punjabi is a highly prosodic language and not much work has been done in this direction on this language. As a first step towards the development of PE, 24.5 hours of data has been collected in three different modes, namely, read speech, lecture speech and conversational speech. The 10 hours of collected data is then manually transcribed using International Phonetic

Alphabet (IPA) chart. The architecture of the PE includes three phases: data preparation, system training and system testing. Initially, 49 symbols were selected by carefully analysing the symbol frequency in IPA transcription and data files have been prepared to train the system accordingly. The prepared data files and speech files have then been used for modeling and feature extraction processes. In the development of PE, Mel-Frequency Cepstral Coefficients (MFCCs) have been used as a feature extraction technique and Hidden Markov Model (HMM) as a classifier. The PE has been developed using HMM ToolKit (HTK).

The performance of PE has been evaluated using three different approaches: (i) By increasing the amount of data from 3 hours to 5 hours, (ii) By decreasing the number of symbols from 49 to 29, and (iii) By increasing MFCC dimensions from 12 to 36. An accuracy of 72.3% has been achieved in this work when 5 hours data with 29 symbols and 12 MFCCs was employed.

The speech data collected in read speech mode has further been used to design and implement a text-independent speaker classification, since, it is one of the popular biometric identification techniques, which establishes the speaker's identity by considering the speech of the person.

Many speaker classification techniques have been designed and implemented so far to efficiently recognize the speaker. From the existing review, it has found that the existing speaker classification techniques suffer from the over-fitting and the parameter tuning issues. An efficient tuning of machine learning techniques can improve the classification accuracy of speaker classification. Therefore, to overcome the over-fitting issue, initially, in this thesis, a novel Ensemble-based Quantum Neural Network (EQNN) technique has been designed. It works on ensembling of novel data splitting strategies. Quantum Neural Network (QNN) has been implemented in MATLAB for the dataset of 7 speakers with 30 samples of read speech from each speaker. QNN has been trained and tested with different data splitting strategies. Along with this, results of previous strategy has been ensembled with the training of next strategy. All the experiments have been repeated 30 times.

For comparison of results, we have implemented four base classifiers, namely, Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM) and Artificial Neural Networks (ANNs) with same dataset. Extensive experiments have been carried out by considering EQNN and the base classifiers. The performance of all the techniques has been evaluated using four performance metrics, namely, accuracy, F-measure, specificity and sensitivity. It has been observed that EQNN outperforms existing speaker classification techniques in terms of all the performance metrics.

However, EQNN based speaker classification technique suffers from the parameter

tuning issue and still there is a chance of over-fitting. To overcome this issue, finally, a Crossover based Particle Swarm Optimization with Support Vector Machine (CPSOSVM) has been designed and implemented in this work using MATLAB. In CPSOSVM, Particle Swarm Optimization (PSO) has been used to tune the parameters of SVM. The crossover operator has been applied on PSO as it has an ability to overcome the issue of getting stuck in local optima with the standard PSO. Thereafter, CPSOSVM and the competitive machine learning techniques have been used to classify the speakers. Finally, the comparisons have been drawn with the competitive machine learning models and CPSOSVM by considering the same performance metrics as we did for EQNN. It has been observed that CPSOSVM has performed better in all the performance metrics when compared with EQNN and other base classifiers.

To

my supervisors

Dr. R. K. Sharma and **Dr. Parteek Kumar**

(for their guidance, support and encouragement)

and

my sister

Dr. Rattan Deep Kaur Virk

(for her advise, patience and faith,

because she always understood)

Acknowledgement

Writing an acknowledgement is the most emotional part of writing thesis. I would like to take an opportunity to pay gratitude to all those who matter in my life and have helped in achieving my goals and aspirations. Pursuing Ph.D. is just like climbing a high peak, step by step, accompanied by hardships, frustration, encouragement, trust and with so many people's kind help. It was, in fact, a teamwork that got me here. Though it will not be enough to express my gratitude in words to all those people who helped me. I would still like to give my thanks to all these people.

First of all, I wish to acknowledge the benevolence of the Almighty, who gave me strength, courage, and patience to overcome all obstacles. With a profound sense of gratitude and heartiest regard, I express my sincere feelings of indebtedness to my supervisors **Dr. R. K. Sharma** and **Dr. Parteek Kumar** for their valuable guidance, motivation, encouragement, moral support, and invaluable co-operation. The generous and encouraging attitude with which they resolved all my problems will always have a shadow on my character. I deeply admire the delightful atmosphere for learning provided by them that made this thesis possible. It has been a great pleasure and experience to work under their guidance.

I am grateful to the Head of the Computer Science and Engineering Department (CSED), **Dr. Maninder Singh**, who made my study a knowledgeable experience during my stay in the department. I am thankful to my Doctoral committee members, **Dr. Seema Bawa**, **Dr. A. K. Verma** and **Dr. M. D. Singh** for their constructive comments and regularly ensuring the progress of my research work. My deep regards to **Dr. Prakash Gopalan**, Director, Thapar Institute of Engineering and Technology (TIET) for giving me access to facilities, which have been immensely helpful for the completion of my work. I am much obliged to the Dean (RSP), **Dr. Rafat Siddique** and the Management of TIET, who provided me all the necessary resources and encouraged to produce results. I sincerely thank **Dr. Rinkle Aggarwal** and **Dr. Sushma Jain**, Ph.D coordinators, CSED, for all their co-operation and motivation. I sincerely thank the faculty and support staff of CSED for their con-

stant motivation.

I offer my deepest gratitude to my father, **S. Tara Singh Aulakh**, whose dream I have lived throughout these years of my Ph.D. I am also thankful to my mother **Mrs. Lakhwinderjit Kaur**, my two sisters and my brother for their love, encouragement, motivation and confidence in me. I am blessed to have nephews, who always understood my frustration, and hardships. They loved me and helped me to look at the things from a lighter side. I am blessed to have powerful mentor in the form of my brother-in-law, **Dr. Jagvinder Singh Virk**. I thank him for inspiring me to always strive for achieving bigger things in life.

I also acknowledge the cooperation and encouragement extended to me by my friends, especially **Dr. Divya Pandove**, for always suggesting me a way out for my problems and being the biggest pillar of support throughout this long journey and **Dr. Karamjit Kaur**, for pushing me in the right direction and taking care of me like an elder sister. I am also obliged to **Dr. Sanmeet Kaur**, who, despite of all her struggles, gave me strength and positive environment during my thesis writing. I also thank **Dr. Anupam Sharma** for debates, dinners, game nights, general help and friendship. All these are greatly appreciated.

Finally, I pay regards to one and all who knowingly or unknowingly supported me during this journey of knowledge. To my friends and relatives scattered around the globe, thank you for your thoughts, well-wishes/prayers, phone calls, e-mails, texts, visits, editing advice, and being there whenever I needed anyone of them.

Rupinderdeep Kaur

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

Abbreviation	Full Form
ADC	Arabic Digit Corpus
AI	Artificial Intelligence
ANN	Artificial Neural Network
APT	Automatic Phonetic Transcription
CAC	Command And Control
CD-HMMs	Continuous Density Hidden Markov Models
CML	Conditional Maximum Likelihood
CPSO	Crossover based Particle Swarm Optimization
CPSOSVM	Crossover based PSO with SVM
DBNs	Deep Belief Networks
DCF	Decision Cost Function
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DNA	Deep Neural Architecture
DNN-HMMs	Deep Neural Network HMMs
DT	Decision Tree
DTW	Dynamic Time wrapping
EAs	Evolutionary Algorithms
EDLF	Eigen Decomposition Like Factorization
EER	Equal Error Rate
EM	Expectation-Maximization
EQNN	Ensemble-based Quantum Neural Network
FFNNs	Feed Forward Neural Networks
GA	Genetic Algorithm
GMMs	Gaussian Mixture Models
GUI	Graphical User Interface

GUMI	GMM-UBM Mean Interval
HMM	Hidden Markov Model
HRI	Human Robot Interaction
HTK	HMM ToolKit
IPA	International Phonetic Alphabet
LDA	Linear Discriminant Analysis
LFPCs	Log Frequency Power Coefficients
LPC	Linear Predictive Coding
LPCCs	Linear Prediction Cepstral Coefficients
LVCSR	Large Vocabulary Continuous Speech Recognition
KL	Kullback-Leibler
MCE	Minimum Classification Error
MFC	Mel-Frequency Cepstrum
MFCCs	Mel-Frequency Cepstral Coefficients
ML	Machine Learning
MLE	Maximum Likelihood Estimation
MLF	Master Label File
MLLR	Maximum Likelihood Linear Regression
MLP	Multi Layer Perceptions
MLP-HMMs	Multi-Layer Perceptrons HMMs
MMF	Master Macro File
MMIE	Maximum Mutual Information Estimation
MSE	Mean Squared Error
NBS	National Bureau of Standards
NEM	Noisy Expectation- Maximization
PCA	Principle Component Analysis
PCP	Pitch Class Profile
PDF	Probability Density Function
PE	Phonetic Engine
PLDA	Probabilistic LDA
PLPCC	Perceptual Linear Prediction Cepstral Coefficients
POS	Part of Speech
PPRT	Phonetic and Prosodically Rich Transcribed
PSO	Particle Swarm Optimization
QNN	Quantum Neural Network
RASTE	Relative Spectral Transform
RBM	Restricted Boltzmann Machine

RF	Random Forest
STM	Spectral Transition Measure
SVM	Support Vector Machine
VEP	Vowel End (or offset) Point
VOP	Vowel Onset Point
VQ	Vector Quantization
WER	Word Error Rate
WRR	Word Recognition Rate

Chapter 1

Introduction

This chapter provides an introduction to the topic of this thesis, along with the need and the applications of the proposed work. The basic terminologies to understand the concept of proposed work have also been explained. This chapter also explains briefly the technical challenges faced during the accomplishment of the proposed work. The research methodology, in the form of techniques and methods used and basic architecture, to achieve the formulated objectives have also been described. Research questions or gaps in existing literature that helped us to formulate our problem statement are also presented in this chapter. Based on the gaps identified in the literature, the objectives of the thesis were formed. Contributions and organization of the thesis are also presented in the chapter. This chapter summarizes the complete thesis enabling the reader to quickly understand the purpose, significance, objectives and the contribution of the thesis in the area of prosody based Phonetic Engine and speaker classification for the Punjabi language.

1.1 Introduction to the Proposed Work

The proposed work in this thesis, "Prosody Based Phonetic Engine and Speaker Classification for Punjabi Language" aims to focus on the development of Phonetic Engine (PE), and to propose a speaker classification framework. The structure of spoken Punjabi language and its prosodic features have primarily been considered in this work. Following two sub-sections give a brief introduction to PE and speaker classification.

1.1.1 Prosody Based PE

To utilize machine's capabilities, it is necessary for human beings to communicate with it efficiently. Speech is a convenient way of communication for human beings, and it has the potential to act as an effective mode of communication between man and machine. A computer is empowered through the Automatic Speech Recognition (ASR) technique to identify the words which are spoken into a microphone or telephone and speech is automatically recognised.

PE is considered as the first step towards ASR. It takes speech file as an input and transforms the speech file into International Phonetic Alphabets (IPA). In this work, an attempt has been made to build a PE for speaker-independent continuous speech, *i.e.*, read speech for Punjabi language. Punjabi is a tonal language and belongs to the Indo-Aryan family of languages. Besides its tonal nature, variations arise from the emotional stress in pronunciation, and this changes the sense of speech. Tonal features are phonetic and segmental in nature, and correspond to prosody. To capture prosodic features of Punjabi language and to get rich phonetic and prosodic transcription, IPA chart (Passy, 2005) has been used to derive transcription of spoken utterances. A statistical approach: Hidden Markov Model (HMM) has been used to develop the proposed system. HMM ToolKit (HTK), implementation of HMMs, explained in Young (2001), has been used to conduct the experiments in this work.

As the first step for the development of PE for Punjabi language, data has been collected in three different modes: (i) Read speech mode, (ii) Lecture speech mode, and (iii) Conversational speech mode from different regions of Punjab and different background environments. Data has been collected in natural noisy environment as well as in isolated room environment. The collected data is not domain-specific and covers a good amount of vocabulary of the language. The collected data is then transcribed using IPA chart. There are symbols for sounds of consonants, vowels, diacritics and few additional symbols in the IPA chart. Phonetic transcription is carefully analysed in order to choose the number of symbols to train the engine. In this work, initially, 49 symbols were selected based on the frequency of the occurrence of IPA symbols, and then few symbols were merged to obtain 29 symbols. The architecture of PE consists of three parts, namely, data preparation, PE training and PE testing. Based on the selected number of symbols, data files are prepared, which are further used for system training. Mel-Frequency Cepstral Coefficients (MFCCs) are extracted, and HMMs are created using HTK. The engine is tested by varying three parameters: (i) Amount of data, (ii) Number of symbols and (iii) Dimensions of MFCC features.

1.1.2 Machine Learning Based Speaker Classification

Speaker classification is the technique of identifying a person using statistical features obtained from speech signal. Many speaker classification techniques have been designed and implemented so far to efficiently recognize the speaker. From the existing review, it has been found that the existing speaker classification techniques suffer from the over-fitting (Rafiq *et al.*, 2001; Panchal *et al.*, 2011) and parameter tuning issues (Keerthi, 2002; Friedrichs and Igel, 2005).

To overcome over-fitting issue, we designed an Ensemble-based Quantum Neural Network (EQNN) technique. In this technique, Quantum Neural Network (QNN) has been used to train the model with different data splitting strategies. Errors from one strategy has then been used with second strategy to make the model learn from its mistakes. In this way, ensembling of four data splitting strategies have been performed with QNN to achieve the results for speaker classification. Extensive experiments have been carried out using the proposed technique and existing competitive machine learning based speaker classification techniques on speech data. Although it has been observed that the proposed EQNN performs better than the existing techniques, but it still suffers from over-fitting issue.

In order to reduce the over-fitting, we have experimented with Support Vector Machine (SVM) classification strategy. A crossover based particle swarm optimization is proposed to tune the parameters of SVM. The crossover operator has an ability to overcome the issue of getting stuck in local optima when used with the standard particle swarm optimization. The proposed and the competitive speaker classification techniques are tested on the read speech data of Punjabi persons. The comparative analysis of the proposed technique reveals that it outperforms existing techniques in terms of accuracy, F-measure, specificity, and sensitivity as obtained from MATLAB implementation.

1.2 Need of the Proposed Work

There are several defining characteristics based on which ASR systems are developed. Dynamic time wrapping (Rabiner and Schmidt, 1980), acoustic template matching, vector quantization (Rabiner, 1989) and probabilistic models (Lee *et al.*, 1989; Lee and Hon, 1989) are the supporting techniques used for ASR Systems. Most of the ASR systems use limited vocabulary (Woodland *et al.*, 1995), and are speaker dependent (Azmi and Tolba, 2008). For limited vocabulary ASR systems, the data collected is mostly in an isolated room environment and not in the natural

noisy environment. The ASR systems, which are speaker independent, are highly domain specific (Sandipan Mandal *et al.*, 2010).

Several attempts have been made in the field of isolated words speech recognition (Ravinder, 2010; Kumar and Singh, 2011). Al-Qatab and Ainon (2010) and Kumar *et al.* (2012) have worked on connected words speech recognition. A very limited work has been reported in literature where ASR systems are exploring prosody based knowledge from the speech database. Only by developing the system such as PE, one can make this exploration possible through which spoken utterances can be converted to a set of symbols and message conveyed by spoken utterance can still be kept intact in symbols.

There are very few attempts for building ASR systems for the Punjabi language that deal even with isolated words and limited vocabulary (Ravinder, 2010; Kumar and Singh, 2011). So, an attempt has been made to develop a speaker independent PE wherein vocabulary is not limited to any domain and speakers are also not fixed.

Neural networks and deep learning models have extensively been used for speech as well as for processing of language and have been specifically successful in the cases of successful sequencing of the task of production such as automatic speech classification (Hiremath and Hiremath, 2017) and modeling of language (Garimella *et al.*, 2012). However, neural network and deep learning variants are prone to get trapped in local minimum value and may suffer from poor convergence speed. Also, neural network-based models are time consuming in nature (Zhang, 2017). Additionally, majority of existing techniques suffer from the over-fitting issue (Rafiq *et al.*, 2001; Panchal *et al.*, 2011).

To improve the performance of speaker classification systems, many techniques have been proposed in the literature. But, speaker classification is still an ill-posed problem. Decision tree based learning techniques such as Random forest (Gokgoz and Subasi, 2015) and J48 (Chen and Howard, 2016) are found to be effective to improve the prediction rate of text independent speaker classification techniques (Mistry *et al.*, 2016). However, decision tree based machine learning techniques provide minimum information on the relationship between predictors and their respective response (Shaikhina *et al.*, 2017). Also, these models are not very effective on the datasets with low sample sizes.

To overcome these issues, SVM variants have been designed and implemented (Zhang and Chen, 2016). It is the principle of structural risk minimisation on which SVM is based. It is able to avoid the occurrence of local optimum and the complex problems such as limited sample can be well handled by it along with high dimensional and non-linear data (Sukawattanavijit *et al.*, 2017). However, the per-

formance of SVM variants depends upon its kernel parameters and penalty factor (Shen *et al.*, 2017). Therefore, an efficient selection of appropriate parameters optimization techniques is required.

Most of the present speaker classification research works has been conducted on the regular speech. Regular speech is the speech which is intensionally produced by the people and clear linguistic contents are present in it. Many algorithms like statistical model approach and the neural model approach are proposed so as to perform speaker classification in this form of speech (Shi *et al.*, 2017). A good information about the speaker can be obtained from the regular speech because of the clear speech hence making the results of speaker identification more acceptable (Saleem and Khattak, 2018).

In spite of considerable progress attained in the field of regular speech, there is still very limited research on the non-linguistic part of speech signals. For example, one may, during the process of talking to others, also do activity of coughing and laughing or may *'tsk-tsk'*, showing the indication of disapproval of some notion or may do *'hmm'* to show annoyance or doubt. Different personal habits produced such event and there is very little linguistic information involved in such notions but they do convey some sort of information about the speaker as well (Xi *et al.*, 2017).

1.3 Applications of the Proposed Work

Speech signal mainly consists of two types of information, namely, linguistic information (information related to spoken language) and speaker information. Once this information is extracted from the speech signal, this can be explored further in many ways. In this work, linguistic information is used to develop the PE and the speaker classification has explicitly been carried out. As such, the PE developed and the speaker classification can possibly be used in the following applications.

1.3.1 Speech Recognition Applications

As mentioned earlier, PE is the first step towards the development of an ASR system. Thus it can ultimately lead to an efficient ASR system. It can be used in applications like route navigation, in health care for record keeping and medical documentation, in training air traffic controllers and for daily education system. The additional information captured by PE in the form of prosodic features can also be useful in other applications like language recognition and language translation, speaker recognition and verification, query-by-example system and speech search engine.

1.3.2 Speaker Classification Applications

The use of speaker classification can be made in variety of applications where information about speaker plays an important role. User authentication by one's speech is an important application. An utterance from a speaker has to be analysed in speaker authentication and comparison with speech models of known speakers be made. Identification of the speaker as authorised and un-authorised is based on the best matching with the input utterance. The speaker is marked as authorised if the match is good enough or above a threshold. Further, speaker classification can be used in surveillance and forensic tasks.

1.4 Basic Terminology

Throughout this work, examples from the Punjabi language have been used. To make it more understandable and clear to the reader, a format for the Punjabi examples has been followed, which comprises of Punjabi language word or letter or sentence, English transliteration, IPA transcription and corresponding translation in English. This order has been followed throughout this document. For example, ਰਾਤ / raat / rát / night, where 'ਰਾਤ' is a Punjabi word, 'raat' is its English transliteration, 'rát' is the IPA transcription and 'night' is its English translation. The basic terms used in PE and classification include phonology, phonemics, phonetics and prosody. These are briefly explained in following sub-sections.

1.4.1 Phonology

Phonology is that branch of linguistics which deals with the systematic organisation of sounds of language. The traditional focus of phonology is on the study of systems of phonemes in languages. A linguistic analysis may also be covered by it either at a level beneath the word or at all levels of language. It is that point at which sound is considered to be structured to convey linguistic meaning (Ladefoged, 1982). This phoneme system and linguistic analysis is a key process for developing PE system for the language under consideration.

1.4.2 Phonemics

The basic unit of a language's phonology is called as phoneme. It is combined to make meaningful units such as words or morphemes with other phonemes. Description of smallest contrastive linguistic unit can be given to the phonemes which may

bring a change of meaning (Ladefoged, 1982). For example, the words ਬਾਪ / baap / bāp / father, ਪਾਪ / paap / pāp / sin and ਮਾਪ / maap / māp / measure, can be obtained by replacing the phoneme /b/ by the phonemes /p/ and /m/. These words are called minimal pairs which differ in meaning through a contrast of a single phoneme. In order to build an efficient PE, a deep study of these phonemes is very necessary.

1.4.3 Phonetics

Phonetics is the study and classification of speech sounds (Ladefoged, 1982). Phonetic transcription of speech uses the IPA as its basis. IPA uses the Latin alphabets for transcription. Most features of speech can be transcribed by it, for example, consonants, vowels and suprasegmental features. A corresponding symbol is assigned to every documented phoneme which is available within the known language in the world. Figure 1.1 contains the IPA chart (Passy, 2005). Following example shows how a sentence from the Punjabi speech can be transcribed using IPA: ਪੁਰਾਣੀ ਕਹਾਵਤ ਹੈ ਕਿ ਹਾਰੀਏ ਨਾ ਹਿਮੰਤ ਵਿਸਾਰੀਏ ਨਾ ਰਾਮ / purani kahawat hai ki hariye na himat visariye na ram / pʊɾáːɳikəháːvə_thæ̀_keː_háːɾienáːhɪmə_tə̆_bɪsaːɾɨynáɾàːm / The old saying is that neither lose courage nor forget God. In the IPA transcription performed here, '_' has been used to depict the silence region or phrase breaks in the speech. Length of these silence regions is also important and we have considered this in our implementation. It is worth mentioning here that phonetics remove the language barrier and converts the set of phonemes into corresponding phonetic symbols. Phonetics plays major roles in ASR system because it not just concatenates the symbols to form a word, as done in phonology, but also represents the symbols for supra-segmental and tonal features of human sounds. Since, Punjabi is highly a tonal language, so after phonology, conversion of phonemes to phonetics is very necessary for PE to be natural.

1.4.4 Prosody

Prosody is the intonation, stress, tone and rhythm present in speech. Various features of a speaker or utterance may be reflected by prosody such as speaker's emotional state or the form of the utterance like question or command *etc.*, the presence of focus, contrast and emphasis. There is supra-segmentation in prosodic features, *i.e.*, these features occur in some higher level of utterance and are not confined to any one segment. Phonetic cues mark prosodic units like a coherent pitch contour or regular decrease in pitch. There can be lengthening of vowels over the duration of the unit till such time, the pitch and the speech get reset to restart the next unit.

Inhaling and exhaling the breath appeared to take place only at those boundaries where the prosody is reset (Ladefoged, 1982).

Intonation

The pitch defines the sensation of the 'altitude' of sound (Nooteboom, 1997). It is the correlate of fundamental frequency (F_0) determined by the rate of vibration of the vocal chords. The intonation in an utterance is defined by the ensemble of pitch variations (Hart *et al.*, 1990). The range of F_0 for individual speakers depends on the length and mass of the vocal chords. It is generally between 80-200 Hz for males and between 180- 400 Hz for females in the context of conversational speech. A speaker can produce F_0 rise and falls within this range. Monological patterns of the constituent words determine either rising or falling due to the direction of F_0 change (Mary, 2011).

Stress

When certain syllables in a word or certain words in a phrase are emphasized, it is referred to as stress in linguistics. Stress is an important feature of a number of languages and Punjabi language is one of them. Stress specifies the strength (in some sense) of a syllable in a word and this ultimately corresponds to a structural and linguistic property of a word. The acoustic and perceptual characteristics by which the stressed syllable is distinguished from the surrounding un-stressed syllable has always been an important topic of research in phonetics. In its further refined form, it tells how an un-stressed realisation of the same syllable is different from its stressed realisation (Mary, 2011).

Tone

As specified in intonation and stress, the pitch of spoken words tends to rise or fall. This is due to the tonal nature of Punjabi language. This pattern can be seen in sentences and also in individual words. Within a word, tone changes from unstressed to stressed syllable and vice versa. There are three tones in the Punjabi language, namely, high-tone, mid-tone and low-tone (Karamat, 2001). For example, ਚਾਹ / chah / ʧá / tea, depicts high-tone; ਚਾਅ / chaa / ʧɑː / happiness, depicts mid-tone and ਝਾਅ / jhaa / ʧà / peep depicts low-tone.

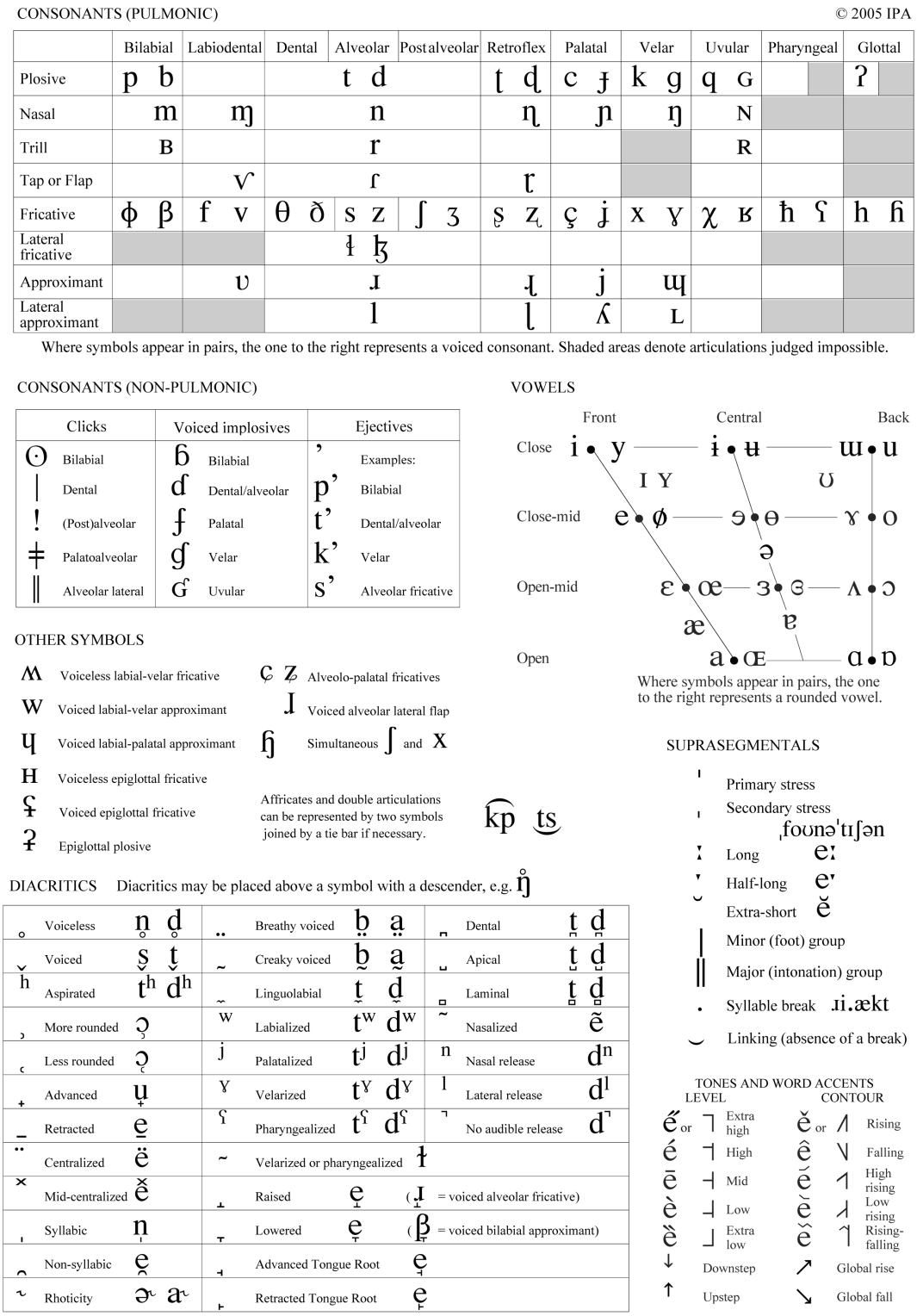


Figure 1.1: The IPA chart (Passy, 2005)

Rhythm

Rhythm refers to the organization of timing in speech, and it has been shown to be different across languages (Nooteboom, 1997). Work on linguistic rhythm has strongly correlated the differences in rhythmic percept found between languages with a set of language-specific phonetic and phonological properties, of which the two most often cited are syllabic structure and vowel reduction. Similar rhythmic patterns are shown to be recognised by infants in the absence of knowledge of linguistics. One cannot separate the different patterns contributing to durational variations (Mary, 2018).

1.5 Challenges in the Development of Phonetic Engine and Speaker Classification

Firstly, handling different speaking styles, dialects, accents and varied data collection environments are the major challenges for the development of PE. Even for a given speaker and given content of speech, the waveforms can considerably be different at different times and thus giving different transcriptions. Punjabi language throws its own challenges because of its prosodic nature. Sentence level modelling is another challenge, and owing to this fact, the sentences have been converted into the chunks of 4 to 6 second duration in this work. Extracting acoustic phonetic description of speech is also a challenge.

There has been a consistent engineering challenge in automatic segmentation of speech signals. Although a lot of work has been done in supervised and unsupervised techniques, there still lies a challenge to equate the manually labelled segments. Extracting acoustic features for spotting trills in continuous speech is also a challenge. As the spectral characteristics are changing continuously during the production of trills, and also due to vowel context, it is a challenge to represent them for PE and speaker classification.

It has been seen that transcription of one speech file is different when transcribed by different persons. This is due to individual's perception of tone and pitch. Modulating these differences to have a uniform training set is a challenge for building an efficient PE.

There are some non-regular and non-linguistic events that occur in our conversations. These are called as 'trivial events'. Cough, laugh, *'ahem'* (short cough made by somebody who is trying to get attention) *etc.*, are included in typical trivial events. These trivial events are very important when it comes to speaker classifi-

cation as a person can be recognised by us through his style of laughing if we are familiar with it. Modeling and recognising such events within the speech is also a challenge.

1.6 Techniques and Methods Used in Proposed Work

Machine Learning (ML) is a subfield of Artificial Intelligence (AI) and it deals with the development of techniques and methods that enable the computer to learn (El-Alfy *et al.*, 2015). In other words, ML includes the development of algorithms which enable the machine to learn and perform tasks and activities. Over the period of time, many techniques and methodologies have been developed for ML tasks. This section gives a brief introduction of the techniques and methods used in this work.

1.6.1 Mel-Frequency Cepstral Coefficients

As per the human perceptional characteristics, MFCCs are well-advised as the best approximation (Lippmann, 1997). The best representation of the short-term power spectrum of speech is given by Mel-Frequency Cepstrum (MFC), and Discrete Cosine Transform (DCT) is the basis of the same. It has been noted from the literature that human ears can act as a filter, and they only focus on certain frequency components. On the frequency axis, there is a non-uniform space between these filters. The low frequency region have more filters whereas high frequency regions have less number of filters. These non-uniformly spaced filters are called as Mel-filters and MFCCs are the cepstral coefficients obtained on Mel-spectrum.
In the process of extracting MFCC features, speech signal is first analysed, over the short analysis window. A spectrum is obtained for each short window of speech using Discrete Fourier Transform (DFT). Then, Mel-spectrum is obtained by passing this spectrum through Mel-filters.Then logarithmic transformation and DCT is done. Cepstral analysis is performed on Mel-spectrum is performed to obtain features that are similar to cepstrum, and they are called as MFCCs. Another Cepstral analysis like filtering is performed in the frequency domain to separate the vocal tract and excitation characteristics of speech.
From each short analysis window, 12 MFCCs and a normalized energy parameter is obtained. Speech signal changes dynamically so the differential and acceleration coefficients are also obtained. This gives us another 12 delta coefficients (representing differential coefficients) and 12 delta-delta coefficients (representing accel-

eration coefficients). Along with these sets of coefficients, energy feature is also obtained. Ultimately, the 39-dimensional feature vector for each speech frame is obtained from the energy parameter and the estimation of first and second derivatives of MFCCs. One can, however, vary the dimension of this feature vector.

1.6.2 Hidden Markov Model

A stochastic process refers to the processing of a set of random variables which tend to change with time or at different positions within a given sequence. HMMs have extensively been used in speech processing because speech can be characterised as a set of random variables and parameters of these stochastic events can be estimated in a precise manner. HMM is an embedded stochastic process with a hidden underlying stochastic process, and an observable set of stochastic processes that produce a series of observations.

Hidden and observable states of HMM are explained in the following example. Suppose that two balls are there in a bucket, white and black. A person named Jack picks one ball at random and tells the colour of that picked ball to another person named Jill, who is unaware of the actual event, *i.e.*, he is only writing down the result without having any idea about what this result corresponds to. So, in this case, the event of selecting a ball randomly from the bucket is a hidden state for Jill, whereas the result (white or black) is an observable state. In a series of such events, the sequence of observable states will look similar to (1.1).

$$O = O_1 \;\; O_2 \;\; O_3 \;\; ... \;\; O_t = B \;\; B \;\; W \;\; B \;\; W \;\; B \;\; ... \;\; B \tag{1.1}$$

In order to build HMMs for such a situation, there are mainly two problems: deciding on what these states corresponds to and how many states are there in total. In the simple case of above stated example, there can be a two state model where each state corresponds to the colour of the ball. In another situation where the number of balls are 4 with 2 white balls and 2 black balls, the states of the model will remain two and the observation sequence will also look similar to (1.1). The number of states will increase with the increase in number of colour of balls. The issue in these cases is to decide the best probability of the output. This is resolved in HMM with the use of unknown parameters, which represents the bias in choosing the next state in a sequence. The more the number states, the larger the size of unknown parameters thus making the HMM model capable of modeling the series of events efficiently (Rabiner, 1989).

In order to build these HMM transitions for a speech signal, which can generate

sequence of symbols, we need parameters of speech and their corresponding transcription. After finding the parameters, HMM transitions are assigned to these parameters, and the process is called the training of HMM. For training of HMM, Baum-Welch algorithm is used which is a well-known forward backward algorithm (Tu, 2015). For decoding the speech signal, the Viterbi and forward algorithms are used. These algorithms help in solving the HMM's problem (problem of decoding and training). The toolkit which is used for creating and shaping HMMs in this work is the HTK. In early stages, HTK was designed for creating speech processing tools based upon HMM. For its execution, it mainly required the UNIX Operating System and contained a set of library modules which are written using ANSI C language. Thereafter, HTK has been developed gradually. Nowadays, it can run on almost all the Operating System environments. There are many versions of HTK available these days. The latest stable version available of HTK is HTK 3.4.1. HTK version 3.5 beta is the most recent release. In this study, HTK 3.4.1 has been used for the development of PE for Punjabi language.

1.6.3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is an evolutionary computational algorithm which is dependent on the selection of intelligence of the swarm. It is a technique, proposed by Kennedy and Eberhart (1995), based on stochastic optimization of population. The PSO is initiated by the process of randomly starting a search for potential solution followed by iterative search for the optimisation.
The optimal position is found in the PSO algorithm by adopting the best particles. A comparison with Evolutionary Algorithms (EAs) suggests that PSO has profound intelligent background which can be performed more easily. Due to the PSO's associated advantages, it is considered to be suitable for research, and also in real life evolutionary computing problems. PSO is also used to solve global optimization problems. It can be said in other words that PSO is associated with AI particularly relating to swarming theories and social behaviour simulations, and also to EAs.
PSO can be easily implemented as it requires low CPU speed and low memory (Khan, 1998), making it computationally inexpensive (Eberhart, 2007). Moreover, no information about the objective functions is required by it such as gradient but only needs its value. PSO was proved to be an efficient method for solving many global optimisation problems. It does not face the difficulties which are faced by other EAs. In this work, a crossover based PSO has been used to tune the parameters of SVM. This has been done to propose an efficient speaker classification technique.

In the general, the crossover is taken between each particle's individual best position. After the crossover, the fitness of the individual best position is compared with that of the two offspring, and the best one is taken as the new individual best position. The crossover can help the particles jump out of the local optimization by sharing the others' information.

1.7 Basic Steps for the Proposed Work

In order to develop an efficient PE and perform speaker classification, followings steps are required to be performed. Figure 1.2 contains a flow chart of these steps.

1.7.1 Data Collection

The first and the foremost requirement to develop the PE and to perform speaker classification is to collect the data for Punjabi language. For this work, data has been collected from native Punjabi speakers of different age groups, genders and from locations to cover all the dialectal aspects of the Punjabi language. Details of the data collection process are presented in Chapter 3. The metadata about the speakers has also been maintained.

1.7.2 Features of Punjabi Language

In order to perform correct and effective linguistic analysis as well as prosody prediction, study of phonological and prosodic features of the Punjabi language is required. This helps in understanding the basics of the language. Specifically for the Punjabi language, this study is required because of the tonal nature of the language. The detailed study of these features is presented in Chapter 3.

1.7.3 Data Processing

Data is processed in two phases. In the first phase, *i.e*, pre-processing, speech files are divided in small chunks of 4 to 6 second duration and each file is given unique name and all the chunks are stored to form a speech database. The chunks where there is no sound or where there is only background noise are deleted from the database.

In the second phase, *i.e*, normalization, the frequency of speech signal is modified to a standard level. Normalization can be performed to maintain uniformity in the recorded speech.

1.7.4 Linguistic Analysis

Scientific analysis of a language sample is termed as linguistic analysis. One can use linguistic analysis for describing unconscious rules and processes which are used by speakers of a language for creating spoken or written language. It is of high usefulness for those who are anxious to learn a language or who desire to translate one language to the other. In this work, this has been useful in processing of speech files for the development of PE. For PE development, linguistic analysis is done by speech tagging, also by prosodic phrase break marking. Work done in this thesis, with respect to linguistic analysis, is presented in Chapter 3.

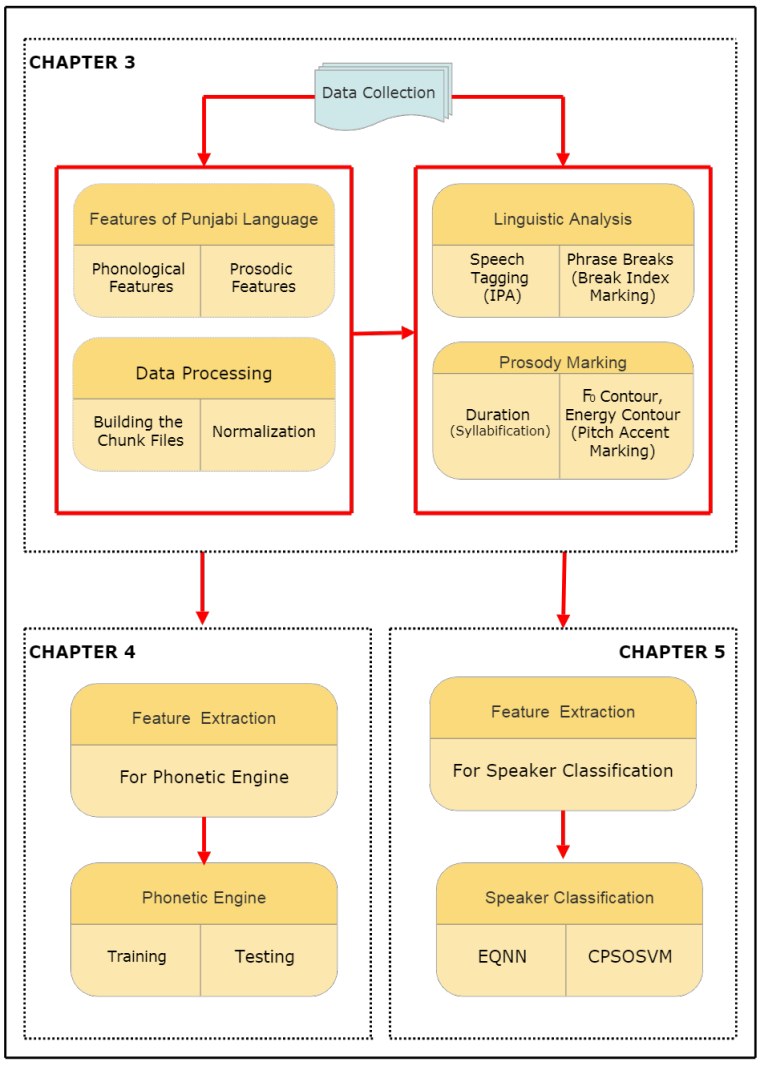


Figure 1.2: Basic steps in the proposed work

Speech Tagging

Speech tagging is the process of assigning the tags a spoken utterance. Tagging is the task of assigning corresponding symbols to a speech signal. For this work, IPA chart has been used to tag the speech data. It is performed by marking IPA symbols corresponding to the speech files. This is important for development of PE because training of PE has to be performed with these IPA transcriptions.

Phrase Breaks

Phrase breaks incorporate the silence region in a speech. For example, consider the Punjabi phrase: ਮੰਤਰੀ ਜੀ ਨੇ ਕਿਹਾ ਕਿ ਅੱਜ ਸਾਡੇ ਦੇਸ਼ / mantri ji ne keha ke ajj saade desh / mənt̪rī jī nē kəhá: kɨ adʒʤ̆ sa:ḍ̪ē dēsh / Minister said that today in our country. The depiction of phrase break in this phrase is given in (1.2) and (1.3).

$$\text{ਮੰਤਰੀ ਜੀ ਨੇ ਕਿਹਾ ਕਿ ਅੱਜ ਸਾਡੇ ਦੇਸ਼} \tag{1.2}$$

$$\text{ਮੰਤਰੀ ਜੀ ਨੇ ਕਿਹਾ ਕਿ}[pause]\text{ਅੱਜ ਸਾਡੇ ਦੇਸ਼} \tag{1.3}$$

Here in (1.2), ਕਿ is the preposition marked by speech tagging and (1.3) tells about the phrase break, *i.e.*, while converting this speech to transcription, there will be a small pause in this region (Singh and Lehal, 2011). The break index marking performed in this work in order to mark the phrase breaks is presented in Chapter 3.

1.7.5 Prosodic Marking

Words are often ambiguous in speech and can be tackled by prosody prediction. Prosodic prediction analysis deals with modeling and generation of syllable boundaries from duration and marking of pitch accent from intonation contours for the given speech. Prosody is not present in text which makes it inherently difficult. To predict appropriate duration and intonation, the input speech needs to be analysed. This can be performed by a variety of algorithms including simple rules, example-based techniques and machine learning algorithms. The prosody can be predicted by duration, F_0 contour and energy contour. These approaches are described in subsequent sub-sections.

Duration

Analysis of syllable's duration is done with regards to positional and contextual factors. Syllables are categorized into groups based on the size of the word and

position of the word in the utterance for the purpose of detailed duration analysis. On each category, the analysis is performed separately. Analysis of duration leads us to observe that the duration of sound units is dependent on various factors at different levels and precise rules are required to be derived for accurate estimation of durations (Sharma and Rajpoot, 2013). This process of duration analysis is called as 'syllabification' in this work. Syllabification has been performed on the transcribed data. The complete process of syllabification is explained in Chapter 3.

F_0 Contour

For voiced speech, F_0 is usually defined as the rate of vibration of the vocal folds. F_0 is considered to be one of the most important features for the characterisation of emotions and is the acoustic correlate of the perceptive pitch. A partial is any of the sine waves of which a complex tone is composed. F_0 is usually the lowest component of frequency, or partial, which relates well to most of the other partials, and is often called as pitch. In a periodic waveform, most partials are harmonically related, meaning that the frequency of most of the partials are related to the frequency of the lowest partial by a small whole-number ratio. The frequency of this lowest partial is the fundamental frequency of the waveform and how this fundamental frequency changes over the period of time is determined by the F_0 contour (Sharma and Rajpoot, 2013).

Energy Contour

In linguistics, specifically in speech synthesis and music, the energy contour of a sound is a function or curve that tracks the perceived pitch of the sound over time. It may be used to utilize many pitches and multiple sounds at a time. It can relate to frequency function at one point in time to the frequency function at a later point. It is a fundamental linguistic concept of tone, where the pitch or change in pitch of a speech unit over time affects the semantic meaning of a sound. It also indicates intonation in pitch accent languages.
For this work, pitch index marking has been performed after carefully observing the speech signal. Details of this work are given in Chapter 3.

1.7.6 Feature Extraction for PE and Speaker Classification

After the linguistic analysis and prosody marking, data files are used to extract the features. Features are extracted in two phases. In first phase, features are extracted for the development of PE and in second phase, features are extracted for speaker

classification. Details of features extraction process for PE followed by training and testing of PE is explained in Chapter 4. Feature extraction and selection process for speaker classification followed by development of EQNN and CPSOSVM are presented in Chapter 5.

1.8 Gaps in Literature

After reviewing the literature, following gaps have been identified.

- It has been observed that a small amount of work has been done in the field of development of an efficient ASR for Punjabi language. Study of the prosodic and phonetic features of Punjabi language for developing an efficient ASR has also not been done by the researchers.

- Most of the work done so far in the ASR for Punjabi language is for isolated word recognition. There is a requirement to work on continuous speech (Ravinder, 2010; Kumar and Singh, 2011).

- There is a limited work done on continuous Punjabi speech recognition (Al-Qatab and Ainon, 2010; Kumar *et al.*, 2012). In these works, however, the speech is recorded in a pre-defined environment. There is a need to record data in a natural environment and from speakers covering all the dialectal variations for Punjabi language.

- Speech, by default, incorporates prosody. The prosodic information has not been used by researchers to propose Punjabi ASR systems. There is a possibility of improving the ASR systems when prosody information is used.

- It is found that the existing speaker classification techniques suffer from the over-fitting issue (Rafiq *et al.*, 2001; Panchal *et al.*, 2011). Therefore, to overcome the over-fitting issue, it is required to design novel machine learning model.

- From the related work, it has also been found that the parameter tuning of the existing models has been neglected by the majority of the existing literature (Keerthi, 2002; Friedrichs and Igel, 2005).

1.9 Objectives

(i) To study the unique prosodic and phonetic features of Punjabi language for the process of Punjabi speaker recognition.

(ii) To collect and analyse the data for read speech, lecture speech and conversational speech modes from native speakers of Punjabi Language.

(iii) To perform pre-processing and normalization on collected data.

(iv) To perform phonetic analysis of Punjabi Language for the evolvement of Phonetic Engine and for the speaker classification.

(v) To propose and validate Punjabi speaker classification framework.

1.10 Contributions

Following are the contributions of the work carried out in this thesis:

- Initially, extensive review has been done to identify unique prosodic and phonetic features of Punjabi language. Punjabi, being highly tonal language, exhibits different properties like stess, intonation and tone. Same word changes its meaning when used with different tone. All these properties have been studied and finally incorporated in building the PE.

- As a first step towards building the PE, data is collected for experimental purpose. The data is collected from different regions of Punjab in order to capture all the dialectal variations of Punjabi language. Data has been collected in three different modes, namely, read speech mode, lecture speech mode and conversational speech mode. Thereafter, feature extraction techniques are applied to collect the features of the data. Finally, feature selection techniques are applied to select the significant features for speaker classification.

- In this research work, a phonetic level engine is designed and implemented for the continuous speech of Punjabi language. The collected data is used for modeling and feature extraction processes. In the development of Phonetic Engine, MFCCs are used as a feature extraction technique and HMM as a classifier.

- An EQNN technique for speaker classification has also been designed. It composes three characteristics as: (i) A novel technique to train a neural network that picks new training data in different iterations along with the error data of training from previous iteration among all available ones to address a particular query in a supervised learning context. (ii) The ensembling of standard machine learning technique at the training-level to form the final speaker classification, and (iii) The features with wide range of values are used by proposed EQNN to classify the speakers.

- A Crossover based Particle Swarm Optimization with Support Vector Machine (CPSOSVM) is designed and implemented for speaker classification. In CPSOSVM, PSO is used to tune the parameters of SVM. The crossover operator is applied on PSO as it has an ability to overcome the issue of of getting stuck in local minima with the standard PSO. Also, the comparisons are done with the competitive machine learning models and CPSOSVM by considering various performance measures in terms of accuracy, F-measure, specificity and sensitivity.

1.11 Thesis Organization

The chapter-wise organization of this thesis is given below.

Chapter 1: Introduction

In first chapter of the thesis, we have introduced the basic concepts related to the work done in this thesis. It includes the introduction of PE, basics of ML models, prosodic features of Punjabi language, and the challenges faced in carrying out this work.

Chapter 2: Review of the Related Work

In Chapter 2, a comprehensive and illustrative literature review in the domains of linguistics, speech recognition, PE and speaker classification is presented. The details of the existing speech recognition and speaker classification techniques along with their strengths and weaknesses are also presented. The overall objective of this chapter is to evaluate the various gaps found in the literature.

Chapter 3: Data Collection and Prosody Marking

In this chapter, the entire process of data collection is discussed. The data is collected in three modes, namely, read speech mode, lecture mode and conversational speech mode for PE and speaker classification. Thereafter, unique phonological and prosodic features of Punjabi language are presented followed by the four layered prosody marking, namely, phonetic transcription using IPA symbols, break index marking, pitch accent marking and syllabification on the collected data.

Chapter 4: Hidden Markov Model Based Phonetic Engine

This chapter discusses that how PE is developed for read speech mode data using HMM. The main architecture of the designed PE and its mathematical formulation are presented in this chapter. The PE developed in this work has been evaluated for its effectiveness. The experiments conducted and their results for this purpose are also presented in this chapter.

Chapter 5: Machine Learning Based Speaker Classification

This chapter describes the existing and two proposed techniques designed in this thesis for text-independent speaker classification. The proposed techniques are EQNN and CPSOSVM. Both the speaker classification techniques are then explained with the help of algorithm and diagrammatic flow. Finally, the performance analysis of the both proposed technique are demonstrated after comparison with the existing techniques.

Chapter 6: Conclusions and Future Work

This chapter concludes the thesis by highlighting the contributions made towards the proposed research domain. Moreover, this chapter also provides the future directions in this research area. PE is an evolving field, and this chapter brings to light the huge amount of scope for developing approaches, and applications in this domain. It highlights the contributions made by our work and enlists the points that need to be worked on in the future.

Chapter 2

Review of the Related Work

This chapter contains a detailed review of the literature on speech recognition and speaker classification. The flow of this chapter has been maintained as per the development process of a conventional speech recognition system. It has been divided into five parts. The first part deals with the existing literature that have used Mel Frequency Cepstral Coefficient (MFCC) as a feature extraction technique. In the second part, existing speech processing systems that have been developed using Hidden Markov Models (HMMs) are reviewed followed by the third part wherein existing speech recognition systems that have been developed using HMM toolkit (HTK) are presented. In fourth and fifth parts, an overview of work done on Phonetic Engine (PE) for different languages and speaker classification, respectively, has been presented. Figure 2.1 shows the organisation of review of related work.

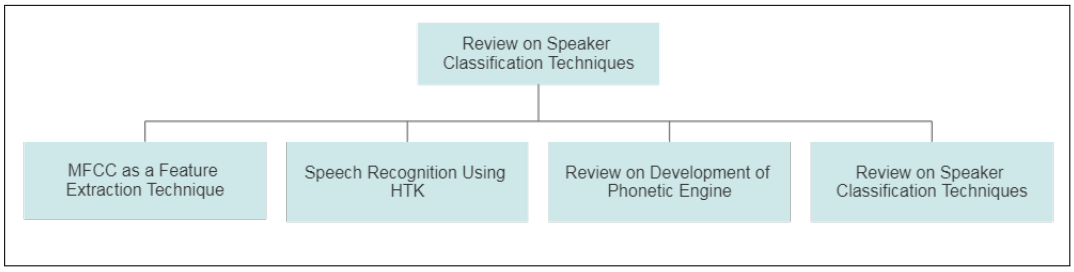


Figure 2.1: Organization of related work

2.1 MFCC as a Feature Extraction Technique

It has been observed in literature that MFCCs have extensively been used as a potential feature extraction technique. This section tends to focus on the usage of MFCC

as a feature extraction technique in recent past and presents the brief review of literature on the same. Comparison of MFCC with other techniques has also been presented. Figure 2.2 shows the organisation of related work in usage of MFCC as a feature extraction technique.The summary of literature on MFCC as a feature extraction technique has been presented in Table 2.1.

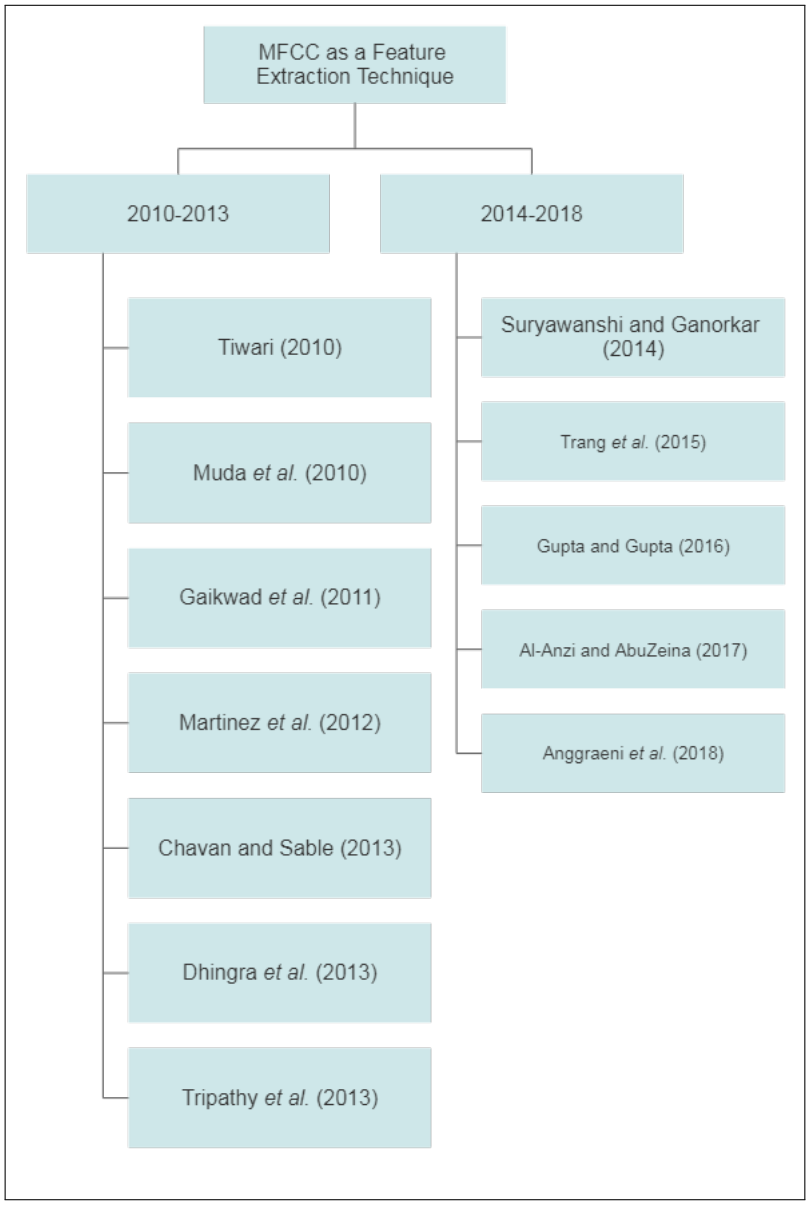


Figure 2.2: Organization of related work in usage of MFCC as a feature extraction technique

Tiwari (2010) exploited the characteristic of speech signal asserting that speech signal and its corresponding spectral properties are all a function of time. Therefore, the time-varying Fourier representation has been used by them to analyse their

spectral properties. In this work, the MFCC features have been used for designing a text-dependent speaker identification system. First of all, MFCCs of each speaker have been computed in both training and testing phase and then Euclidean distance between each speaker have been measured by them; and the speaker with minimum Euclidean distance is assumed as the correct speaker. Muda *et al.* (2010) proposed feature extraction and feature matching for digital signal processing that have been applied to represent the speech signal. Myriad techniques like Linear Predictive Coding (LPC), HMM, Artificial Neural Network (ANN) and others have been exercised with an intention to propose an efficient and effective method for speech recognition. As a first step, pre-processing and signal filtering have been performed, then the matching process was implemented. To model the human auditory perception system, the non-parametric, logarithmically aligned frequency bands, MFCCs have been extracted. The coin-flipper of Dynamic Time Wrapping (DTW), has been applied for features matching process. The work has presented the feasibility of MFCCs to compute speech feature parameters and DTW to match the test patterns. Gaikwad *et al.* (2011) proposed a Fusion MFCC feature extraction technique which is a combination of MFCC and Linear Discriminant Analysis (LDA) for speech recognition of Marathi language. The results of fusion MFCCs have been compared with MFCC and LDA, individually. The proposed technique has been implemented on 625 Marathi sentences. The accuracy of 77.8%, 73.0% and 89.2% have been achieved in case of MFCC, LDA and Fusion MFCC, respectively. Martinez *et al.* (2012) presented a fast and accurate text-dependent voice recognition system. MFCCs have been used to extract the features from voice and vector quantization technique has been used to identify the speaker. Vector quantization has been used because it allows to model a probability functions by the distribution of different vectors. 100% precision has been achieved by them for a dataset of 10 English speakers. Chavan and Sable (2013) implemented and measured the performance of text-dependent speaker independent isolated word speech recognition system, which has been developed using HMM and MFCC as the parameterization and feature extraction technique respectively. These MFCC features have been employed in the training of the system. Forward backward algorithm with Expectation-Maximization (EM) principle has been used for parameter estimation and re-estimation in HMM modeling of the system. The probability of generation of speech observations and the most probable path sequence using each stored HMM model has been calculated to recognize the spoken utterance. In the end, the path with maximum likelihood, *i.e.*, path with maximum probability has been selected as the recognized word. The system has been trained with their own built

dataset, which consists of 60 speech samples of selected words. Dhingra *et al.* (2013) discussed an avenue of isolated speech recognition through the MFCC and DTW. Multiple feature parameters have been extracted from a wave signal of the spoken utterance. A speech dataset of five speakers has been prepared for the experimentation. Each of these speakers spoke 10 digits under an acoustically controlled room. Then, as a process of feature extraction, MFCCs have been computed from a wave signal of the spoken utterance. To cope with the variation of speaking speeds, DTW has been used. DTW has been used for measuring the match between two speech segments, which may fluctuate in time or speed. Some coefficients of these logarithmically aligned frequency bands corresponding to the frequencies of Mel scale of speech Cepstrum have been computed with the help of spoken word samples that were stored in the dataset. For mapping the unknown speech utterance with the speech dataset, a variation measure based on minimizing the Euclidean distance has been applied. Using MATLAB, the experimental results have been analysed and it has been shown that the results were efficient for the experiments. Tripathy *et al.* (2013) proposed Hindi speech recognition system using MFCCs and LPC as the feature extraction techniques. They have also compared the two techniques. In this work, HMM has been used as a classifier and has been implemented through HTK toolkit. The proposed system has been tested on both the environments: speaker-dependent and speaker-independent. For this work, a vocabulary of 35 Hindi words has been prepared and five speakers (2 males and 3 females) have been engaged for recording the Hindi speech.

Suryawanshi and Ganorkar (2014) suggested a speech recognition system based on digital signal processor with enhanced performance score in terms of accuracy and cost of computation. Their work has demonstrated a technique of isolated speech recognition by using MFCC and Euclidean distance. With various speech features, a vector has been computed from speech signal of the spoken utterance. An experimental speech dataset involving five speakers has been built, in which each speaker spoke 5-10 words under an acoustically controlled room. MFCCs were extracted from the speech signal of a spoken utterance. Concept of minimal Euclidean distance has been applied to compare inter speaking differences. Trang *et al.* (2014) presented the usage of MFCCs with two novel methods that combine MFCC feature extraction method with Principle Component Analysis (PCA) technique. The speed of training of HMM and its accuracy with conventional MFCC feature extraction method along with two other approaches has been compared. The other approaches include usage of PCA with MFCC to reduce dimensions and time complexity. The proposed method has been trained on a set of 4,500 utterances of 50 different words,

and tested on another set of 500 utterances of these words. The accuracy of 89.0% and 90.8% has been achieved with conventional MFCCs and the proposed MFCCs with PCA respectively. Gupta and Gupta (2016) presented a study of ASR system with respect to widely used feature extraction techniques. They have presented a comparison of MFCC, LPC and Relative Spectral Transform (RASTA) features. The work done by them for proposing an AR for English language yielded an accuracy of 99.9%. For other Arabic and Indian languages, MFCCs have been proved to be a better choice in comparison with LPC and RASTA.

Al-Anzi and AbuZeina (2017) used MFCCs to identify different speech segments to obtain the Arabic language phonemes and used these phonemes for further training and decoding steps. HTK has been used to obtain the MFCCs for Arabic ASR system. MFCC features from a single Arabic word speech file of 0.323 seconds, has been extracted. The detailed 13 feature values of MFCCs have been presented and they have concluded that MFCCs are most widely used feature extraction technique for Arabic ASR.Anggraeni *et al.* (2018) used MFCCs to extract features of speech signal for the implementation of an application to pick and place an object with a Robot Arm. A 12-feature vector has been used to train the Support Vector Machine (SVM). The trained model has then been tested by trained respondent for speaker classification. With trained respondent, an accuracy of 80.0% has been achieved whereas with untrained respondent, an accuracy of 70.0% has been achieved.

2.2 HMM Based Speech Processing

HMMs have widely been used for classification task in the area of speech processing. A detailed review of the usage of HMMs in speech processing, specifically in speech recognition, has been presented in this section. Figure 2.3 shows the organization of related work in HMM based speech processing. Table 2.2 summarizes the literature review carried out during this work.

Lee and Hon (1989) proposed speaker-independent phone recognition system based on discrete HMMs. In this system, HMMs have been trained using *TIMIT* sentences from 357 speakers and has been evaluated on 160 *TIMIT* sentences from 20 speakers. LPCs have been used as a feature extraction technique in their work. A new novel smoothing algorithm which smooths the HMM output parameters has also been proposed in this work. For 39 English phones, 64.0% recognition rate with context-independent phone models has been achieved and with context-dependent phone models, a recognition rate of 73.8% has been achieved. Rabiner *et al.* (1989) proposed connected digit recognition system based on HMMs.

Table 2.1: Summary of literature on MFCC as a feature extraction technique

S. No.	Reference	Supporting Tools/Techniques	Toolkit/Environment	Language	Application	Results
1	Tiwari (2010)	Euclidean Distance	MATLAB	-	Text-dependent Speaker Identification	An accuracy of 85.0% has been achieved on dataset of 5 unknown speakers.
2	Muda *et al.* (2010)	DTW	MATLAB	English	Voice Recognition	Input test voice has been matched optimally with the reference voice on dataset of 2 Speakers speaking 5 phrases.
3	Gaikwad *et al.* (2011)	LDA	-	Marathi	Continuous Speech Recognition	The accuracy of 77.8%, 73.0% and 89.2% have been achieved in case of MFCC, LDA and Fusion MFCC respectively on 625 Marathi sentences.
4	Martinez *et al.* (2012)	Vector Quantization	MATLAB	English	Text Dependent Voice Recognition	100% precision has been achieved for a dataset of 10 speakers.
5	Chavan and Sable (2013)	HMM	MATLAB	-	Text-Dependent Speaker Independent Isolated Word Speech Recognition	92.0% accuracy has been achieved for a dataset of 60 speech samples of selected Noun words.
6	Dhingra *et al.* (2013)	DTW and Euclidean Distance	MATLAB	English	Isolated Speech Recognition	Efficient results have been achieved for a speech dataset of total five speakers speaking 10 digits individually.
7	Tripathy *et al.* (2013)	LPC and HMM	HTK	Hindi	Speaker-dependent and Speaker-independent Speech Recognition System	An accuracy of 76.4% has been achieved for a vocabulary of the size of 35 words spoken by five speakers.

Table 2.1: Continued

S. No.	Reference	Supporting Tools/Techniques	Toolkit/Environment	Language	Application	Results
8	Suryawanshi and Ganorkar (2014)	Euclidean Distance	MATLAB	English	Isolated Speech Recognition	Accuracy of 76.7% has been achieved for the dataset built from five speakers in which each speaker spoke 5-10 words.
9	Trang *et al.* (2014)	PCA	ARM Base A8	English	Isolated Speech Recognition	Accuracy of 89.0% has been achieved for the dataset of 90 utterances per word of 50 words.
10	Gupta and Gupta (2016)	LPC and RASTA	-	English and Urdu	Automatic Speech Recognition	Accuracy of 99.9% and 86.7% has been achieved for the English and Urdu dataset respectively.
11	Al-Anzi and AbuZeina (2017)	-	HTK	Arabic	Automatic Speech Recognition	MFCC features from single word speech file of 0.323 seconds has been extracted.
12	Anggraeni *et al.* (2018)	SVM	-	English	Speech Recognition to Pick and Place an Object with a Robot Arm	Accuracy of 80.0% and 70.0% has been achieved respectively for trained and untrained respondent for two words: pick and place.

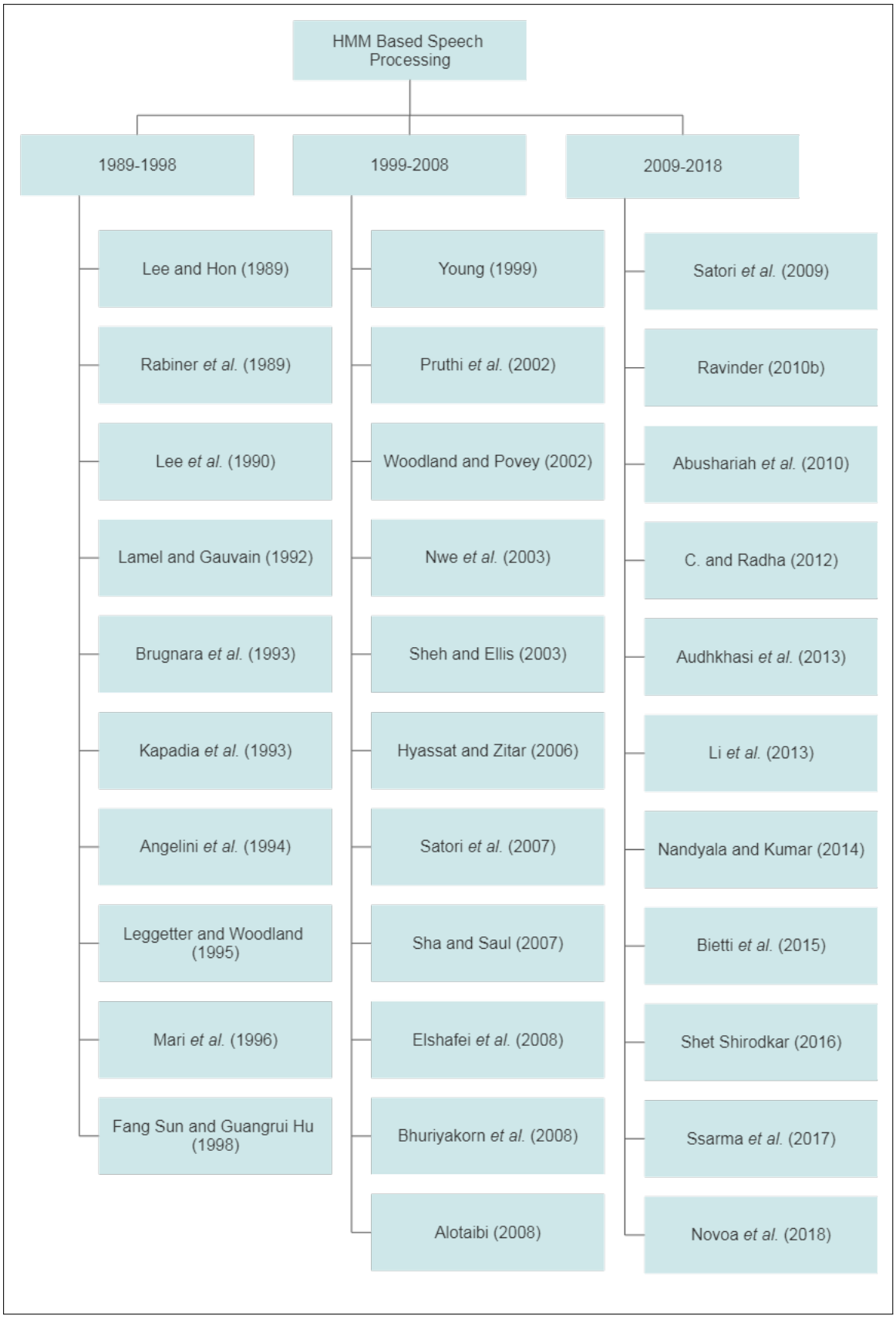


Figure 2.3: Organization of related work in HMM based speech processing

The system proposed by them has been trained and tested in three modes: (i) Speaker trained, (ii) Multi-speaker and (iii) Speaker-independent. The evaluation has been done on three datasets: (i) National Bureau of Standards (NBS) dataset, (ii) Isolated digit dataset of 225 adult talkers and (iii) Connected digit dataset of 50 talkers. The string error rates of 0.8%, 2.9% and 2.9% have been observed for the three modes, posed in their work. Lee *et al.* (1990) proposed speech recognition system using SPHINX tool. This tool is based on discrete HMMs and LPCs as feature vector. In order to train the 48 context independent phonetic HMMs, 4200 sentences have been used spoken by 105 speakers. Phone HMMs have been concatenated to build word HMMs, which are further concatenated to build large sentence HMMs. It has been observed that when words occur in a cluster, it becomes difficult to recognize them. The training has been done in two stages: (i) In the first stage 48 context-independent phonetic HMMs were trained, and (ii) In the second stage, trained models from the first stage initialized context-dependent phone models. This system has been evaluated on 150 sentences spoken by 15 speakers. With word-pair grammar, a word recognition accuracy of 96.0% has been obtained and with null grammar, a word recognition accuracy of 82.0% was achieved.

Lamel and Gauvain (1992) proposed speaker-independent phoneme recognition system for a continuous speech of French language. Data has been collected from 43 speakers to train the HMMs and from other 19 speakers for testing. Data has been collected using large read speech corpus. For 35 context-independent phoneme models, they achieved a phone accuracy of 60.0%. Also, for the context-dependent models, a phone accuracy of 68.6% has been obtained by them. Brugnara *et al.* (1993) proposed automation of segmentation and labelling of speech of Italian language using HMMs. Training and performance evaluation have been performed on the *TIMIT* dataset. For training purpose, 64 speakers have been selected, with eight sentences from each speaker. The performance has been evaluated on 24 different speakers. They have observed an accuracy of 93.2% for automatic segmentation of 64 utterances with 40ms tolerance. Kapadia *et al.* (1993) proposed a phoneme recognition system based on continuous density mono-phone HMMs. HMMs have been trained using Maximum Mutual Information Estimation (MMIE) algorithm. In this work, a comparison has been made between maximum likelihood and MMIE training algorithms for both types of models, namely, diagonal and full covariance models. Performance and implementation issues related to MMIE training were also discussed. As a result, it has been observed that as the complexity of models increases. The performance of MMIE trained recognition system improves but the performance of maximum likelihood trained recognition system decreases.

Angelini *et al.* (1994) proposed speaker-independent speech recognition system for Italian language using continuous density HMMs. The proposed system has been trained and tested using *APASCI* corpus. In this work, a set consisting of 38 context independent units have been evaluated and two sets of other context dependent units were also considered which performed differently. A vocabulary of the size of 3900 words read by 88 male and 88 female speakers consisting of most frequent Italian words has been used. Performance of the model has been evaluated in terms of phone loop recognition accuracy and word loop recognition accuracy. Leggetter and Woodland (1995) proposed maximum likelihood linear regression technique for speaker adaptation of continuous density, *i.e.*, Gaussian mixture HMMs. Modeling of the new speaker has been improved by using an initial speaker-independent system by updating the HMM parameters. Maximum Likelihood Linear Regression (MLLR) has been used to maximise the likelihood of adaptation. They have performed experiments on the *ARPA RM1* dataset using HMMs with continuous density mixtures and crossword tri-phones. It has been observed by them that with supervised adaptation, 37.0% error reduction was achieved and with unsupervised adaptation, 32.0% of error reduction has been achieved using 40 adaptation utterances. Mari *et al.* (1996) proposed second order HMMs using Maximum Likelihood Estimation (MLE) paradigm for word and phone based, speaker-independent, continuous speech recognition. In this work, it has been observed that the second order HMMs yield better performance than first order HMMs. Data for this work has been collected from speech telephone corpus and experiments have been conducted on spelt names over telephone. More than 4000 people were asked to spell their first and last names with and without pauses over telephone and their respective voices were recorded. For training purpose, 1200 calls and for testing purpose, 491 calls were selected. It has been observed that the second order HMMs can achieve more than 69.0% of accuracy.

Fang Sun and Guangrui Hu (1998) proposed a Genetic Algorithm (GA) to train HMMs for speech recognition system and a comparison has been made between the proposed algorithm and traditional HMM training algorithm (Baulm-Welch algorithm). The proposed algorithm has been tested on recognition of isolated words. For training purpose, 3000 words and for the testing purpose, 500 words have been used. At the time of evaluation of the recognition system, it has been observed that with GA, the accuracy of the system had been 96.2% and with traditional training algorithm, the accuracy was 94.0% under the same conditions. Young (1999) presented acoustic modeling based on N-gram model for Large Vocabulary Continuous Speech Recognition (LVCSR). The objective of LVCSR has been to transcribe

input speech into an orthographic transcription. It was assumed that input speech consisted of a sequence of words and using the language model probability of any specific word sequence could be determined. MFCCs have been used as the feature extraction technique. It has been observed that to get the good phonetic discrimination, for each different context, HMMs required to be trained and the most common context had been the tri-phone. Cross-word tri-phones provided the best modeling accuracy but too many parameters required to be computed. To overcome that, state-tying context was used by them. Pruthi *et al.* (2002) proposed the implementation of *Swaranjali*: An HMM based speaker-dependent real-time isolated word recogniser for Hindi. Data has been collected from two male speakers, who were asked to utter Hindi digits from zero (0) to nine (9), two times. Using these 20 tokens, HMMs have been trained. Vector Quantization (VQ) has been used for processing speech signal. After training, the evaluation was performed on the proposed system to check the accuracy. Some errors have been recognised due to plosive sounds at the beginning and at the end of each word. On an average, 84.5% has been the accuracy of the system for first speaker and for second speaker it has been 84.3%. Woodland and Povey (2002) proposed a framework based on continuous density HMMs for providing the discriminative training to the large vocabulary speech recognition systems. To train HMMs, the MMIE method has been used. They used 265 hours of training data for conversational telephone speech transcription. In this, tri-phone and quin-phone HMM parameters have been estimated, which led to a reduction in word error rate for the transcription of conversational telephone speech. Also, a scheme which reduced the danger of over-training has also been shown. This scheme has been based on linear interpolation of MMIE and MLE objective functions.

Nwe *et al.* (2003) proposed a text-independent method for emotion classification of the speech based on discrete HMMs. Log Frequency Power Coefficients (LFPCs) have been used to represent the speech signals. In their system, emotions have been classified into six categories: anger, disgust, fear, joy, sadness and surprise. Data has been collected from twelve speakers and every speaker has been asked to contribute 60 emotional utterances. A comparison of LFPC feature parameters has been made with Linear Prediction Cepstral Coefficients (LPCCs) and MFCC feature parameters. It has been observed in their work that LFPC, as feature parameter, showed better performance than other traditional feature parameters. Their system achieved an accuracy of 78.0% Sheh and Ellis (2003) proposed a system for automating the chord segmentation and recognition based on EM-trained HMMs. In this work, an automated chord transcription system has been built. Pitch Class Profile (PCP) features have been extracted from the speech signal and HMMs have been used in

their system for sequence recognition and have been trained using the EM algorithm. The chord sequences have been given as input without requiring the precise timings of the chord changes which were computed automatically at the time of training only. The system showed 75.0% accuracy when evaluated on a small set of 20 *Beatles* songs. Hyassat and Zitar (2006) proposed Arabic speech recognition. In this work, first SPHINX-IV based Arabic recognizer has been introduced and then an automatic toolkit has been proposed that is capable of producing pronunciation dictionary for both: (i) The Holy Quran and (ii) Standard Arabic language. In this work, three corpus have been developed: (i) Holy Quran corpus of about 18.5 hours, (ii) Command And Control corpus (CAC-1) of about 1.5 hours and (iii) Arabic Digit Corpus (ADC) of about less than one hour. For each corpus, three acoustic models have been developed by providing the training to SPHINX-IV engine, which is based on HMM model. Satori *et al.* (2007) proposed a novel approach for building an automated speech recognition system for the Arabic language. For building this system, utilities of Sphinx-4 engine have been used. They designed the system for recognizing 10 Arabic digits and named it as Hello_Arabic_Digit application. They created a primary dataset consisting of 300 wave files, each file corresponds to a spoken Arabic digit. These 300 files contains the spoken utterances of each of the 10 Arabic digits spoken 5 times by different speakers. For training purpose, all 300 utterances have been used. For checking the performance of the trained system, three new male speakers have been asked to utter all the Arabic digits. Mean recognition rate for each of the three speakers has been computed. For speaker 1, it was found to be 86.7%, for speaker 2 it was 86.7% and for speaker 3 it was 83.3%.

Sha and Saul (2007) proposed a new approach for providing discriminative training to Continuous Density Hidden Markov Models (CD-HMMs). In this work, two popular approaches (based on Minimum Classification Error (MCE) and Conditional Maximum Likelihood (CML)) have been compared with a new approach which is based on margin maximization. This new approach removed the problem of spurious local minima which had been observed in other approaches. This approach leads to convex optimization over the parameter space of CD-HMMs. On the *TIMIT* speech data corpus, phonetic recognizers have been built using trained CD-HMMs from all three approaches. It has been observed that the new proposed approach is better than the other two approaches as there has been less phonetic error rate as compared to others.

Bhuriyakorn *et al.* (2008) presented phoneme recognition of continuous speech of Thai language. In this work, an approach of estimating HMM topology has been proposed. The whole process includes combining different objective func-

tions and topology generation methods; and then a set of suitable topologies has been constructed. Now GA has been used as the topology selection algorithm considering global fitness. This resulted into 4.4% of error reduction when compared with already defined left-to-right HMM models. Elshafei *et al.* (2008) proposed a speaker-independent natural Arabic speech recognition system. This system has been based on HMMs and developed using Sphinx tools. This system has tri-phone based acoustic model using five state HMMs in which first and the last state was non-emitting and others were emitting states. It has used a continuous density of eight Gaussian mixture distributions. A Total of 5.4 hours of data has been used in their work. Out of this 5.4 hours data, 4.3 hours of data has been used for training and the remaining 1.1 hours of data has been used for testing. In the pronunciation dictionary, 14,232 words have been defined and language model contained both bi-grams and tri-grams. After testing the system, word error rate has been observed to be 9.0%. Alotaibi (2008) proposed Arabic digit recognition system and did a comparative study of HMMs and ANNs. The proposed system has been implemented using HMM for isolated word phoneme based recognizer. After evaluating the performance of both recognizers it has been observed that ANN based recognizer, obtained 99.5% accuracy in multi-speaker mode and 94.5% in speaker-independent mode while HMM based recognizer obtained 98.1% accuracy in multi-speaker mode and 94.8% in speaker-independent mode. Satori *et al.* (2009) proposed a novel approach for building an automated speech recognition system for Arabic language. For building this system, utilities of Sphinx-4 engine have been used. The difficulties that were faced in developing this system for Arabic language have been: large amount of non-diacritized content, huge variety of dialectal and morphological complexity. Dataset for this system consisted of 35 male speakers and 25 female speakers, who have been asked to utter all 10 digits five times. For training purpose, all 3000 utterances have been used. MFCCs have been used as a feature extraction technique. For checking the performance of trained system, three new male speakers and three new female speakers have been asked to utter all 10 Arabic digits and their mean recognition ratio has been computed. Their recognition rates were 96.7% for first male speaker, 93.3% for second male speaker and 93.3% for third male speaker. For first female speaker, it was found to be 86.7%, for second female speaker it was 83.3% and for third female speaker it was 90.0%. Kumar (2010) proposed a comparison between HMM and DTW techniques for speaker-dependent isolated word recognition of Punjabi language. In DTW approach, the time warping technique has been combined with linear predictive coding analysis and in HMM approach, HMMs have been combined with LPC analysis.

The DTW used Nearest-Neighbour as the decision rule and HMM used the maximum likelihood as the decision rule. For implementing this system, Visual C++ with multimedia API, has been used on Windows platform. Data has been collected from one male speaker. For the comparison between both the techniques, codebook of the size 256 words have been used. After comparison, it has been observed that DTW based recognizers showed better performance than HMM based recognizers because of the insufficiency of the training data but the time and space complexity of HMM based approach have been observed to be less than DTW based approach.Abushariah *et al.* (2010) proposed English digits speech recognition system based on HMMs. MFCCs have been used as a feature extraction technique. Two modules have been implemented: (i) Isolated word speech recognition, and (ii) Continuous speech recognition. Both modules have been tested in a clean and noisy environment. It has been observed that in both the environments, multi-speaker mode performed better than the speaker-independent mode. C. and Radha (2012) presented speaker-independent speech recognition system for Tamil language. This system has been developed for recognising the isolated words and it has been based on HMMs. MFCCs have been used as a feature extraction technique. Word Error Rate (WER) parameter has been considered for measuring the performance. Data has been collected from ten speakers, out of which, four speaker's data has been used for performance evaluation. An accuracy of 88.0% has been achieved for the dataset of isolated words from 10 speakers. Audhkhasi *et al.* (2013) presented a new Noisy Expectation- Maximization (NEM) algorithm to show the likelihood of effect of noise in HMMs. It has been shown through simulations that convergence of noisy HMMs are faster than that of conventional HMMs. HTK has been used to perform the experiments. This analysis has been performed on the *TIMIT* dataset. There has been a significant improvement in per-frame log-likelihood for the noisy HMM as compared with the HMM.

Nandyala and Kumar (2014) proposed a hybrid approach based on HMM and DTW for speech recognition. Kernel adaptive filter has been used for speech enhancement and MFCCs have been used for feature extraction. Also, HMMs have been used for training and DTW for classification. A relative improvement has been observed in results when compared with traditional methods. An accuracy of 94.0% has been achieved on the dataset of 10 nouns. Bietti *et al.* (2015) proposed an online clustering and joint segmentation framework for audio signals. An incremental EM algorithm has been proposed after employing HMMs and Semi-HMMs for online parameter learning. The main aim of this was to propose online unsupervised joint segmentation and clustering in one pass. The segmentation algorithms were applied

to auditory scenes using examples from *Office Live Dataset*. It was observed that online EM algorithm runs faster and gives a lower likelihood value as compared to incremental EM algorithm.

Shet Shirodkar (2016) proposed an isolated word speech recogniser for Konkani language digits. HMMs have been used as a classifier and the recogniser has been developed using HTK. The dataset of Konkoni digits has been collected from different native speakers. It has been claimed that an accuracy of 80.0% was achieved for phoneme level acoustic model and an accuracy of 79.4% was achieved for word level acoustic model. Li *et al.* (2013) proposed an emotion recognition system using Deep Neural Network HMMs (DNN-HMMs) with Restricted Boltzmann Machine (RBM) based unsupervised and discriminative pre-training. Experiments have been performed for emotion recognition using the *eNTERFACE'05* dataset and *Berlin* dataset using the above stated models. The results have been compared with the GMM-HMMs, as well as with the Multi-Layer Perceptrons HMMs (MLP-HMMs). The DNN extends the labelling ability of GMM-HMM when the number of hidden layers and units have been set properly. For the *eNTERFACE'05* dataset, the recognition accuracy improves by 12.2%, 11.7%, 10.6% and 17.2% respectively from the DNN-HMMs with unsupervised pre-training, the GMM-HMMs, the MLP-HMMs and the shallow-NN-HMMs.

Ssarma *et al.* (2017) presented the HMM based speaker-independent isolated word ASR system for Nepali language. The system has been developed in Python using *NumPy* and *YAHMM* libraries, and MFCCs have been used as a feature extraction technique. Noise reduction and voice activity detection modules have also been added in the pipeline to reinforce the accuracy of the system. The data has been collected from different native Nepali speakers in a room environment and ASR system has been trained and tested in the same environment. The overall accuracy of system has been reported as 75.0%. Novoa *et al.* (2018) proposed a Human Robot Interaction (HRI) environment based representation and modeling for automatic speech recognition. This modeling has been performed by training DNN-HMMs based ASR system using clean utterances and noise from HRI test bed with *PR2* mobile manipulation robot. A reduction of 26.0% and 38.0% in WER has been reported over the available speech recognition APIs and human testing databases respectively.

Table 2.2: Summary of literature on HMM based speech recognition

S. No.	Reference	Supporting Tools/Techniques	Feature Extraction Technique	Language	Application	Results
1	Lee and Hon (1989)	A novel Smoothing Algorithm	LPC	English	Speaker-independent Phone Recognition System	The accuracy of 64.0% and 73.8% have been achieved for context-independent and context-dependent phone models respectively on *TIMIT* dataset of 6300 sentence.
2	Rabiner *et al.* (1989)	-	LPC	English	Connected Digit Recognition System	A string error rate of 0.78, 2.85 and 2.94 have been observed for databse from NBS, 225 isolated digits and 50 connected digits.
3	Lee *et al.* (1990)	SPHINX	LPC	English	Speaker-independent Continuous Speech Recognition	With word-pair grammar, word recognition accuracy of 96.0% has been obtained and with null grammar word recognition accuracy has been 82.0% for the dataset of 4200 sentences.
4	Lamel and Gauvain (1992)	SPHINX	LPC	French	Speaker-independent Phoneme Recognition System	The accuracy of 60.0% and 68.6% have been achieved for 35 context-independent and 428 context-dependent phoneme model for the dataset collected from 62 speakers.
5	Brugnara *et al.* (1993)	-	-	Italian	Automatic Segmentation and Labelling of Speech	An accuracy of 93.5% has been achieved for dataset of sentences from 88 speakers.

Table 2.2: Continued

S. No.	Reference	Supporting Tools/Techniques	Feature Extraction Technique	Language	Application	Results
6	Kapadia *et al.* (1993)	MMIE	-	English	Phoneme Recognition System	Accuracy of MMI has been more than ML on *TIMIT* dataset
7	Angelini *et al.* (1994)	-	-	Italian	Speaker-independent Speech Recognition	An accuracy of 79.0% has been achieved by increasing the mixture components for APASCI dataset.
8	Leggetter and Woodland (1995)	HTK	MFCC	English	Improved Speaker-independent Modeling	37.0% and 32.0% error reduction have been achieved with supervised and unsupervised adaptation of error reduction using 40 adaptation utterances.
9	Mari *et al.* (1996)	MLE	MFCC	English	Phone based, Speaker-independent, Continuous Speech Recognition	An accuracy of more than 69.0% has been achieved with second order HMMs for dataset of first and last name of 4000 people.
10	Fang Sun and Guangrui Hu (1998)	GA	-	English	Isolated Word Recognition	An accuracy of 96.2% has been achieved by GA for 4500 isolated words
11	Young (1999)	HTK	MFCC	English	Large Vocabulary Continuous Speech Recognition	Maximum error of 5.0% to 15.0% has been seen on a large vocabulary of 10000 words.
12	Pruthi *et al.* (2002)	VQ	LPC	Hindi	Speaker-dependent Real-time Isolated Word Recogniser	An accuracy of 84.5% has been achieved for dataset of 20 token of digits

Table 2.2: Continued

S. No.	Reference	Supporting Tools/Techniques	Feature Extraction Technique	Language	Application	Results
13	Woodland and Povey (2002)	MMIE	-	English	Large Vocabulary Speech Recognition	The reduction of 5.0% to 10.0% in word error rate has been observed over MLE for 265 hours of telephonic conversation data.
14	Nwe *et al.* (2003)	VQ	LFPC	Burmese and Mandarin	Text-independent Emotion Classification of the Speech	An accuracy of 78.0% has been achieved for the dataset of 60 emotional utterances.
15	Sheh and Ellis (2003)	EM	PCP	English	Automatic Chord Segmentation and Recognition	75.0% of accuracy has been achieved when evaluated on a small set of 20 *Beatles* songs.
16	Hyassat and Zitar (2006)	SPHINX-IV	MFCC	Arabic	Arabic Speech Recognition	The accuracy of 70.8%, 98.1% and 99.2% have been achieved for Holy Quran corpus of about 18.5 hours, CAC-1 of about 1.5 hours and ADC of about less than one hour respectively.
17	Satori *et al.* (2007)	SPHINX-IV	MFCC	Arabic	Hello_Arabic_Digit Speech Recognition	An average accuracy of 85.5% has been achieved for the dataset of 300 utterances of Arabic digits.
18	Sha and Saul (2007)	MCE and CML	MFCC	English	Margin Maximization to train CD-HMMs	The proposed margin maximization approach has performed better than MCE and CML for the *TIMIT* dataset.

Table 2.2: Continued

S. No.	Reference	Supporting Tools/Techniques	Feature Extraction Technique	Language	Application	Results
19	Bhuriyakorn *et al.* (2008)	GA	MFCC	Thai	Phoneme Recognition of Continuous Speech	About 4.4% of error reduction in well-trained topologies has been observed over already defined left-to-right HMM models for the dataset of utterances from 248 speakers.
20	Elshafei *et al.* (2008)	HTK	MFCC	Arabic	Speaker-independent Speech Recognition System	The word error rate of 9.0% has been observed for the dataset of 5.4 hours of Arabic words.
21	Alotaibi (2008)	ANN	MFCC	Arabic	Digit Recognition System	For multi-speaker mode, the accuracy of 99.5% and 98.1% have been achieved by ANN and HMM, respectively, for the dataset of 1700 tokens of Arabic digits.
22	Satori *et al.* (2009)	SPHINX-IV	MFCC	Arabic	Automated Speech Recognition System	The accuracy of 94.4% and 86.6% have been achieved for male and female speakers, respectively, on the dataset of 3000 utterances of Arabic words.
23	Kumar (2010)	DTW	LPC	Punjabi	Comparison between HMM and DTW technique for Speaker-dependent Isolated Word Recognition	The overall accuracy of 92.3% and 87.5% have been achieved for DTW and HMM recognizer, respectively, for the dataset of 256 words.

Table 2.2: Continued

S. No.	Reference	Supporting Tools/Techniques	Feature Extraction Technique	Language	Application	Results
24	Abushariah *et al.* (2010)	MATLAB	MFCC	English	Isolated Word Speech Recognition and Continuous Speech Recognition	The accuracy of 88.0% and 82.5% have been achieved for speaker-dependent isolated and continuous speech recognition respectively for the dataset of 1380 sound files.
25	C. and Radha (2012)	SPHINX-IV	MFCC	Tamil	Speaker-independent Speech Recognition System	An accuracy of 88.0% has been achieved for the dataset of isolated words from 10 speakers.
26	Audhkhasi *et al.* (2013)	NEM and HTK	MFCC	English	Speech Recognition using Noisy HMMs	Significant improvement has been shown in per-frame log-likelihood and the convergence for the NHMM over HMM for the *TIMIT* dataset.
27	Li *et al.* (2013)	DNN and HTK	MFCC	Multi-Lingual	Emotion Recognition System	The accuracy of 77.9% and 76.2% have been achieved for DNN-HMM and GMM-HMM, respectively, for the dataset of 495 utterances from *eNTERFACE'05*.
28	Nandyala and Kumar (2014)	DTW	MFCC	-	Speech Recognition	An accuracy of 94.0% has been achieved on dataset of 10 Noun words.
29	Bietti *et al.* (2015)	EM	-	English	Segmentation and Clustering Framework	Online EM algorithm runs in 231s and gives a lower final-likelihood value as compared to iterative EM algorithm for *Office Live* dataset.

Table 2.2: Continued

S. No.	Reference	Supporting Tools/Techniques	Feature Extraction Technique	Language	Application	Results
30	Shet Shirodkar (2016)	HTK	MFCC	Konkoni	Automatic Speech Recognition	The accuracy of 80.0% and 79.4% have been achieved for phoneme and word level acoustic model, respectively, for the dataset of 600 speech files.
31	Ssarma *et al.* (2017)	-	MFCC	Nepali	Speaker-independent Isolated Word ASR System	An accuracy of 66.7% has been achieved for the untrained single world test dataset.
32	Novoa *et al.* (2018)	Kaldi Toolkit	MFCC	English	Human Robot Interaction (HRI) environment based representation and modeling for automatic speech recognition	A reduction by 26.0% and 38.0% in WER has been shown over the available speech recognition APIs and human testing databases, respectively, using the *PR2* dataset.

2.3 Speech Recognition Using HTK

HTK is a HMM toolkit, used to build HMMs. This can also be used to manipulate the existing HMMs. It can be used in many speech processing applications and has been very popular in the field of speech recognition. This section presents a brief survey of speech recognition using HTK. Figure 2.4 shows the organization of related work in speech recognition using HTK. Table 2.3 presents the summary of literature on speech recognition using HTK.

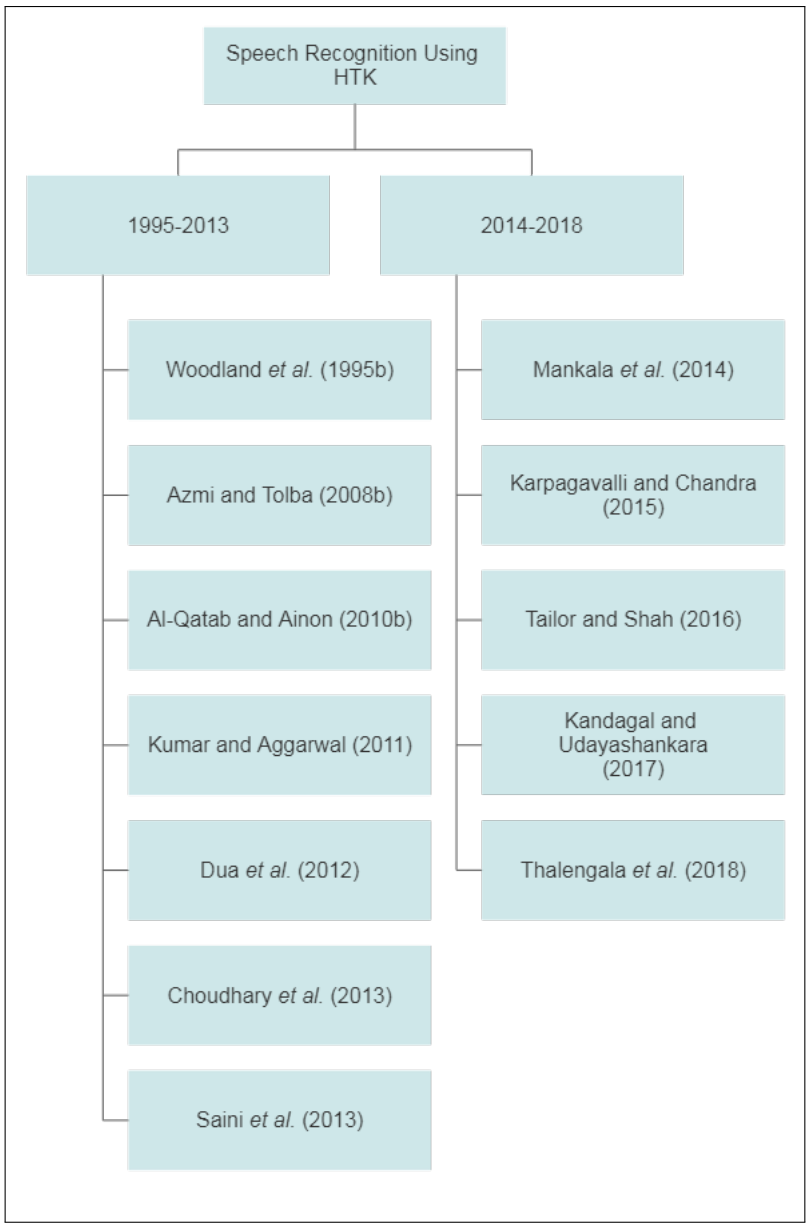


Figure 2.4: Organization of related work in speech recognition using HTK

Woodland *et al.* (1995) proposed HTK extensive vocabulary speech recognition system. The system used tied-state cross-word context-dependent mixture Gaussian HMMs and a dynamic network decoder that can operate in a single pass. The extension of MLLR has been used to estimate the parameters. The system has a vocabulary of up to 65 thousand words. The final acoustic models have been extended to be sensitive to more acoustic context (quin-phones). A 4-gram language model has also been used and unsupervised incremental speaker adaptation has been incorporated. The error rate has been reduced between 8.0% to 12.0 % for the database of 65 thousand words. Azmi and Tolba (2008) presented an improved automatic recognition system with a speech from a noisy background based on HMMs. The improvement has been made by changing acoustic units during the recognition process. Automatic Arabic speech recognition has been described by showing their constructed mono-phones, tri-phones and syllables. It has been a syllable based speaker-independent speech recognition system designed using HTK. Syllables outperform mono-phones and tri-phones by 21.5% and 15.6% respectively for the dataset of 59 Speakers.

Al-Qatab and Ainon (2010) proposed automatic speech recognition system for the Arabic speech based on HMMs. This system has been implemented using HTK and a comparison of mono-phone, tri-phone and word-based recognition with syllable based recognition has been performed. In the proposed system, MFCCs have been used as feature extraction technique. The overall performance of the system has been 90.6%, 98.0% and 98.0% for sentence correction, word correction and word accuracy, respectively, for the dataset of 13 speakers. Kumar and Aggarwal (2011) proposed connected-word speech recognition system for the Hindi language based on HMMs, using HTK toolkit on Linux platform. Training data has been collected from twelve speakers, including both males and females and data from five speakers has been collected for testing purpose. MFCCs have been used as a feature extraction technique. A vocabulary of the size of thirty words has been used for training purpose. An accuracy of 87.0% has been achieved for the dataset of 17 speakers. Dua *et al.* (2012) presented an automatic speech recognition system for isolated words for the Punjabi language, based on HMMs using HTK. Data has been collected from eight speakers for training consisting of one hundred fifteen distinct Punjabi words in a closed room environment and few samples have also been collected from six speakers in real-time environment for analysing the performance of the system. Data has been collected by them using audacity recording tool. A Graphical User Interface (GUI) has also been implemented in java language to make the system more interactive. For feature extraction, MFCC technique has

been used. An accuracy of 95.6% has been achieved in a class room environment for the dataset of 21 speakers.

Choudhary *et al.* (2013) proposed an automatic speech recognition system for Hindi language using HMM. The proposed system has been developed for recognising the isolated and connected words of Hindi speech and implemented using HTK. The system has been trained for hundred different isolated words and each word has been uttered ten times. Their system achieved a recognition rate of 95.0% and 90.0% at isolated and connected word level, respectively, in a closed room environment for the dataset of 100 distinct words. Saini *et al.* (2013) proposed Hindi speech recognition system based on HMMs using HTK in the Linux environment. Three different state HMM topology has been used by them. For recognising the speech, a word model has been used. For data parametrisation, MFCC features have been extracted. For training purpose, data has been collected from six speakers. An accuracy of 96.6% has been achieved for 10 states in HMM topology for the dataset of 118 isolated Hindi words spoken by 6 speakers. Mankala *et al.* (2014) proposed automatic speech recogniser for Telugu language using HTK. This has been developed for recognising isolated words using the acoustic word model. Data has been collected from 9 Telugu speakers for training purpose and the system has been trained using 113 isolated Telugu words. The overall accuracy of the system that has been observed is in the range of 95.5% and 96.6% for this dataset of isolated words spoken by 9 speakers.

Karpagavalli and Chandra (2015) proposed a phoneme and word based model for Tamil speech recognition using GMM-HMM. The recognition system that has been developed is speaker-independent isolated- phoneme and word recognition systems, and it has been developed using HTK with MFCC feature extraction technique. The vocabulary of 50 words has been used to collect the data from 10 native speakers. The performance of both phoneme and word based models have been analysed. The recognition accuracy of the models has been observed in the range of 80.0% to 83.3% for the word model and 90.0% to 95.0% for the phoneme model. Tailor and Shah (2016) proposed an ASR based on HMMs using HTK for Gujarati language. The data for this work has been collected from the persons in the age group of 18-36 years. There were a total of 6 speakers from whom the data has been collected. The performance of system has been measured in terms of Word Recognition Rate (WRR) and WER. The WRR reported in this paper is 95.9% and WER reported is 5.9% for the lab environment, and for the open noisy environment, these figures are 95.1% and 7.4%, respectively.

Kandagal and Udayashankara (2017) proposed a speaker independent speech recog-

nition using maximum likelihood approach for isolated words. Word and phoneme level acoustic modeling have been developed using stochastic procedure. Acoustic features have been estimated by MFCC and the performance has been reported for different sizes of vocabulary. WERs for phone and word acoustic models have been reported to be 97.6% and 94.7%, respectively, for the vocabulary of 90 words. It has also been reported that they tested the system at phone level for vocabulary of 70 words and achieved an accuracy of 98.1%. Thalengala *et al.* (2018) proposed isolated-word speaker-independent speech recognition system for Kannada language using HMM based on HTK. Performance of the system has been analysed for different acoustic models. The data has been collected from Kannada news channel. The system has used MFCC as a feature extraction technique and its derivatives as acoustic features whereas acoustical models have been developed using HMMs. Various mono-phone models, namely, word-level, syllable-level and phone-level have been considered in this work. The maximum word recognition accuracy of 67.8% and 70.6% have been reported for mono-phone and tri-phone based systems, respectively.

2.4 Review on Development of Phonetic Engine

In this section, a review on the development of PE has been carried. The main aim of this review is to collect all the information related to the development of PE for Indian languages. A number of approaches have been proposed in the field of segmentation, transcription, and feature extraction for different Indian languages. Figure 2.5 shows the organization of related work in the development of PE. Table 2.4 presents the summary of literature on the same.

Patil *et al.* (2012) addressed the phonetic transcription related issues for Gujarati and Marathi languages. They worked on the research issues like ambiguity between frication and aspirated plosive, and the effect of dialectal variations on phonetic transcription. Vachhani and Patil (2013) proposed the use of Perceptual Linear Prediction Cepstral Coefficients (PLPCCs) for phonetic segmentation task. Spectral Transition Measure (STM) has been used to detect phonetic boundaries in this work. An accuracy of 85.0%, and an over-segmentation rate 15.0% for automatic boundary detection of 2, 34 and 925 phone boundaries corresponding to 630 speakers of entire *TIMIT* dataset has been achieved by them.

Table 2.3: Summary of literature on speech recognition using HTK

S. No.	Reference	Feature Extraction Technique	Language	Application	Results
1	Woodland *et al.* (1995)	MFCC	English	Text-dependent Speaker Identification	The error reduction rate from 8.0% to 12.0 % has been observed for the dataset of 65 thousand words.
2	Azmi and Tolba (2008)	MFCC	Arabic	Automatic Speech Recognition	Syllables outperform monophones and triphones by 21.5% and 15.6%, respectively, for the dataset of 59 Speakers.
3	Al-Qatab and Ainon (2010)	MFCC	Arabic	Automatic Speech Recognition	The overall performance of the system has been 90.6%, 98.0% and 98.0% for sentence correction, word correction and word accuracy, respectively, for the dataset of 13 speakers.
4	Kumar and Aggarwal (2011)	MFCC	Hindi	Connected Word Speech Recognition	An accuracy of 87.0% has been achieved for the dataset of 17 speakers.
5	Dua *et al.* (2012)	MFCC	Punjabi	Isolated Word Speech Recognition	An accuracy of 95.6% has been achieved in a class room environment for the dataset of 21 speakers.
6	Choudhary *et al.* (2013)	MFCC	Hindi	Isolated and Connected Word Speech Recognition	The accuracy of 95.0% and 90.0% have been achieved for isolated and connected words, respectively, in a closed room environment for the dataset of 100 distinct words.
7	Saini *et al.* (2013)	MFCC	Hindi	Speech Recognition System	An accuracy of 96.6% has been achieved for 10 states in HMM topology for the dataset of 118 isolated Hindi words spoken by 6 speakers.

Table 2.3: Continued

S. No.	Reference	Feature Extraction Technique	Language	Application	Results
8	Mankala *et al.* (2014)	MFCC	Telugu	Automatic Speech Recognition	The overall accuracy has been observed in the range of 95.5% and 96.6% for the dataset of 113 isolated Hindi words spoken by 9 speakers.
9	Karpagavalli and Chandra (2015)	MFCC	Tamil	Phoneme and Word Based Model Speech Recognition	The recognition accuracy of the models have been observed between the range of 80.0% to 83.3% for word model and 90.0% to 95.0% for phoneme model for the vocabulary of 50 words.
10	Tailor and Shah (2016)	MFCC	Gujarati	Automatic Speech Recognition	The WRR has been observed as 95.9% and 95.1% in lab and noisy environment for the dataset of 6 speakers.
11	Kandagal and Udayashankara (2017)	MFCC	Kannada	Speaker Independent Speech Recognition for Isolated Word	WER for phone and word acoustic models have been 97.6% and 94.7% respectively for the vocabulary of 90 words whereas it has been 98.1% for 70 words at phone level.
12	Thalengala *et al.* (2018)	MFCC	Kannada	Isolated-word Speaker-independent Speech Recognition System	The accuracy of 67.8% and 70.6% for word recognition have been observed for mono-phone and tri-phone models, respectively, for the dataset collected from Kannada broadcasting news.

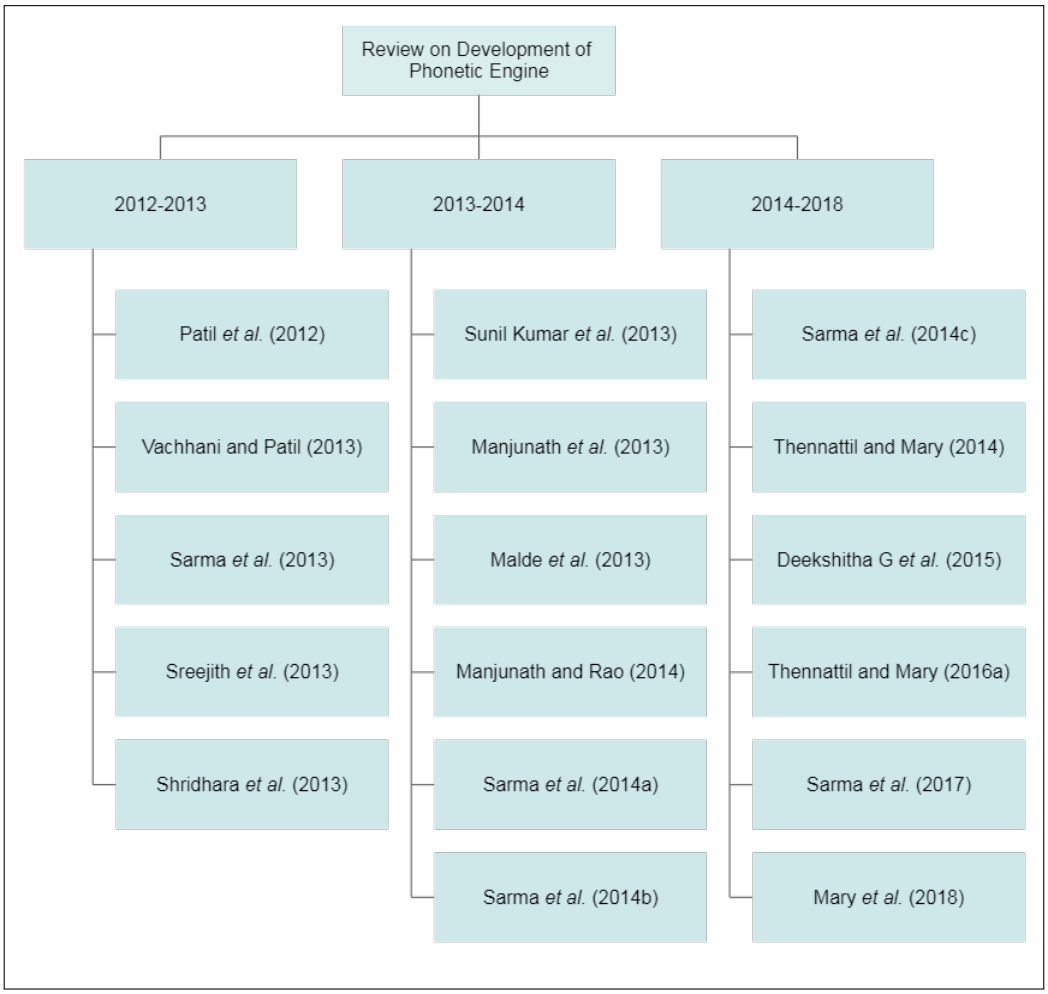


Figure 2.5: Organization of related work in the development of PE

Sarma *et al.* (2013) discussed PE development for Assamese language and discussed some issues related to it. They have implemented the proposed work using HTK toolkit and it is a phoneme based recognizer. The speech data for this work has been recorded in three different modes: (i) Read speech mode, (ii) Lecture speech mode, and (iii) Conversational speech mode. Read speech data has been used to train HMMs. The trained HMMs have then been used to derive a sequence of phonetic units from a test speech signal. They could achieve an accuracy of 47.3% for read speech mode data, of 45.3% for lecture speech mode data and 36.1% for conversational speech data, on the ASCII symbols. Confusion among the phonetic units specific to Assamese has also been discussed in this work. Along with this, issues related to different recording modes, language and native speaker dependencies. They have also collected the speech data for Hindi language from three different sets of speakers to study speaker, language and native dependencies. The accuracy of 40.5%, 36.1% and 29.6% have been achieved by them in native

speaker dependent, native speaker independent and non-native speaker independent cases, respectively.

Sreejith *et al.* (2013) proposed an approach for automatic labelling of prosodic events and discussed the implementation of broad class PE for Malayalam language. A baseline PE has been developed by them based on speech data which they collected from various regions of Kerala consisting of read speech, extempore and conversational speech. The proposed algorithm for automatic labelling has been used by them to capture the prosodic information like pitch variations and pauses in the speech utterance. They have further incorporated this prosodic information into the baseline PE. The accuracy of 80.0% and 77.0% have been achieved by them for stops and vowels labelling, respectively. Shridhara *et al.* (2013) presented the work carried out by them in order to collect the speech data in Kannada language for the development of prosodically guided phonetic search engine. The issues involved in transcription have also been discussed in this work. The speech corpus collected by them consists of data in three different contexts, namely, read mode, conversation mode and extempore mode. They also presented a four layered transcription, namely, phonetic transcription using IPA symbols, syllabification, pitch marking and break marking. In this work, the four-layer transcription has been performed for the entire collected data and a baseline recognition system has been built for Kannada language using HTK. They were able to achieve the phone recognition accuracy of 73.3%, 70.8% and 61.4% on 2 hours of data for the three modes, namely, broadcasting, lecture and conversational data, respectively. Kumar *et al.* (2013) developed Phonetic and Prosodically Rich Transcribed (PPRT) speech corpus of Bengali and Oriya languages, which provides phonetic and prosodic information. In the process of preparation of the speech corpus, they have collected 10 hours of read speech, 5 hours of conversation speech and 5 hours of extempore speech data. The collected data has then been transcribed using IPA to represent all possible phoneme variations. The prosodic information such as duration patterns of syllables, intonation patterns of phrases and break patterns within and across phrases have also been represented on the collected data in this work. After a careful analysis of the transcribed data, they observed that 45 IPA symbols out of 182 used symbols cover 87.6% of the transcription for 1752 Bengali sentences and 88.2% for 1823 Oriya sentences.

Malde *et al.* (2013) described the development of speech corpora for two Indian languages, namely, Gujarati and Marathi for the task of phonetic transcription. In this work, dialectal variations have also been analysed using spectrograms and phonetic transcription. It has been found by them that vowel sounds and plosive sounds

cover a large amount of dataset among 5 broad phonetic categories for both the languages. Manjunath *et al.* (2013) proposed a framework for the development PE for two Indian languages, namely, Bengali and Oriya. They proposed that the developed framework can be extended for any Indian language. For developing the PEs, they have used read speech corpus with 35 phones for Bengali language and 32 phones for Oriya language. The PE developed in their work uses HMMs and Feed Forward Neural Networks (FFNNs) with MFCCs as the features for building the models. In speaker dependent case, the accuracy of 41.6% and 53.9% have been achieved for Bengali PE using HMMs and FFNNs, respectively. Likewise for Oriya PE, the accuracy of 46.2% and 59.9% have been obtained using HMMs and FFNNs, respectively. Manjunath and Rao (2014) proposed the Automatic Phonetic Transcription (APT) system for read, extempore and conversation modes of speech data for Bengali language. They also calculated that the framework of deriving APT can be used for any Indian language after applying some extensions. In this work, separate APT systems have been developed for read, extempore and conversation modes of speech using 35, 33 and 30 phones, respectively. APT system has been developed using HMMs and FFNNs. Again, MFCCs have been used as the features for building the models. The accuracy of 41.6%, 29.2% and, 23.5% have been achieved in this work using HMMs for read, extempore and conversation modes, respectively. While using FFNNs, they have been able to achieve the accuracy of 53.9%, 46.2% and 33.6% for read, extempore and conversation modes, respectively. Sarma *et al.* (2014a) proposed a semi-automatic process of pitch marking for prosodic analysis of a speech corpus of Assamese language. To accomplish semi-automatic pitch marking, a method for automatic segmentation speech is described in this work, into certain regions where there is a continuous pitch contour and nature of pitch change within those regions is marked. In order to segment the speech into two segments of voiced and unvoiced, Zero frequency filtering was used. They have measured the height value of pitch contour in the final segment and accordingly done the marking using proposed method. Further there has been manual marking and correction in the results produced by the proposed method by deleting the incorrect segmentation and subtitling or inserting the correct marking. They have measured the performance of the proposed method with respect to the number of deletions, insertions, shifts and substitutions done. Sarma *et al.* (2014b) also proposed a semi-automatic method for syllable labelling of speech utterances of Assamese language using HMMs, Vowel Onset Point (VOP) and Vowel End (or offset) Point (VEP). They have used 15 broad classes and built HMM models them. They have used forced alignment procedure of HMM models to obtain the time label of

the transcription. Further in this work, the word transcription has been converted to syllable transcription with the help of a parser using certain syllabification rules. In order to obtain the time label of the syllables, this syllable transcription and the time label of the phones have then been used. They have also presented the refined syllable labelling output which has been done using the knowledge of VOP and VEP derived from the speech signal with the help of different signal processing techniques. They have achieved an accuracy of 93.4% in syllable labelling for 50 sentences of Assamese language. Sarma *et al.* (2014c) further proposed an automatic transcription of Assamese speech using HTK. In this work, the speech files has been manually transcribed using IPA symbols and ASCII symbols have been chosen for automatic transcription. An accuracy of 65.7% has been achieved for 3 hours of extempore speech data with 38 ASCII phones.

Thennattil and Mary (2014) proposed a method to improve the performance of the real time PE. A front-end has been developed by them for automatically segmenting long test-speech utterances of Malayalam language to short segments. This was done by automatically detecting pauses using a FFNN designed for speech/non-speech classification. They observed that the PE with this segmentation front end performs 10.0% to 20.0% better depending on the mode of speech and silence content of test-speech data. Deekshitha *et al.* (2015) proposed a method for segmentation of continuous speech of Malayalam language into phrase-like regions to be used in broad PE. A frame level speech/non-speech classifier using ANN has been employed in this work for detection of pause/break regions in continuous speech. Automatic marking of breaks in continuous speech has been achieved using this approach. Performance of broad phonetic engine in Malayalam has been evaluated with and without segmentation of input speech, and they have observed an improvement in the performance of the model when segmented data has been used. They have used the segmented data and the correctness of 87.9% for male and 88.1% for female speech data have been observed on 43 speech files. A method for searching audio database has been proposed by them using the output of this broad phonetic engine.

Thennattil and Mary (2016) described the work carried out for the implementation of a PE for continuous speech of Malayalam language, which is speaker-, gender-, and domain-independent. For this work, a speech database has been collected, which consists of speech in three modes: (i) Read mode, (ii) Lecture mode, and (iii) Conversation mode. This data has then been manually transcribed using IPA. By analysing the transcription, they have mapped the 40 frequently occurring phonemes to 26 phonemes, which were then modelled using continuous HMMs. The perfor-

mance of this PE has been evaluated by them using 75 minutes of speech data and it has been observed that the accuracy has been increased from 26.3% to 40.9% with decrease in number of phonemes from 40 to 26. They have also developed a GUI for the PE to perform real-time recognition of speech. Sarma *et al.* (2017) presented an analysis on different aspects of speech-to-text processing, starting from building a speech corpus, defining syllable rules, and finally developing a speech search engine of Assamese language. About 20 hours of speech in three modes, namely, read, extempore and conversation, has been collected and manually transcribed. Issues and challenges faced during development of the corpus have also been discussed by them. An automatic syllabification model has been presented by them which was developed with 11 rules for Assamese language with an accuracy of more than 95.0%. They have seen that there were 12 different syllable patterns and 5 among them were most frequent patterns. DNN has been used for speech recognition model, and they obtained an accuracy of 78.1% for automatic transcription of Assamese speech.

Mary *et al.* (2018) proposed a three step method for automatic syllabification for Malayalam and Bengali speech data. They detected non-speech and speech regions in order to mark silence and non-silence regions in speech. Then the Hilbert Envelope of LP residual has been used to locate VOPs. An indication on location of syllable boundaries has been given with respect to existence of more than one VOP for the region of continuous speech. The fixing of valley threshold has been performed by them in order to get the additional syllable boundaries from STE contour. This fixing has been done of values which was equal to the mean value of STE between two consecutive syllable boundaries of each speech region. They have evaluated this method for 50 sentences each in read, extempore and conversational mode speech of Malayalam and Bengali languages. with reference to manually marked syllable boundaries, an accuracy of 80.0% has been obtained with $\pm$ 50ms tolerance for this database. They showed that their method also gives a good accuracy on *TIMIT* and *NTIMIT* datasets without tuning thresholds and other parameters.

Table 2.4: Summary of literature on development of PE

S. No.	Reference	Supporting Tools/Techniques	Feature Extraction Technique	Language	Application	Results
1	Patil *et al.* (2012)	WaveSurfer	-	Gujarati and Marathi	Phonetic Transcription	Research issues like ambiguity between frication and aspirated plosive, and the effect of dialectal variations on phonetic transcription has been addressed.
2	Vachhani and Patil (2013)	STM	PLPCC	English	Phonetic Segmentation	An accuracy of 85.0% and an over-segmentation rate of 15.0% for automatic boundary detection of 2, 34 and 925 phone boundaries corresponding to 630 speakers of entire *TIMIT* dataset has been achieved.
3	Sarma *et al.* (2013)	HTK	MFCC	Assamese	Phonetic Engine	The accuracy of 47.3% in read speech mode, 45.3% in lecture speech mode and 36.1% in conversation speech mode has been achieved. Issues related to different recording modes, language and native speaker dependencies have been discussed. The accuracy of 40.5%, 36.1% and 29.6% have achieved in native speaker dependent, native speaker independent and non-native speaker independent cases, respectively, for dataset collected from Hindi speakers.

Table 2.4: Continued

S. No.	Reference	Supporting Tools/Techniques	Feature Extraction Technique	Language	Application	Results
4	Sreejith *et al.* (2013)	HTK	MFCC	Malayalam	Automatic Labelling of Prosodic Events and Baseline PE	Maximum accuracy of 80.0% and 77.0% have been achieved for stops and vowels, respectively, for 64 mixture model as compared to other classes for 2 hours of data.
5	Shridhara *et al.* (2013)	HTK	MFCC	Kannada	Four Layered Transcription and Automatic Speech Recognition	The phone recognition accuracy of 73.3%, 70.8% and 61.4% have been achieved for 2 hours each of broadcasting, lecture and conversation phonetic transcription respectively.
6	Kumar *et al.* (2013)	-	-	Bengali and Oriya	PPRT Speech Corpus	It has been shown that 45 IPA symbols out of 182 used symbols covers 87.6% of symbols for 1752 Bengali sentences and 88.2% for 1823 Oriya sentences.
7	Manjunath *et al.* (2013)	HMM and FFNN	MFCC	Bengali and Oriya	Phonetic Engine	In speaker dependent case, the accuracy of 41.7% and 53.9% have been obtained for Bengali language, and 46.2% and 59.9% for Oriya language using HMMs and FFNNs, respectively.

Table 2.4: Continued

S. No.	Reference	Supporting Tools/Techniques	Feature Extraction Technique	Language	Application	Results
8	Malde *et al.* (2013)	WaveSurfer	-	Gujarati and Marathi	Speech Corpora for Phonetic Transcription	Dialectal variations has been analysed using spectrograms and phonetic transcription. In addition, it has been found that for consonant sounds, plosive sounds are having large coverage in broad phonetic category.
9	Manjunath and Rao (2014)	HMM and FFNN	MFCC	Bengali	Automatic Phonetic Transcription	The performance accuracy using HMMs for read, extempore and conversation modes have been 41.7%, 29.2% and 23.5%, respectively, whereas using FFNNs, it has been 53.9%, 46.2% and 33.6%, respectively.
10	Sarma *et al.* (2014a)	ZFF	-	Assamese	Semi-automatic Process of Pitch Marking for Prosodic Analysis	9.4%, 7.2%, 6.9% and 18.9% of segment boundaries have been deleted, inserted, shifted and substituted, respectively, in the manual correction process for tha data of 30 speakers containing 100 speech sentences with 700 pitch contour segments.
11	Sarma *et al.* (2014b)	HMM and VOP/VEP	-	Assamese	Semi-automatic Syllable Labelling	The detection rate of 93.4% and average deviation of 12 has been observed for 50 sentences.
12	Sarma *et al.* (2014c)	HTK	MFCC	Assamese	Automatic Transcription	An accuracy of 65.7% has been achieved for 3 hours of extempore data.

Table 2.4: Continued

S. No.	Reference	Supporting Tools/Techniques	Feature Extraction Technique	Language	Application	Results
13	Thennattil and Mary (2014)	HTK and ANN	MFCC, STE, SFM and MDF	Malayalam	Phonetic Engine with Segmentation	Improvement of 10.0% to 20.0% has been observed according to the mode of speech and silence content of test data.
14	Deekshitha *et al.* (2015)	HMM and ANN	STE, SFM and MDF	Malayalam	Segmentation of Continuous Speech	The correctness of 87.9% for male and 88.1% for female speech data have been observed on 43 wave files.
15	Thennattil and Mary (2016)	HTK	MFCC	Malayalam	Phonetic Engine for Continuous Speech	An overall phoneme recognition correctness of 44.8% and an improved recognition accuracy of 40.9% has been achieved after mapping 40 symbols to 26 symbols based on their occurrences for 1 and half hours of read speech.
16	Sarma *et al.* (2017)	HTK	MFCC	Assamese	Analysis on Speech-to-Text Processing	The accuracy of 78.1% with 38 graphemes using HTK 3.5 with DNN and 65.7% using HTK 3.4 have been achieved for 3.5 hours of speech data.
17	Mary *et al.* (2018)	VOP	STE	Malayalam and Bengali	Three Step method for automatic syllabification	Overall accuracy of 80.0% has been obtained with $\pm$ 50 ms tolerance with reference to manually marked syllable boundaries for 50 sentences.

2.5 Review on Speaker Classification Techniques

In this section, a review of the existing machine learning techniques for speaker classification has been presented. The objective of this section is to critically examine existing machine learning techniques that can be applied to the speaker classification process. Figure 2.6 shows the organization of related work on speaker classification techniques.Table 2.5 shows the summary of literature on speaker classification techniques used by various researchers.

Adami *et al.* (2003) utilized DTW to evaluate the distance between words and templates from test message. Prosodic contours for text-independent speaker verification have been proposed by them and reduction in Equal Error Rate (EER) by 3.4% has been observed on *NIST* dataset for a system based on short-term pitch and energy features alone. Solomonoff *et al.* (2005) implemented pattern classification using Support Vector Machines (SVMs). In this work, SVM mapped the inputs into a high-dimensional space and separated classes with a hyperplane. They specified the role of SVMs here to successfully design the kernel and inner product, induced by higher dimensional mapping and ultimately classifying the speakers. A reduction in EER by 20.0% below baseline for male speakers and 30.0% for female speakers has been observed on *switchboard 2* dataset. You *et al.* (2009) implemented an integrated speaker classification system using SVMs and Gaussian Mixture Models (GMMs). In this work, GMMs have been used to characterize the voice of a speaker with parameters such as covariance matrices, mean vectors, and mixture weights. In the proposed technique, conventional Kullback-Leibler (KL) kernel has been used to limit the adaptation of GMM to mean value and they kept the covariance unchanged. GMM-UBM Mean Interval (GUMI) concept has been introduced by them based on the *Bhattacharyya* distance and it leads to a new kernel for SVM classifier. Experiments in this work have shown that GUMI kernel has outperformed KL kernel for *NIST-SRE* dataset

Chen and Salman (2011) extracted bottleneck features at frame level using DNN. They used these features as the input for speaker classification. The Deep Neural Architecture (DNA) has been presented to design a speaker specific representation of six different English datasets. In this work, representation learned by DNA captures the intrinsic speaker-specific characteristics and generally outperform MFCCs by incorporating a frame based speaker modeling technique. Abdel-Hamid and Jiang (2013) proposed a hybrid speech classification method based on joint learning. It has been applied by them for classification of large number of speakers as well as multiple small number of speakers. In this method, training data along with

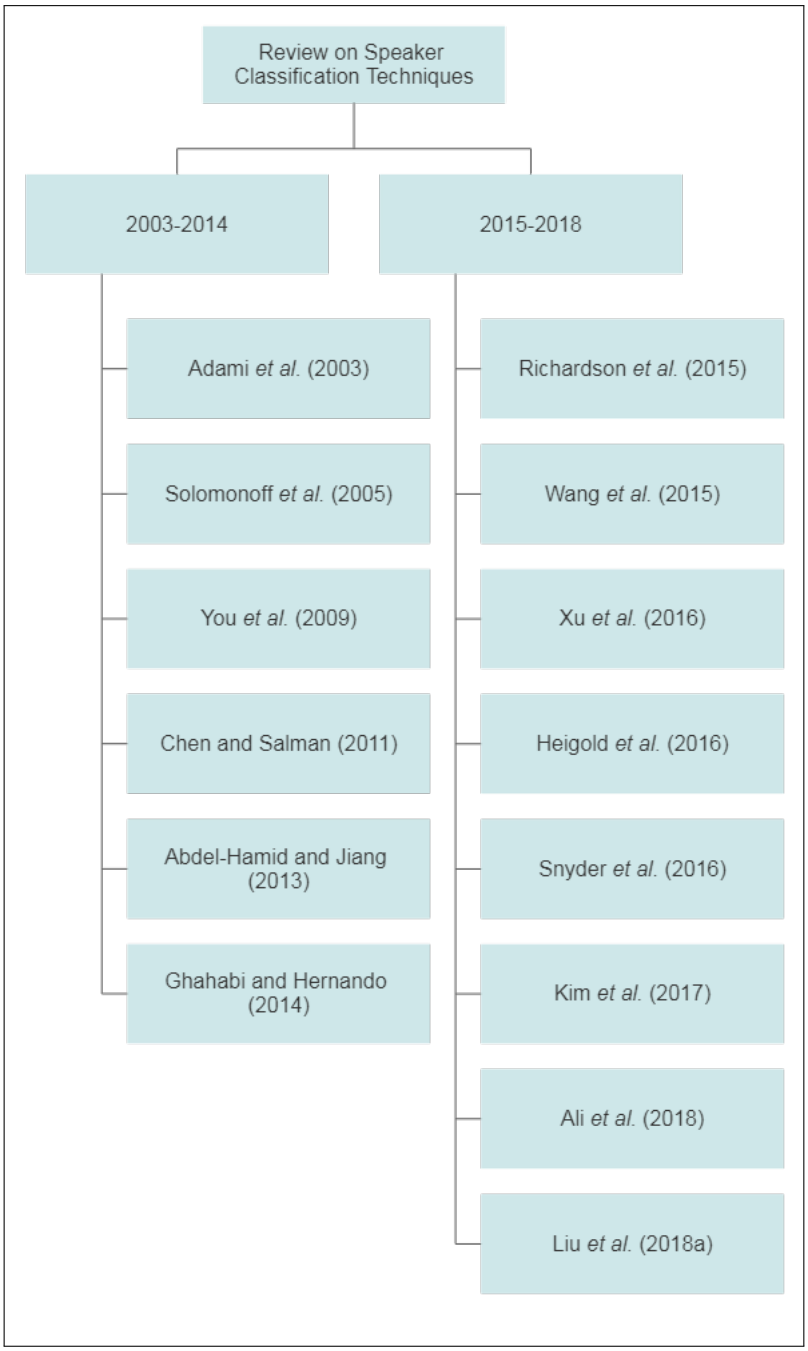


Figure 2.6: Organization of related work in speaker classification techniques

speaker labels, has been used by joint training technique. Then, updating of the speaker codes and adaptation neural weights based on back-propagation algorithm has been carried out by them. For small speaker code, features of each speaker have been transformed through learned adaptation neural network into a generic speaker-independent feature space in this work. The authors have observed that there has been a relative reduction of over 10.0% in phone error rate for 7 utterances from

TIMIT dataset. Ghahabi and Hernando (2014) used back-propagate class errors up to first layer to train the network. In the technique proposed by them, Deep Belief Networks (DBNs) have been used to model and target i-vectors in a speaker verification task. An impostor selection approach has also been utilized by them which helps deep belief networks to outperform the cosine distance classifier. Performance has been evaluated using the EER and the minimum Decision Cost Function (minDCF). A relative improvement of 10.0% in EER and 8.0% in minDCF has been observed by them for *NIST-SRE* 2006 corpora. A deep neural network classifier has been trained by Richardson *et al.* (2015) for maping frame-level features in a certain context relative to the speaker identity target. They extracted a feature vector, referred to as a deep vector or "d-vector", from a utterance of speaker by averaging the activations taken from the last hidden layer of DNN. They called the method as the d-vector front-end Method. After extraction of features from a front-end, a speaker verification from back-end, to build speaker models for classification, has been used by them . A reduction in EER by 55.0% has been observed by them for the out-of-domain condition on *switchboard* dataset. It has been proposed by Wang *et al.* (2015) that there is also building of a speaker space development stage from development data, where each speaker performs like a coordinate axis of the space. The enrolment is given to the test stage and test speaker models the trial from the speaker space in this work. The evaluation of similarity of the two models by a classifier is then performed. A SVM classifier is latter on trained for the purpose of distinguishing true speakers from impost speakers with the attribute projection of new sense for compensating session variability. They observed that the recognition rate of the proposed system has been improved by 4.2% from pyknogram-based system for *CHAIN* corpus. Xu *et al.* (2016) proposed the speaker models extracted from an i-vector based front-end and then Probabilistic LDA (PLDA) has been used as the classifier. A novel algorithm Eigen Decomposition Like Factorization (EDLF) has also been proposed by them for classification. they have observed that EDLF has performed well for speaker identification on 37800 sentences. Heigold *et al.* (2016) implemented an integrated approach using RNNs and DNNs for speaker classification. This approach maps the test and reference utterances directly into a single score for evaluation and then it jointly optimizes the components of system using same evaluation protocol. It has been observed by them that the EER has been reduced by 1.4% using a RNN instead of a simple DNN for an utterance 'OK Google' spoken by 4000 speakers.

Snyder *et al.* (2016) utilised the DNNs which took different length speech segments and mapped it to a speaker embedding. A remarkable performance has been seen

by them as compared to an i-vector based speaker classification variants. This system outperformed i-vector baseline system, on an average, by 13.0% and 29.0% pooled EER on telephone conversation data. Kim *et al.* (2017) explored the training of DNNs for a different task, *e.g.* speech classification, where speech frames' posterior probability is generated as speaker classification alternative of supervised leaning, and i-vectors are extracted from speaker classification based on DNN by using factor analysis. The method has been denoted as the deep neural network/i-vector front-end. The proposed system has been observed to be more stable when a series of experiments were performed on both 20 dysarthric and 10 control speakers of Korean language. Ali *et al.* (2018a) used DBN to convert audio signals into vectors. They proposed that these vectors represent audio signals of different lengths and it is easier to build a model using these vectors. In this work, these vectors were obtained from mixture of DBN features. An accuracy of 92.6% has been achieved when DBN has been combined with MFCCs and other set of features for data from 10 speakers. Liu *et al.* (2018a) proposed a novel speaker classification system where sufficient statistics has been extracted for state-of-the-art i-vector model through the training by DNN. It has been observed by them that the DBN has been a better replacement of GMMs in order to generate frame alignments. They observed an EER reduction from 4.9% to 2.5% when the proposed technique has been tested on a dataset involving 50 speakers.

2.6 Speaker classification using Deep Learning

Deep learning models have been extremely successfully in speech recognition, speaker classification, speech synthesis, imaging processing, computer vision, natural language processing (NLP) and understanding (NLU). Deep learning has been the used in these applications, unless the data size is too small for deep models. In this section, a survey on the use of deep learning models for speaker classification has been performed.

LeCun *et al.* (2015) presented a good introduction to deep learning and its use in speaker recognition. It states that deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. Apart from speech recognition, deep learning models can also be used in visual object recognition, object detection and many other domains such as drug discovery and genomics. It also states that deep learning is well suited for large data set where it discovers the structure using the back propagation algorithm. Latif *et al.* (2020) focuses on an intermediate representation of the input

signal which will can be produced automatically to suit the task at hand in a better way to improve the performance. With advances in deep learning, representations have become more useful and less dependent on human knowledge, making it very conducive for tasks like classification, prediction, etc.

Mu and Zeng (2019) have presented a exhaustive review on the research done on the deep learning technique.They claim that deep learning technology has become an important research direction in the field of machine learning, which has been widely applied in the image processing, natural language processing, speech recognition and online advertising and so on. They have introduced various applications, frameworks and platforms of deep learning. Along with that, they have also introduced several common models of deep learning and optimization methods. The applied deep learning practices in the field of speaker recognition, both verification and identification are well presented by (Sztahó *et al.*, 2019). They also claim that deep learning techniques have done advancements in most machine learning fields and becomes the now state-of-the-art solution for both speaker verification and identification. The increasing amount of gathered data opens up the territory to Deep Learning, where they are the most effective.

Tirumala and Shahamiri (2016) states that Deep Learning approaches has shown success in speech recognition and speaker identification over traditional approaches such as those that use MFCCs for feature extraction with GMMs. They presented a review of the Deep Learning methodologies used for speaker identification and surveys important Deep Learning algorithms that can potentially be explored for future works.

Ali *et al.* (2018b) proposed an approach to combine the learned features and the MFCC features for speaker recognition task, which can be applied to audio scripts of different lengths. They showed in their paper that the audio word count vectors generated from mixture of deep belief network features at different layers give better performance than the MFCC features. Chung *et al.* (2018) made some key contributions as they introduced a very large-scale speaker recognition dataset collected from open-source media containing over a million utterances from over 6,000 speakers. Also, they developed and compared Convolutional Neural Network models and training strategies that can effectively recognise identities from voice under various conditions. The models trained on their dataset surpasses the performance of previous works on a benchmark dataset by a significant margin.

Yu and Li (2017) summarized recent progresses that has been made in deep learning based acoustic models. They have also discussed about recurrent neural networks and convolutional neural networks that these can be effectively exploit vari-

able length contextual information, and their various combination with other models. They further illustrated robustness issues in speech recognition systems, and modeling techniques that lead to more efficient decoding. Amodei *et al.* (2015) showed that an end-to-end deep learning approach can be used to recognize either English or Mandarin Chinese speech, two being vastly different languages. Key to their approach is the application of HPC techniques, resulting in a 7x speedup over their previous system. Their system is competitive with the transcription of human workers when benchmarked on standard datasets.

2.7 Summary

This chapter focused on the available literature in the area of tools and techniques used for speech recognition and speaker classification tasks. The review on five different aspects, namely, usage of MFCCs as feature extraction, speech processing using HMMs, development of ASR systems using HTK, process for the development of PE and existing techniques used for speaker classification has been carried out. It can be concluded from the survey that MFCCs are performing well as a feature extraction technique and HTK has been used widely in speech recognition tasks.

After careful analysis of the available literature, a PE has been proposed in this thesis, which explores prosodic aspects of Punjabi language. The complete development process is explained in the next two chapters. Along with this, to overcome the issues of existing speaker classification techniques, an Ensemble-based Quantum-Neural Network and Crossover based Particle Swarm Optimization with SVM have been proposed in Chapter 5.

Table 2.5: Summary of literature on speaker classification techniques

S. No.	Reference	Classification Technique	Feature Extraction Technique	Language	Application	Results
1	Adami *et al.* (2003)	DTW	MFCC	English	Prosodic Contours for Text-independent Speaker Verification	An EER of 3.7% has been achieved on extended NIST dataset, which has been relative improved by 77.0% over a system based on short-term pitch and energy features alone.
2	Solomonoff *et al.* (2005)	SVM	LPCC and MFCC	English	Channel Compensation for Speaker recognition	Experiments has shown significant improvement in performance for the method applied on subset of the switchboard 2 corpus.
3	You *et al.* (2009)	SVM and GMM	LPCC	English	GUMI based New Kernel for SVM	Experiments has shown that GUMI kernel has outperformed KL kernel for *NIST-SRE* dataset.
4	Chen and Salman (2011)	DNA	MFCC	English	Speaker-specific Over-complete Representation	Representation learned by DNA can capture intrinsic speaker-specific characteristics and generally outperform MFCCs by incorporating a state-of-the-art speaker modeling technique.
5	Abdel-Hamid and Jiang (2013)	NN-HMM	MFCC	English	Fast Speaker Adaptation	By using only 7 utterances for adaptation, Relative reduction of over 10.0% has been seen in phone error rate on *TIMIT* dataset.
6	Ghahabi and Hernando (2014)	DBN	Frequency Filtering	English	DBN based Speaker Recognition	Relative improvement of 10.0% in EER and 8.0% in minDCF has been observed for *NIST-SRE* 2006 corpora.

Table 2.5: Continued

S. No.	Reference	Classification Technique	Feature Extraction Technique	Language	Application	Results
7	Richardson *et al.* (2015)	DNN	MFCC and PLP	English	Speaker Identification With Whispered Speech	A 55.0% reduction in EER for the out-of-domain condition has been achieved for switchboard data.
8	Wang *et al.* (2015)	SVM	MFCC and LPCC	English	Speaker and Language Recognition	The recognition rate of the proposed system has been improved by 4.2% from pyknogram-based system for *CHAIN* corpus.
9	Xu *et al.* (2016)	EDLF	MFCC	English and Chinese	I-vector based Speaker Identification	EDLF has performed good for speaker identification on 37800 sentences.
10	Heigold *et al.* (2016)	RNN/LSTM	FFT and PLP	English	Text-dependent Speaker Verification	The EER has been reduced to 1.4% using a RNN instead of a simple DNN for an utterance 'OK Google' spoken by 4000 speakers.
11	Snyder *et al.* (2016)	DNN	MFCC	English	Speaker Embeddings for End-to-end Speaker Verification	This system outperformed i-vector baseline by 13.0% average and 29.0% pooled EER on telephone conversation data.
12	Kim *et al.* (2017)	KL-HMM	MFCC	Korean	Speaker Embeddings for End-to-end Speaker Verification	The proposed system has been observed to be more stable when a series of experiments have been performed on both 20 dysarthric and 10 control speakers.

Table 2.5: Continued

S. No.	Reference	Classification Technique	Feature Extraction Technique	Language	Application	Results
13	Ali *et al.* (2018a)	DBN and SVM	MFCC	Urdu	Speaker Recognition with Hybrid Features	The accuracy of 92.6% has been achieved when DBN has been combined with MFCC and other set of features for data from 10 speakers.
14	Liu *et al.* (2018a)	GMM and CNN	MFCC	Chinese	Speaker Recognition with Hybrid Features	The EER reduction from 4.9% to 2.5% has been observed when the proposed technique has been tested on data from 50 speakers.

Chapter 3

Data Collection and Prosody Marking

Development and availability of spoken language corpora in regional languages is of utmost importance for speech recognition. Collection of speech data of Punjabi language for prosody based Phonetic Engine (PE) and prosody marking is explained in this chapter. The speech corpus is developed which consists of data in three different contexts, namely, read mode, lecture mode, and conversation mode. A four layered transcription, namely, phonetic transcription using IPA symbols, break index marking, pitch accent marking and syllabification (also called as syllable labeling) has been performed. Prosodic knowledge is incorporated in the PE and explained in next chapter.

3.1 Data Collection

Data is collected from different regions of Punjab in order to capture all the dialectal variations of Punjabi language. Initially, seven persons were selected who had good command on Punjabi language. Thereafter, their speeches were recorded at different time and also at varying microphone distance. At least thirty different samples of the same person have been recorded. Eventually, data has been collected from people with varying command on Punjabi language as well as from different sources. The details of data collection are shown in Figure 3.1. Data has been collected in three different modes, namely, read speech mode, lecture speech mode and conversational speech mode.

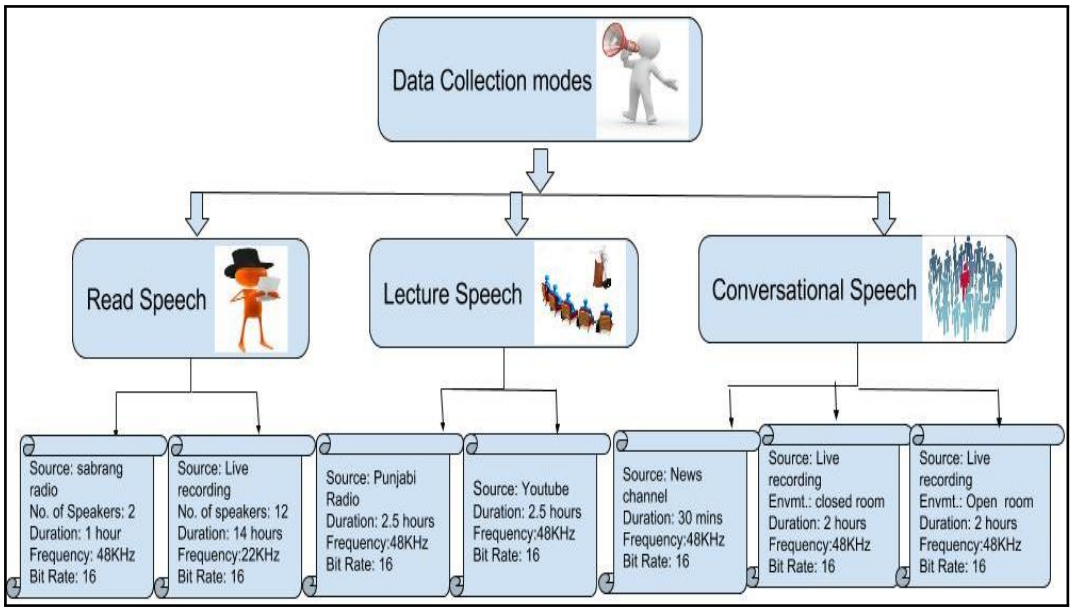


Figure 3.1: Data collection modes

3.1.1 Read Speech Mode

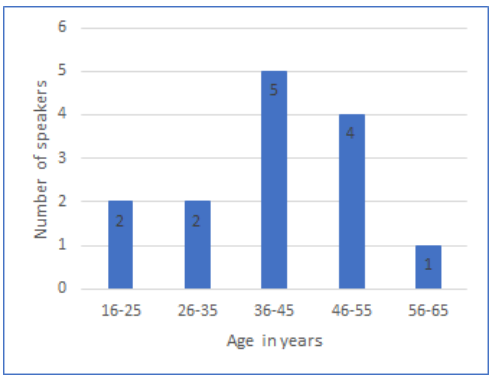
In order to collect the speech data for the read speech mode, we have followed two methods. In the first method, one hour of data has been downloaded from the radio channel, *Sabrang* Radio. In the second method, the speakers were given a written text and they were asked to speak this text and their speech was recorded individually. Here, data has been recorded in a normal room environment using a microphone channel maintained at a sampling frequency of 22 KHz. The zoomH4 device has been used to record this data. The metadata for this mode of data is given in Table 3.1.

3.1.2 Lecture Speech Mode

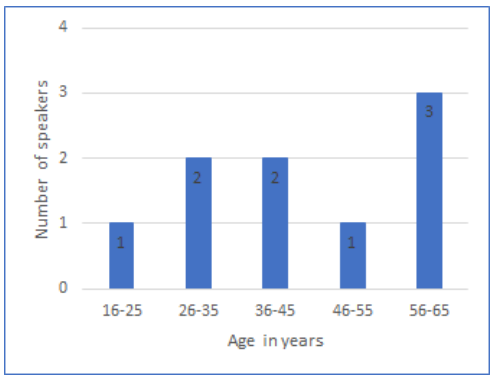
For lecture speech mode, a speaker presents his own ideas in the form of speech on a particular topic. The data for the lecture mode has been taken from the radio channel, Punjabi Radio USA and also from YouTube. The recording of this data has been done with a sampling frequency of 48 KHz and a bit rate of 16 bits per sample. Total time of collected data for this mode is 5 hours. The metadata for this mode of data is given in Table 3.2.

3.1.3 Conversational Speech Mode

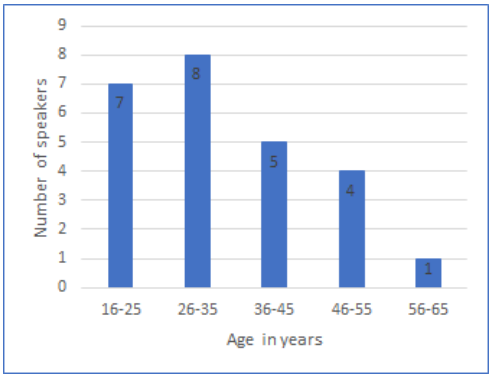
In this mode of data collection, multiple speakers were given a topic to discuss and all of them had to present their ideas in a group discussion. Thirty minutes of data for this mode has been taken from a Punjabi news channel. This data has a sampling frequency of 48 KHz and a bit rate of 16 bits per sample. Besides this, data has also been recorded by native Punjabi speakers for a duration of 4 hours. Out of these 4 hours, 2 hours of data has been recorded in a normal room environment and other 2 hours of data is collected in an open environment using a microphone channel maintained at a sampling frequency of 48 KHz. A total of 4 hours and 30 minutes data has been collected for this mode. The metadata for this mode of data is given in Table 3.3. Figure 3.2(a) to 3.2(c) shows the age group of speakers in read speech, lecture speech and conversational speech, respectively.



(a) Read speech mode



(b) Lecture speech mode



(c) Conversational speech mode

Figure 3.2: Distribution of speakers' age in (a) Read speech mode, (b) Lecture speech mode, and (c) Conversational speech mode

Table 3.1: Characteristics of read speech mode data

Date Type	Data Source	Recording Environment	Sampling Frequency (in KHz)	Bit Rate (in bits per sample)	Number of Speakers		Total Collected Data (in hours)
					Male	Female	
Primary Data	Live Recording	Closed Room	22	16	5	7	14
Secondary Data	*Sabrang* Radio	Studio Room	48	16	1	1	1

Table 3.2: Characteristics of lecture speech mode data

Date Type	Data Source	Recording Environment	Sampling Frequency (in KHz)	Bit Rate (in bits per sample)	Number of Speakers		Total Collected Data (in hours)
					Male	Female	
Secondary Data	Punjabi Radio	Studio Room	48	16	3	1	2.5
Secondary Data	Youtube Channel	Studio Room	48	16	3	2	2.5

Table 3.3: Characteristics of conversational speech mode data

Date Type	Data Source	Recording Environment	Sampling Frequency (in KHz)	Bit Rate (in bits per sample)	Number of conversations	Number of Speakers		Total Collected Data (in hours)
						M	F	
Primary Data	Live Recording	Closed Room	48	16	2	5	6	2
Primary Data	Live Recording	Open Room	48	16	7	6	2	2
Secondary Data	News Channel	Closed Room	48	16	1	5	1	0.5

3.2 Unique Phonetic and Prosodic Features of Punjabi Language

Punjabi belongs to the Indo-Aryan family and is a tonal language. Written form of Punjabi relies on Gurmukhi script, and spoken form relies heavily on Sanskrit vocabulary. It is known to possess 13 dialects. Majhi is the most popular dialect spoken widely in many districts of Pakistan's Punjab province and many districts in the state of Punjab in India. Apart from Majhi, the other dialects, namely, Malwai, Doabi and Puadhi are spoken in the state of Punjab in India and Shahpuri, Jhangochi, Jangli, Pothohari, Hindko, Dhani, Jafri, Chenavari, Saraiki dialects are spoken in Pakistan's Punjab province. People write Gurmukhi script from left-to-right and also spell it phonetically. There are 35 consonants in Gurmukhi script wherein, 3 distinct consonants form the basis for vowels. Along with these, there are 6 special consonants with a dot at the foot, 9 vowel diacritics, 3 auxiliary diacritics and 3 subscript letters in the Gurmukhi script (Appendix A) (Dhanjal and Bhatia, 2013). Based on the tones of the language, phonological and prosodic features are discussed below (Lata *et al.*, 2015).

3.2.1 Phonological Features

Phonological features deals with the arrangement of sounds in a language. The basic phonological features of Punjabi language include conjunct consonants, diphthongs, geminates, prolative vowels and nasalization. All of these are explained below:

Conjunct Consonants

Conjunct consonants are the type of letters, used to write consonant clusters. The modified form of the second consonant, called as subscript letter in the Punjabi language, sometimes referred to as 'half-consonant', is sub-joined to the first unaltered full consonant letter. There are three subscript letters which contributes to conjunct consonants, namely, ੍ਹ /h/, ੍ਰ /r/ and ੍ਵ /v/. Examples of these conjunct consonants are shown in Table 3.4 below.

Table 3.4: Examples of conjunct consonants in Punjabi language

Punjabi Word	Transliteration	IPA Transcription	Meaning in English
ਚੜ੍ਹ	charh	$c^{h}ə\ɾ$	climb
ਪ੍ਰਥਮ	pratham	$pɾt^{h}m$	First
ਸ੍ਵਰਗ	swarg	svərəgə	Heaven

Diphthongs

A diphthong is a combination of two adjacent vowel sounds within the same syllable. It has been observed that in Punjabi language, the first member of diphthong is always a short vowel whereas the second member is always a long vowel. There are six diphthongs in Punjabi language as listed in Table 3.5 with example of each pair.

Table 3.5: Examples of diphthongs in Punjabi language

Diphthong	Example in Punjabi	Transliteration	IPA Transcription	Meaning in English
ਇ +ਓ	ਦਿਓ	deo	dɪo	Give
ਇ +ਅੌ	ਧਿਆਉਣਾ	dheaona	tɪɔɳaː	Worship
ਅ +ਈ	ਕਈ	kai	kəi	Many
ਅ +ਏ	ਪਏ	pae	pəe	Placed
ਅ +ਊ	ਗਊ	gau	gəu	Cow
ਉ +ਆ	ਗੁਆਚਾ	guacha	gUvatʃa	Not found

Geminates

The use of a diacritic called as *addak* (ੱ) on a consonant can make it geminated. When *addak* is placed on the previous character, the subsequent character is pronounced as full plus half of its sound. For example: ਦੁੱਧ / duddh / dʊd^{h} / milk. Gemination, with the use of this diacritic, also changes the meaning of a word, for example, ਸਤ / sat / sət / essence and ਸੱਤ / satt / sətt / seven.

Prolative Vowel

With the help of *Addak*, long vowel is elongated. Occurrence of the vowel at the end of the word can make the vowel bigger the length of the vowel to one plus half times. For example: ਰਲਾ / rla / rla / mix and ਰਲਾੱ / rlla / rlaː / raw. Due to the increase in length of vowel, meaning of the word also changes.

Nasalization

Tippi (ੰ) and *bindi* (ਂ) diacritics are used to produce a nasal sound in the Punjabi language. This is called as nasalization, and it also changes the meaning of word, for example, ਘਟਾ / ghta / kəʈa / to subtract or decrease and ਘੰਟਾ /ghanta / kə̃ʈa / large Bell.

3.2.2 Prosodic Features

Prosodic features are also called as suprasegmental features. These prosodic features are aspects of speech which are beyond the study of phonemes and they handle the auditory quality of sound. Since Punjabi is a tonal language, it is highly prosodic in nature. Prosodic features of the Punjabi language, namely, intonation, stress and tone are discussed in following sub-sections.

Intonation

In Linguistics, the name given to the pitch fluctuation pattern is called as Intonation. It is applied to a unit which is bigger than the word, such as a clause or a sentence. It, therefore, becomes important in ASR for its naturalness. We can speak a given sentence in more than one way with a view to present and explain different situations. For example, Table 3.6 contains one Punjabi sentence with three intonations. In this sentence, if stress is given on 'ਮਾਰਿਆ', or 'ਸ਼ਾਮ' and 'ਰਾਮ', intention of sentence will change drastically.

Table 3.6: Example of intonation in Punjabi language

Punjabi Phrase	IPA Transcription	Meaning in English	Intonation
ਸ਼ਾਮ ਨੇ ਰਾਮ ਨੂੰ ਮਾਰਿਆ|	sʰɑːm_ŋeɪ_ɽɑːm_ŋʊ_mɑːɽeɪʏa	Sham hit Ram	Information
ਸ਼ਾਮ ਨੇ ਰਾਮ ਨੂੰ ਮਾਰਿਆ?	sʰɑːm_ŋeɪ_ɽɑːm_ŋʊ_mɑːɽeɪʏáː	Did Sham HIT Ram?	Question
ਸ਼ਾਮ ਨੇ ਰਾਮ ਨੂੰ ਮਾਰਿਆ!	sʰáːm_ŋeɪ_ɽáːm_ŋʊ_mɑːɽeɪʏa	SHAM hit RAM	Surprise

Stress

Punjabi language contains stress as a prominent feature. It is used to distinguish between the grammatical categories of the disyllable syllables. There is a use of stressed syllable in Punjabi accent and it combines length and pitch. There is lack of length and high pitch in unstressed syllable. Greater amount of energy is contained in emphasized syllables. The auxiliary diacritics of Punjabi language plays a major role in exhibiting the stress. Table 3.7 contains three examples of stress in Punjabi words.

Table 3.7: Examples of stress in Punjabi language

Punjabi Word	Transliteration	IPA Transcription	Meaning in English
ਨੱਕ	nakk	cnəkk	Nose
ਕੰਨ	kann	kə̃n	Ear
ਸਿੰਗ	sing	sɪŋgᵊ	Antler

Tone

The understanding of the mid-tone, low-tone and high-tone on similar-sounding minimal pairs is required for building an efficient PE. Two pairs of examples of these minimal pairs in the Punjabi language are given in table 3.8.

Table 3.8: Examples of minimal pairs in Punjabi language

Tone	Example in Punjabi	Transliteration	IPA Transcription	Meaning in English
Mid-tone	ਕੋੜਾ	korra	koɽɑː	Whip
Low-tone	ਘੋੜਾ	ghora	kòɽɑː	Horse
High-tone	ਕੋਹੜਾ	kohra	kóɽɑː	Leper
Mid-tone	ਕਰ	kar	kəɽ	Do
Low-tone	ਕ੍ਰਹ	karh	kèɽ	Dandruff
High-tone	ਘਰ	ghar	kéɽ	House

3.3 Prosody Marking

In order to fetch prosodic features of Punjabi speech, four layered transcription, namely, phonetic transcription using IPA symbols, break index marking, pitch accent marking and syllabification is performed on the collected data. WaveSurfer tool has been used as an interface to load the speech files and perform transcription tasks (katspaugh, 2012). Manual prosody markings have been done by a group of 7 persons, including the author of this thesis, from 2012 to 2014, under the project "Development of Prosodically Guided Phonetic Engine for Searching Speech Databases in Indian Languages" funded by DietY, MoCIT, Government of India, New Delhi.

3.3.1 IPA Transcription

Transcription of data collected in three modes, namely, read speech, lecture speech, and conversational speech has been performed manually using IPA chart. There are 64 IPA symbols including vowels, semi vowels, and consonants that have been used

in transcription of Punjabi speech data. This also includes diacritics; tone and word accents; and suprasegmentals were also used in transcription. Consonants include steps, velar, affricates, nasals, laterals, and fricatives. Figure 3.3 contains a sample transcription of a read speech mode on WaveSurfer interface. After selecting a segment, its transcription is noted down in the transcription pane (*.ph*) using IPA chart. Table 3.9 contains the statistics on the collected data and transcribed data for three modes of data collection.

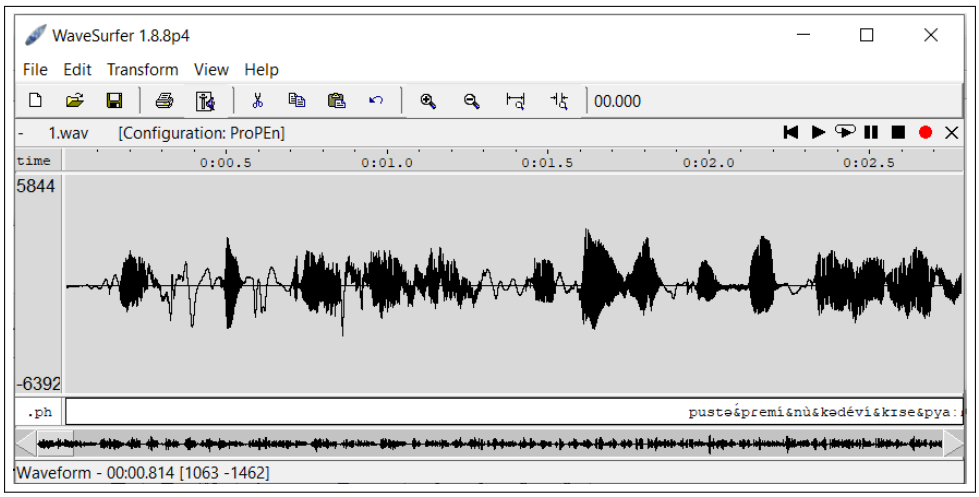


Figure 3.3: Transcription of a read speech file

Table 3.9: Amount of data transcribed using IPA

Mode	Data Collected (in hours)	Data Transcribed (in hours)
Read	15	5
Lecture	5	2.5
Conversational	4.5	2.5

3.3.2 Break Index Marking

Break index marking indicates the duration of pause between spoken words in a speech data. Boundaries of the silence region in the speech data are marked using one of the four labels, namely, B0, B1, B2, and B3. B0 is the smallest break where adjacent syllables are joined together and no physical break is present. B3 corresponds to long pauses such as sentence break. The labels B1 and B2 are used to mark the breaks where the duration is somewhere between B0 and B3. This work is done with an objective of semi-automating the break index marking and silence removal. In order to do this, each wave file of collected data in read speech mode

is divided into overlapping frames, then energy level of each frame is computed, if it is less than 1.2 dB then it is detected as silence. This work has been performed by following the similar work done by Sarma *et al.* (2014a) for speech data of Assamese language. Detected silence is then marked with labels accordingly as per the duration of silence region. Table 3.10 contains the sample of a time stamping for the break indices and Figure 3.4 contains these markings for a speech file on WaveSurfer tool's break index marking (*.bm*) pane.

Table 3.10: System generated time stamping for break index marking

Time Stamps		Break Index Marking Labels
Starting from (sec)	Ending at (sec)	
0	0.549	B3
0.954	1.422	B3
1.804	3.119	B3
3.767	3.802	B0
4.988	6.762	B3
7.338	8.667	B3
9.043	9.666	B2
9.979	9.979	B0
10.08	10.225	B1
10.7	11.973	B3

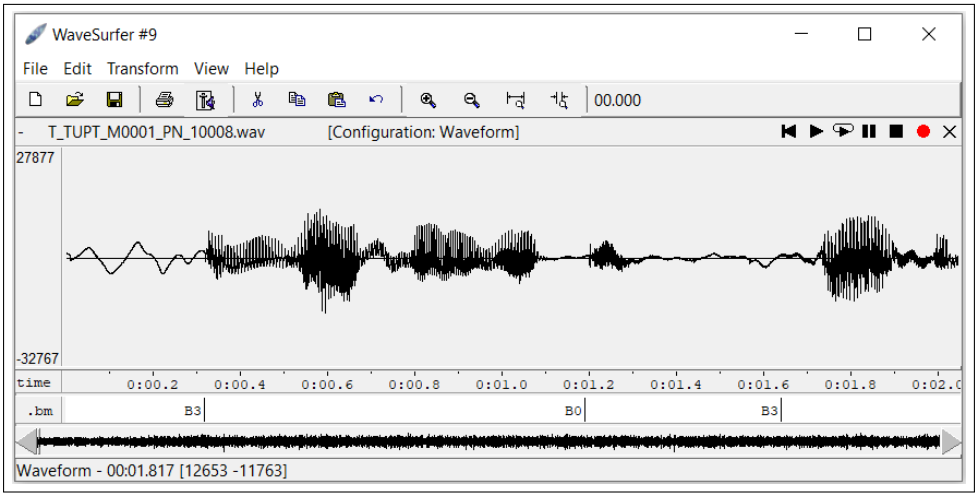


Figure 3.4: Semi-automated break index marking

3.3.3 Pitch Accent Marking

Pitch is a perceptual attribute of sound which can be described as a sensation of the relative altitude of sound. The physical correlate of pitch is the fundamental fre-

quency (F_0) determined by the rate of vibration of the vocal chords. The ensemble of pitch variations in the course of an utterance is defined as intonation. The direction of F_0 change, either rising or falling, is determined by the phonological patterns of the constituent words (Mary, 2011). The work carried out by Sarma *et al.* (2014a) is the basic building block of our work. They have done the work for Assamese language. This work has been carried out with the objective of semi-automating the process of pitch accent marking for Punjabi language. Following process has been implemented in this work.

Initially, in the whole speech, voiced and unvoiced regions are detected so that only the voiced regions are pitch marked. Zero frequency filtering technique has been used to segment the speech into voiced and unvoiced regions (Sunil Kumar and Sreenivasa Rao, 2016). Now, pitch accent marking has been done for each voiced region. In a particular voiced segment of speech, pitch accent may have 7 different marks, namely, LH (low pitch to high pitch), HL (high pitch to low pitch), F (flat pitch, *i.e.*, no change in pitch), VLH (very low pitch to high pitch), VHL (very high pitch to low pitch), LVH (low pitch to very high pitch), and HVL (high pitch to very low pitch). For pitch accent marking, sampling rate of the signal should not be more than 8000 Hz. If not so, it is re-sampled to 8000 Hz. Table 3.11 contains the sample of pitch accent marking process. It has been noted that this semi-automated process incorrectly marks the pitch boundaries in quite a few cases. As such, semi-automatic segmentation and markings have manually been corrected by deleting, inserting or shifting the segmentation boundaries and substituting the wrong markings using line fitting technique with linear regression to detect pitch variation. The results of semi-automatic and manually corrected pitch accent marking on WaveSurfer tool's pitch accent marking (*.pt*) pane are shown in Figures 3.5 and 3.6, respectively.

Table 3.11: System generated time stamping for pitch accent marking

Time Stamps		Pitch Accent Marking Labels
Starting from (sec)	Ending at (sec)	
0.192375	0.449250	LH
0.449375	0.894375	LVH
1.353500	1.605000	HL
1.684250	1.858000	F
2.043375	2.197375	F
2.785500	2.893125	LH
3.094125	3.247500	LH
3.462750	3.618750	LH
3.618875	4.573250	HL

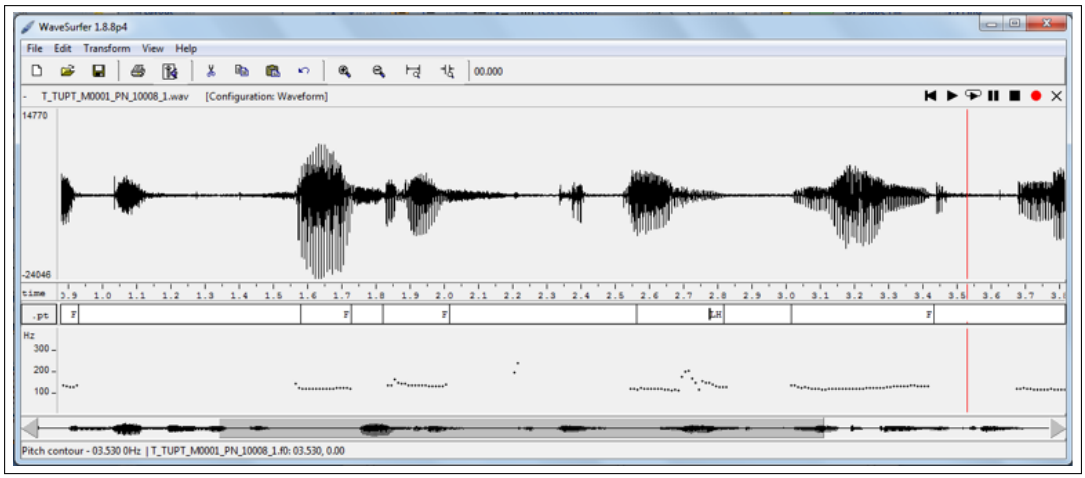


Figure 3.5: Semi-automated pitch accent marking

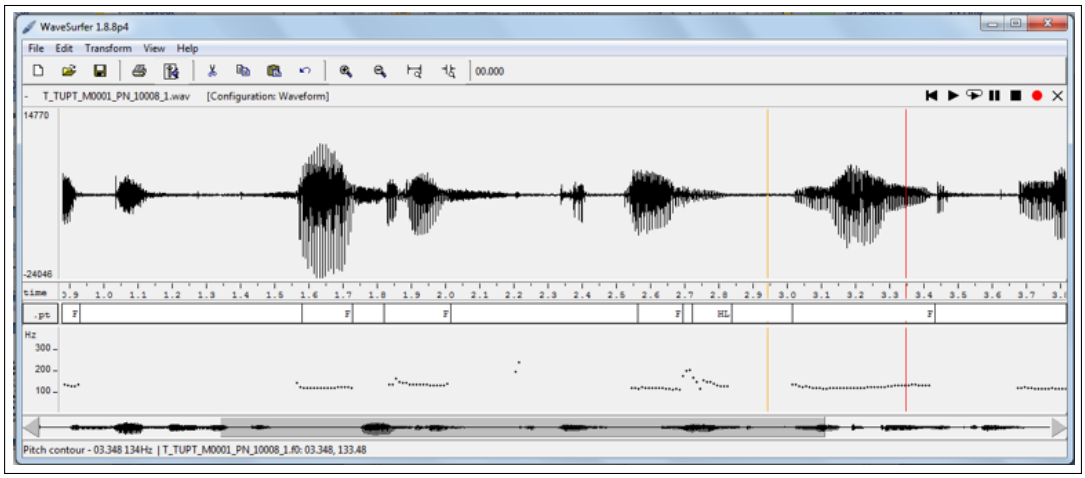


Figure 3.6: Manually corrected pitch accent marking

3.3.4 Semi-Automatic Syllabification

Syllable labelling is the process of partitioning a word into syllables with time durations. A syllable is a sub-division of a word, typically consisting of a vowel, called the nucleus and the consonant preceding and following the vowel, called the onset and the coda, respectively (Bartlett *et al.*, 2008). Most linguists consider the syllable as an important unit of prosody because many phonological rules and constraints apply within syllables or at syllable boundaries. Apart from the linguistic significance, syllables play an important role in speech synthesis and recognition (Bartlett *et al.*, 2009). One of the major reasons for considering syllable as a basic unit for ASR system is its better representation and duration stability compared to the phoneme (Nagarajan *et al.*, 2003). The syllable was proposed as a unit for ASR

by Fujimura (1975).
The syllabification is a 3-step process. The first step is phonetic segmentation and alignment of the speech data which is to be syllabified. Phonetic segmentation and alignment determines the time position of the phones of speech corpus based on manual phonetic transcription. It is done by using the HTK tool's HVite with trained phones and manual transcription. The sample of phonetic segmentation and alignment of a sentence: ਕਰਨਲ ਵਡੇਰਾ ਦੀ ਭੁੱਖੀ ਸੋਚ ਨੂੰ ਪੇਸ਼ ਕਰਦਾ ਹੈ। / karnal wadera di pukhi soch to pesh karda hai / kərnəl_ʊəɖeraːdi_púkʰi_soʧnũ_peʃkarḍaːhɛ / It presents the greedy thinking of Colonel Wadera, is shown in Table 3.12.
The second step is to perform automatic syllabification of manual phonetic transcription by applying the syllabification rules of the language under consideration as performed by Sarma *et al.* (2014b). Based on this work, we have applied the following syllabification rules for Punjabi language:

(i) Each vowel or diphthong will produce a separate nucleus.

(ii) If a single consonant is present in the left of the nucleus, it will be a onset of the right syllable.

(iii) If two consonants are present in between two vowels, the first consonant will be part of coda of previous syllable and the second will be onset of the next syllable.

(iv) If there are three or more consonants between two consecutive vowels, the first consonant will be a part of the coda of the previous syllable while the remaining consonants will be onset of the next syllable.

The third step is the syllable labeling. In this step, we extracted the time alignment for each syllable using the time alignment of their corresponding phonetic Segmentation that was marked in the first step. Table 3.13 contains the results of syllabification process for a sample data of 4.27 seconds duration. This is the same data as in Table 3.12. For this step, phone unit 'sil' has been replaced with 'x' as 'sil' was confusing the process of syllabification. The symbol 'x' has been used as it does not conflict with other phones of Punjabi language.
Finally we achieved four pane transcription including IPA transcription, syllabification, pitch accent marking and break index marking. Figure 3.7 shows the example of a speech file with all four panes in WaveSurfer.

Table 3.12: Semi-automatic phonetic segmentation and alignment

Phone Onset (in sec)	Phone Offset (in sec)	IPA (Phone Unit)
0	0.0300000	sil
0.0300000	0.0700000	k
0.0700000	0.1000000	é
0.1000000	0.2100000	r
0.2100000	0.3300000	n
0.3300000	0.3900000	é
0.3900000	0.5500000	l
0.5500000	0.7000000	ê
0.7000000	0.7500000	é
0.7500000	0.8100000	d
0.8100000	0.9200000	ée
0.9200000	1.0000000	r
1.0000000	1.0700000	aː
1.0700000	1.1900000	d
1.1900000	1.4500000	i
1.4500000	1.6500000	sil
1.6500000	1.7200000	p
1.7200000	1.8000000	u
1.8000000	1.8500000	p^h
1.8500000	2.0900000	i
2.0900000	2.2000000	s
2.2000000	2.3500000	o
2.3500000	2.4600000	χ
2.4600000	2.5400000	n
2.5400000	2.7800000	ũ
2.7800000	3.1400000	sil
3.1400000	3.1900000	p
3.1900000	3.3300000	e
3.3300000	3.5300000	ʃ
3.5300000	3.6300000	k
3.6300000	3.6700000	a
3.6700000	3.7100000	ɾ
3.7100000	3.8100000	d
3.8100000	3.8700000	aː
3.8700000	4.0600000	h
4.0600000	4.2000000	ɛ
4.2000000	4.2700000	sil

Table 3.13: Semi-automatic syllabification with time alignment

Syllable Onset (in sec)	Syllable Offset (in sec)	Syllable
0	0.0300000	x
0.0300000	0.2100000	kér
0.2100000	0.5500000	nél
0.5500000	0.7400000	êé
0.7400000	0.9900000	éeɾ
0.9900000	1.0700000	aː
1.0700000	1.4500000	di
1.4500000	1.6400000	x
1.6400000	1.7700000	pu
1.7700000	2.0800000	p^hi
2.0800000	2.4600000	soχ
2.4600000	2.7800000	nũ
2.7800000	3.1100000	x
3.1100000	3.5200000	peʃ
3.5200000	3.7100000	kaɾ
3.7100000	3.8700000	daː
3.8700000	4.1800000	hɛ
4.1800000	4.2700000	x

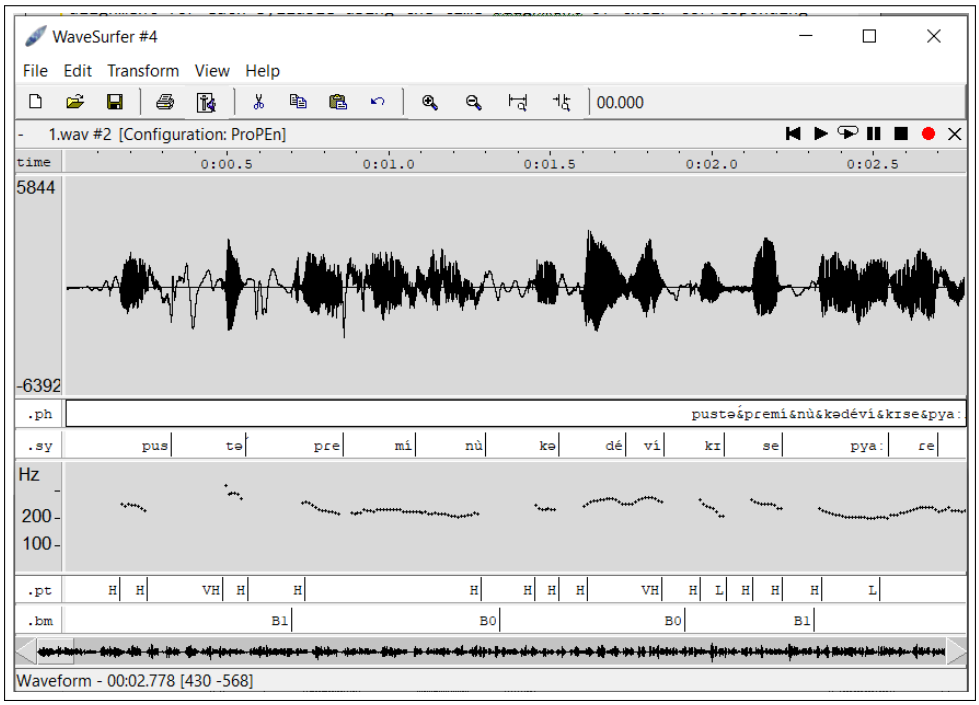


Figure 3.7: Four pane prosody marking in WaveSurfer

3.4 Summary

This chapter explains the preliminary work done for building Phonetic Engine for Punjabi language. For this purpose, three modes of speech data has been collected. Phonetic transcription is carried out using IPA chart. Possibility of incorporating prosodic knowledge into baseline system is explored. Break index marking, semi-automatic pitch accent marking and syllabification were performed to represent prosody. In order to create a finer Phonetic Engine, the prosody marking information plays an important role. The develpoment of this Phonetic Engine is explained in Chapter 4.

Chapter 4

Hidden Markov Model Based Phonetic Engine

A phonetic level speech recognition system for continuous speech of Punjabi language, also called as Phonetic Engine (PE) has been proposed in this chapter. As a first step towards the development of PE, 24.5 hours of data has been collected in three different modes, namely, read speech, lecture speech and conversational speech as explained in previous chapter. The 10 hours of total collected data has manually been transcribed using IPA chart. 5 hours of read speech data and its corresponding IPA transcription has been used for the development of PE. In this chapter, the architecture of the PE has primarily been explained. It includes three different phases: data preparation, PE training and PE testing. Initially, the vocabulary of 49 phones has been chosen by carefully analysing the symbol frequency in IPA transcription and data files were prepared to train the system accordingly. The prepared data files and speech files were then used for feature extraction and modelling. Further in this chapter, the development of PE has been explained where MFCCs have been used as a feature extraction technique and HMM as a classifier. The PE has been developed using HTK Toolkit. In the last part of this chapter, the performance of PE is presented using three different approaches: (i) By increasing the amount of data from 3 hours to 5 hours, (ii) By decreasing the number of symbols from 49 to 29, and (iii) By increasing MFCC dimensions from 12 to 36.

4.1 Mapping of Phones as per IPA Transcription

In order to train a PE, labelled data is required. In this work, IPA chart is used to label data in the form of phonetic transcription. The visual representation of speech sounds (phones) is called the phonetic transcription. IPA is the most common type of phonetic transcriptions. Phonetic transcription deals with the sound of phones used in words, *i.e.*, it tells us about the pronunciation of the words. Another reason for selection of IPA chart for transcription is that almost all spoken sounds uttered by human beings can be represented using IPA chart and in addition to message, supra-segmental features and word accents that correspond to prosody can also be represented. Three examples of phonetic transcription of Punjabi phrases are given below in Table 4.1.

Table 4.1: IPA transcription of Punjabi phrases

Punjabi Phrase	IPA Transcription
ਗੁਰੂ ਰਾਮਦਾਸ ਜੀ	gʊru_ramdas_ʤi
ਵਾਹਿਗੁਰੂ ਜੀ ਕਾ ਖਾਲਸਾ	vahɪgʊru_ʤi_ka_k^{h}alsa
ਸੱਜੇ ਹਥ ਮੁੜ ਜਾਣਾ	səʤʤe_həth_mʊɳ_ʤaɳa

In order to build PE, there is a need to split the speech data into chunks of 4 to 6 second duration. These chunks are then transcribed using IPA. In this work, WaveSurfer has been used as a front end tool for transcription. Using WaveSurfer, a human being carefully listened the speech signal and the transcription was affected. One simultaneously viewed the spectrogram of the speech signal so that the errors are minimized. After carefully analysing the transcribed data, it was found that 64 unique IPA symbols have been used for transcription of Punjabi speech data. Based on the frequency of occurrences of these symbols, 49 IPA symbols were selected and their corresponding ASCII symbols were used to train the PE. It was noticed that the PE was not that well trained when 49 IPA symbols were used. The confusing symbols were critically analysed keeping in mind that accuracy of the PE should be increased. Merging of the diacritics of vowel sounds is one approach that we followed in this work to reduce the number of symbols. For example, long 'a', short 'a', and all other diacritics of 'a' are merged into one symbol 'a'. This is worth mentioning here that the example set of different diacritics of a vowel is very less but this contributes in decreasing the efficiency of PE. We reduced the set of unique symbols from 49 to 29 using this approach. Table 4.2 shows the basis of mapping of various IPA symbols used in manual transcription to their corresponding IPA and

ASCII symbols. The variations in results due to decrease in number of symbols is discussed in Section 4.3.

Table 4.2: Mapping of IPA symbols to ASCII symbols

S. No.	Phonetic Symbols in IPA	Mapped Phonetic Symbol in IPA	Alternate Symbol in ASCII
1	a, ã, aˈ, aː, ă	a	aa
2	e, ẽ, eˈ, eː ĕ, ɛː, ɛ̆	e	ee
3	ĩ, iˈ, iː, ĭ	ĩ	i
4	õ, oˈ, oː, ŏ	õ	o
5	ũ, uˈ, uː, ŭ	ũ	u
6	b^h, b	b	b
7	d^h, d	d	d
8	f	f	f
9	g	g	g
10	h, ɦ, £	h	h
11	k, g^h	k	k
12	z	z	y
13	m	m	m
14	n, ɳ, ɲ	n	n
15	p	p	p
16	ɹ, ɾ	ɾ	r
17	s͡, ʂ	s͡	s
18	ʃ	ʃ	sh
19	ʈ	ʈ	t
20	v	V	v
21	ɔ, ɔ̃, ɔˈ, ɔː, ɔ̆, ɛ, ɛ̃, ɛˈ	ɔ	ao
22	l	l	l
23	t^h	t^h	th
24	p^h	p^h	ph
25	k^h	k^h	kh
26	ŋ	ŋ	ng
27	j	j	j
28	χ	χ	ch
29	d͡z	d͡z	dz

4.2 Architecture of Phonetic Engine

As shown in Figure 4.1, the architecture of proposed system is divided into three modules, namely, data preparation, PE training and PE testing. Each module takes specific files in specified formats as input and produces desired output based on HTK. All the files used in the development of PE are given in Appendix B. The

description of each of these modules is given in following sub-sections.

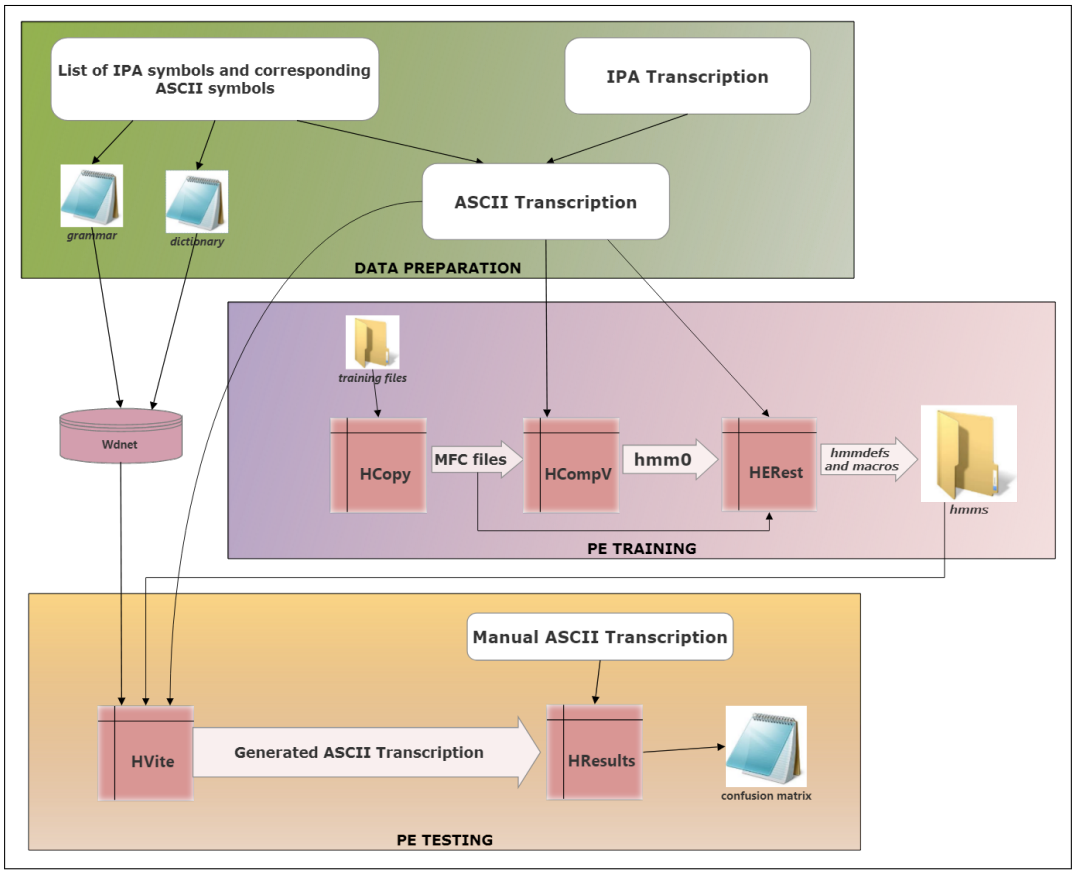


Figure 4.1: Architecture of PE

4.2.1 Data Preparation

Speech files and IPA transcription cannot directly train the PE, we need to convert the data into desired format. All speech files are converted into appropriate parametric form and their associated transcriptions are converted to ASCII format in this module. The Process of data preparation module of PE is shown in Figure 4.2.

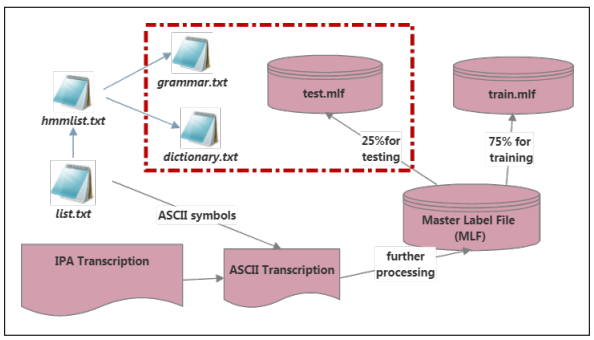


Figure 4.2: Process of data preparation module of PE

In this module, speech files, their corresponding IPA transcription and list of ASCII symbols are used to prepare the PE for training. The file *list.txt* contains 49 IPA symbols and their corresponding ASCII symbols (Appendix B.1). As shown in Figure 4.2, by using IPA transcription file and *list.txt* file, ASCII transcription is generated. Along with ASCII transcription, this file also contains names of corresponding speech files. Master Label File (MLF) has been created by removing speech file names from ASCII transcription file. MLF has further been used for training and testing of the PE and it is divided into two parts, *i.e., train.mlf* (includes 75% of data for training the PE) (Appendix B.2) and *test.mlf* (includes rest of 25% of data for testing the PE) (Appendix B.3). The file *list.txt*, containing IPA and ASCII symbols has also been used to generate *hmmlist.txt* (Appendix B.4), which is the list of ASCII symbols used in transcription of speech files. From *hmmlist.txt*, two files, namely, *grammar.txt* (Appendix B.5) and *dictionary.txt* (Appendix B.6) have been created. The *grammar.txt* file contains all the possible combinations of symbols listed in *hmmlist.txt* file and it will act as input for creation of acoustic model. The *dictionary.txt* file contains literal meanings of symbols listed in *hmmlist.txt* file.

4.2.2 PE Training

For PE training module, two processes have been executed: (i) Extraction of features from speech files and (ii) Generation of HMMs for testing the PE.

Feature Extraction

In order to train the PE, we need to extract the features from the speech files. The process of feature extraction sub-module is shown in Figure 4.3. HTK toolkit is used to extract features and to train the engine with requisite files. MFCCs are considered to give the best approximation of speech as per the human perception (Memon *et al.*, 2012; Sahoo *et al.*, 2012). Following them, MFCCs have been used for feature extraction in this work. For feature extraction, path of all the speech files are kept in *target.list* file (Appendix B.7). A configuration file has been prepared following the HTK configuration file format as presented in Young and Young (1994). It specifies SOURCEKIND (type of source file as waveform), SOURCEFORMAT (format of source file as *wav*), TARGETKIND (target parameters as MFCC_0_D_A_Z), TARGETRATE (the frame period as 10 ms), SAVECOMPRESSED (the output should be saved in compressed format), SAVEWITHCRC (a crc checksum should be added), WINDOWSIZE (the window size of 25 ms), USEHAMMING (FFT should use a Hamming window), PREEMCOEF (the signal should have first or-

der preemphasis applied using a coefficient of 0.97), NUMCHANS (the filterbank should have 28 channels), CEPLIFTER (apply a lifter to final cepstral coefficients with value 22), NUMCEPS (12 MFCC coefficients should be output), ZMEANSOURCE (bias is subtracted from waveform if any), LOFREQ (lowest band edge of mel filters 0) and HIFREQ (highest band edge of mel filters as 8000). Using this configuration file and *target.list*, PE creates the MFC files database containing all the extracted MFCC features. The information of speech files specified in *target.list* file and MFCC database has been used to create *train.list* (includes 75% of information for training the PE) (Appendix B.8) and *test.list* (includes rest of 25% of information for testing the PE) (Appendix B.9) files.

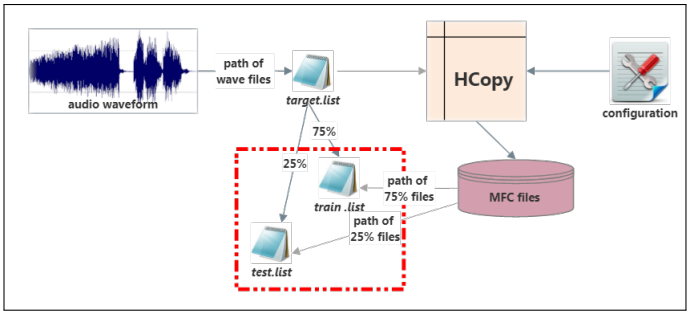


Figure 4.3: Process of feature extraction in training module of PE

Generation of HMMs

In this sub-module, speech files and their respective MFCC features are used to generate the models for selected phones. As shown in Figure 4.4, HMMs are used for modelling the phones in the proposed PE. The topology required for each HMM is written in a prototype definition file stored in the directory *hmm0*. It includes sample global mean and variance to specify the overall characteristics of HMMs. For the proposed PE, 5 states and 32 mixtures have been specified in training to generate the HMMs. Along with all these inputs, HTK takes *train.list*, that specifies information of speech files and their corresponding MFCCs. It also takes *train.mlf*, which is a master label file for training data. Given these inputs, HTK creates a Master Macro File (MMF) called *hmmdefs* containing a copy for each of the required mono-phone HMMs and their cosponsoring marco definition in the directory *hmm0*. The format of an MMF is similar to that of an MLF and it serves a similar purpose in that it avoids having a large number of individual HMM definition files. The labels, in the form of macro, stored in the directory *hmm0* are re-estimated using the embedded re-estimation tool of HTK to create HMMS for 32 mixtures.

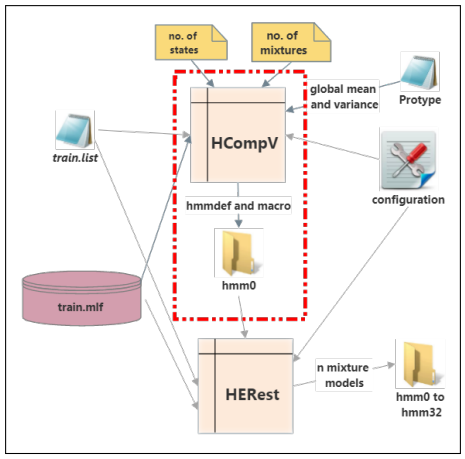


Figure 4.4: Process of generation of HMMs in training module of PE

4.2.3 PE Testing

Testing of the proposed system has been performed in two steps, as shown in Figure 4.5. In first step, PE testing is performed on *test.list* file, which contains information on the location of all the speech files designated for testing, and MFCC features of these files. In order to perform testing, PE takes files from data preparation module and HMM folders to create a text file, namely, *result.txt* that contains the PE generated ASCII transcription for speech files listed in *test.list* file.

The output of PE, *i.e., result.txt*, is then compared in step two using HTK with manual ASCII transcription, *i.e., test.mlf* to find accuracy of generated transcriptions. This creates a confusion matrix, as shown in Figure 4.6, depicting the performance parameters of the proposed PE. The performance parameters include %Corr (correctness in %), %Acc (accuracy on %), H (number of correctly detected phones), D (number of deleted phones), S (number of substituted phones), I (number of inserted phones) and N (total number of phones)

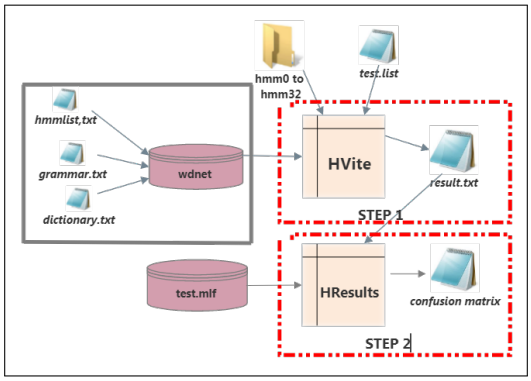


Figure 4.5: Process of testing module of PE

```
---------------------- Overall Results --------------------------
SENT: %Correct=0.00 [H=0, S=429, N=429]
WORD: %Corr=72.24, Acc=63.22 [H=10928, D=1218, S=2981, I=1364, N=15127]
---------------------- Confusion Matrix -------------------------
      a   e   i   o   u   b   d   g   h   k   y   m   n   p   r   s   s   t   v   a   l   t   p   k   n   c   d
      a   e                                                       h       o       h   h   h   g   h   z
                                                                                                          Del [ %c / %e]
  aa 1251  5   0   3   1   0   0   0   5   0   7   2   6   0   0   0   0   0   4  64   1   0   0   0   1   0   1   56 [92.6/0.7]
  ee    5 921 17   0   0   1   0   0   2   0   7   0   0   0   3   0   0   3   0  12   2   0   0   0   2   0   0   45 [94.5/0.4]
   i    0  67 1032  2   1   3   1   0   9   1   7   1  12   7   1   0   0   8   2  15   4   0   1   0   0   0   1  106 [87.8/0.9]
   o    1   0   0 322   9   0   0   0   1   0   0   0   1   1   0   0   0   1   2   1   0   0   0   0   0   0   0    3 [95.0/0.1]
   u    0   1   0  27 301   2   0   0   0   1   0   1   2   3   1   0   0   2   5   7   1   0   1   0   0   2   0   24 [84.3/0.4]
   b    0   0   0   0   0 154   8   2   1   0   0   6   0   4   0   0   0   0  11   0   0   0   0   0   0   0   1    3 [82.4/0.2]
   d    1   1   0   0   0   1 642   7   1   1   0   0   3   1   1   0   0  14   6   0   0   0   0   0   0   0   3   13 [94.1/0.3]
   f    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0    0
   g    0   0   0   1   0   3   4 108   0   2   0   0   0   0   0   0   0   1   0   1   1   0   0   0   0   0   2    8 [87.8/0.1]
   h    3   2   0   1   0   1   0   1 588   2   3   0   2  13   1   5   3  10   1   7   0   1   2   1   0   1   1   79 [90.6/0.4]
   k    0   0   0   0   0   0   4  11   3 820   0   3   0   3   0   0   0  16   0   1   0   0   1   0   0   1   1   23 [94.9/0.3]
   y    0   2   1   0   0   0   0   1   0   0  97   0   0   0   0   0   0   0   0   0   1   0   1   0   0   0   9   24 [86.6/0.1]
   m    0   0   0   0   1   0   0   1   1   0   0 211  11   3   0   0   0   0   2   1   0   0   0   0   0   0   0    1 [91.3/0.1]
   n    2   1   0   0   2   1   3   0   2   1   0   9 538   2   1   0   0   0   1   3   2   1   0   0  17   0   1   36 [91.7/0.3]
   p    0   0   0   0   0   3   0   0   0   0   1   1   0 270   0   0   0  19   0   2   0   1   0   0   0   2   0    8 [90.3/0.2]
   r    7  12   1   0   0   1   7   1   1   0   3   1   0   1 660   1   0   1   3  11  11   1   1   0  11   0   4   36 [89.3/0.5]
   s    1   2   1   0   0   0   0   0  43   0   0   0   1   3   1 369   3   6   2   0   1   2   5   0   0   1   2    8 [83.3/0.5]
  sh    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1  49   1   0   0   0   0   0   0   0   1   0    1 [94.2/0.0]
   t    1   0   0   0   0   0   4   0   0   1   0   0   0   0   0   0   0 517   0   1   0   3   0   0   0   0   0    7 [98.1/0.1]
   v    0   0   0   3   5   7   0   1   1   0   0   2   0   2   1   0   0   0 306   0   0   0   1   0   1   0   0   41 [92.7/0.2]
  ao  158  19   4  26   6   2   3   1  11   2   1   3   5   5   3   1   2  12   2 682   0   0   1   0   0   0   1  357 [71.8/1.8]
   l    0   0   2   1   0   0   0   0   2   0   0   1   0   0   7   0   0   2   1   3 426   0   0   0   1   0   0    5 [95.5/0.1]
  th    0   0   0   0   0   0   0   0  12   0   0   0   0   0   0   0   0  11   0   0   0  38   0   0   0   0   1    0 [61.3/0.2]
  ph    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0    0
  kh    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0    0
  ng    1   1   1   0   2   0   1  14   0   1   0   0   8   1   1   0   0   1   2   5   0   0   1   0 165   0   0    7 [80.5/0.3]
   j    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   3    1 [ 0.0/0.0]
  ch    0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   1   2   0   1   0   0   0   0   0 200   1    2 [97.1/0.0]
  dz    0   0   2   0   0   2   0   0   0   0   0   0   3   0   0   0   0   0   0   0   0   0   0   0   0   4 261    6 [96.0/0.1]
 sil    4   7   7   5   2  24   7   5 170  56   0  13  16 349   0   9   0 934   2  85   4  16  17   0   0   4   6  318 [ 0.0/11.5]
Ins    29  53  34  13  21  32  20  13 102  28  45  18  70 129  27  14   2 235  35 350   9  17  46   0   6   5  11
```

Figure 4.6: Confusion matrix generated by HTK

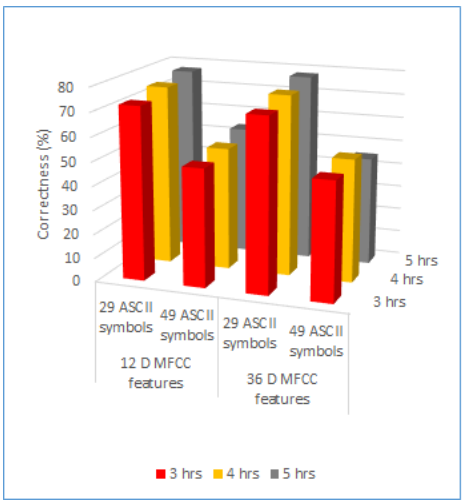
4.3 Performance Analysis

The proposed PE has been tested with three different approaches: (i) Amount of data, (ii) Number of ASCII symbols and (iii) MFCC dimensions. We have considered 3, 4 and 5 hours of data; two levels (29 and 49) of ASCII symbols and two levels (12 and 36) of MFCC dimensions. This gives us 12 cases as included in Table 4.3. The correctness and accuracy, along with all other performance parameters, of these cases have also been shown in Table 4.3. The correctness and accuracy of proposed PE are calculated using (4.1) and (4.2).

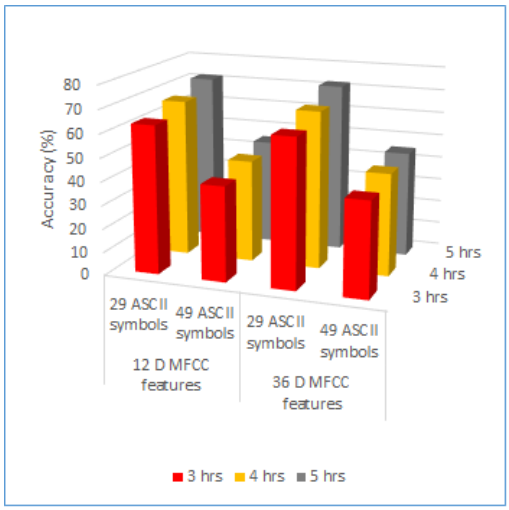
$$\text{Correctness} = \frac{\text{H(Correctly detected phones)}}{\text{N(Total number of phones)}} \times 100 \tag{4.1}$$

$$\text{Accuracy} = \frac{\text{H(Correctly detected phones)} - \text{I(Number of inserted phones)}}{\text{N(Total number of phones)}} \times 100 \tag{4.2}$$

Here, total number of phones includes sum of correctly detected phones, deleted phones and substituted phones. Figures 4.7 (a) and 4.7 (b) shows the correctness and accuracy of all 12 cases. It has been observed that correctness and accuracy is highest, *i.e.*, 77.98% and 72.32% respectively, in case of 29 symbols, 5 hours of data and 12 dimension MFCC features. Figures 4.8 (a) to 4.8 (d) and 4.9 further analyse the results obtained by the proposed system.



(a) Correctness of PE

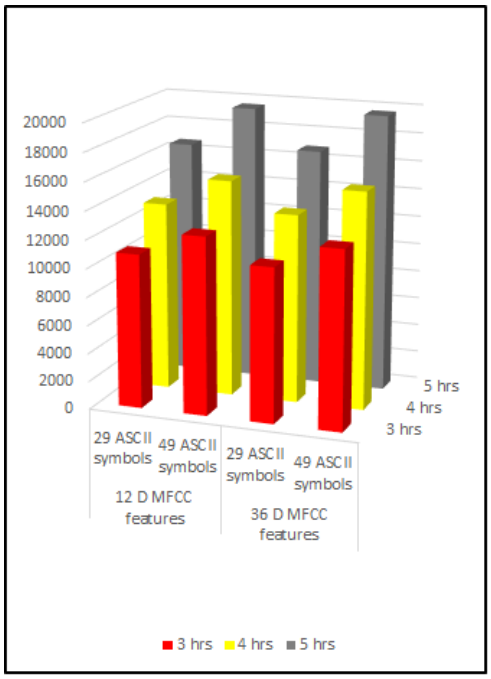


(b) Accuracy of PE

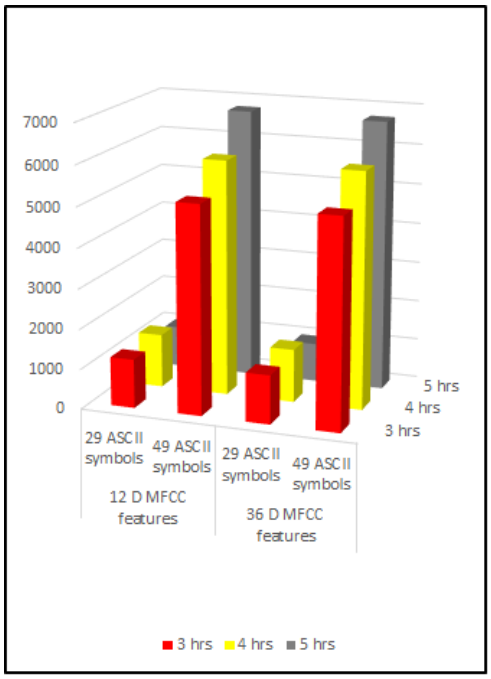
Figure 4.7: Performance of PE in 12 cases. (a) Correctness of PE, and (b) Accuracy of PE

Table 4.3: Results of PE for Punjabi language

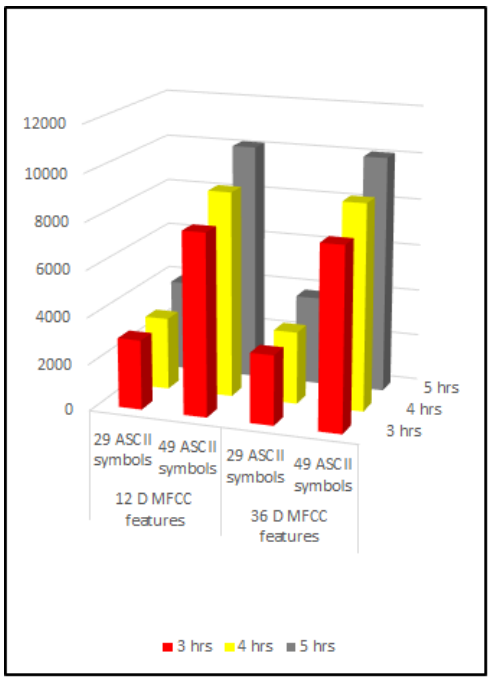
CASE No.	Data in Hours	Number of Symbols	MFCC Dimension	Number of Correctly Detected Phones	Number of Deleted Phones	Number of Substituted Phones	Number of Inserted Phones	Total Number of Phones	Correctness (in %)	Accuracy (in %)
1	3	29	12	10928	1218	2981	1364	15127	72.24	63.22
2	3	29	36	10928	1218	2981	1364	15127	72.24	63.22
3	3	49	12	12595	5186	7778	2226	25559	49.28	40.57
4	3	49	36	12595	5186	7778	2226	25559	49.28	40.57
5	4	29	12	13348	1336	3097	1389	17781	75.06	67.25
6	4	29	36	13369	1325	3087	1402	17781	75.19	67.30
7	4	49	12	15367	5870	8806	2256	30043	51.15	43.64
8	4	49	36	15373	5868	8802	2253	30043	51.17	43.67
9	5	29	12	16675	1017	3889	1167	21581	77.27	71.85
10	5	29	36	16829	963	3789	1221	21581	77.98	72.32
11	5	49	12	19563	6760	10141	3107	36464	53.65	45.13
12	5	49	36	19650	6726	10088	3172	36464	53.89	45.19



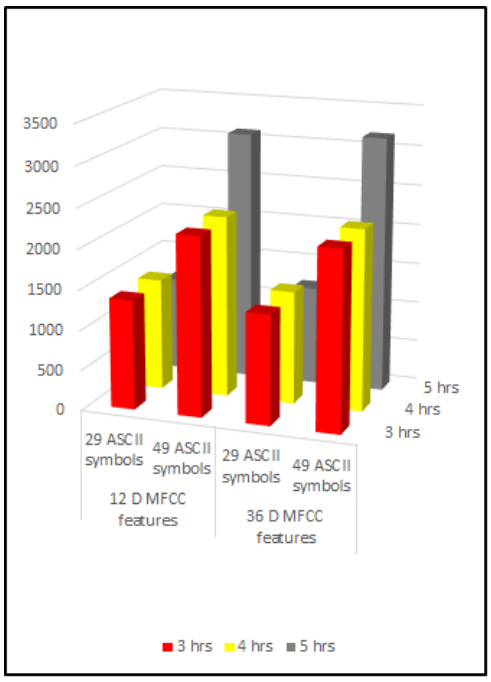
(a) Number of correctly detected phones



(b) Number of deleted phones



(c) Number of inserted phones



(d) Number of substituted phones

Figure 4.8: Performance analysis of PE for Punjabi language for 12 cases. (a) Number of correctly detected phones, (b) Number of deleted phones, (c) Number of inserted phones, and (d) Number of substituted phones

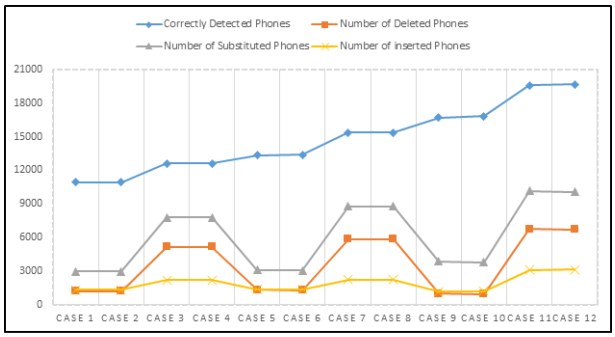


Figure 4.9: Comparison of performance of PE for Punjabi language based on number of correctly detected phones; deleted phones; inserted phones; and substituted phones in 12 cases

It has been observed from Figure 4.8 (a) and 4.8 (c) that the number of correctly detected phones and substituted phones increases as the amount of data is increased from 3 hours to 5 hours. This trend has, however, not been observed for deleted and inserted phones. Figure 4.9 further depicts the number of correctly detected phones, number of deleted phones, number of substituted phones and number of inserted phones. One can observe here that number of these phones (deleted, substituted and inserted) increases when the number of symbols is increased from 29 to 49. This was expected as increase in the number of symbols contributes in increasing the level of confusion for classifier. This has also affected the accuracy of system in these cases and the rate of increase in accuracy is not proportional to the rate of increase in data only. Thus, it can be concluded that the accuracy of the PE can be improved with increase in amount of training. We will, however, have to consider the number of symbols very carefully. There is a minor effect of changing the dimension of MFCC features on correctness and accuracy of the PE.
Also, the performance of individual phone models of the case with highest accuracy, *i.e.*, case 10, has been shown in Figure 4.10 and Table 4.4.

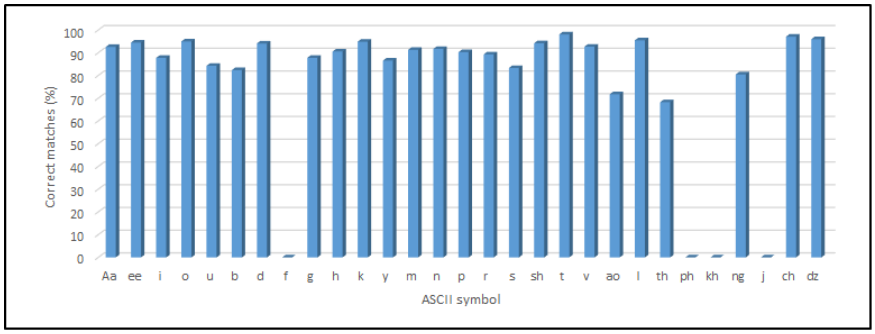


Figure 4.10: Performance of individual phone models

Table 4.4: Performance of individual phone models

S. No.	ASCII Symbol	% of Correct Matches	% of Error	Confusing Symbols
1	aa	92.6	0.7	ee, o, u, h, y, m, n, v, ao, l, ng
2	ee	94.5	0.4	aa, i, b, h, y, r, t, ao, l, ng
3	i	87.8	0.9	ee, o, u, b, d, h, k, y, m, n, p, r, t, v, ao, l, ph, dz
4	o	95.0	0.1	aa, u, h, n, p, t, v, ao
5	u	84.3	0.4	ee, o, b, k, m, n, p, r, t, v, ao, l, ph, ch
6	b	82.4	0.2	d, g, h, m, p, v, dz
7	d	94.1	0.3	aa, ee, b, g, h, k, n, p, r, t, v, dz
8	f	0.0	0.0	
9	g	87.8	0.1	o, b, d, k, t, ao, l, dz
10	h	90.6	0.4	aa, ee, o, b, g, k, y, n, p
11	k	94.9	0.3	d, g, h, m, p, t, ao, ph, dz
12	y	86.6	0.1	ee, g, l, ph, dz
13	m	91.3	0.1	u, g, n, p, v
14	n	91.7	0.3	aa, u, d, k, m, v, ao, th, ng
15	p	90.3	0.2	b, m, t, ao, th, ch
16	r	89.3	0.5	aa, ee, d, y, m, p, s, t, v, ao, l, th, ng, dz
17	s	83.3	0.5	ph, dz
18	sh	94.2	0.0	s, ch
19	t	98.1	0.1	d, th
20	v	92.7	0.2	o, u, b, g, m, p, ph
21	ao	71.8	1.8	aa, ee, I, o, u, b, d, h, m, p, r, sh, t, v, dz
22	l	95.5	0.1	I, h, r, t, v, ao
23	th	68.3	0.2	h, sh
24	ph	0.0	0.0	
25	kh	0.0	0.0	
26	ng	80.5	0.3	aa, ee, i, g, k, n, p, ao
27	j	0.0	0.0	
28	ch	97.1	0.0	
29	dz	96.0	0.1	b

4.4 Summary

In this chapter, the PE has been developed for read speech mode using HTK and has been tested. Experiments have been conducted by changing: (i)Size of data, (ii)Number of symbols and (iii)Number of MFCC dimensions. From the results, it has been concluded that the accuracy of PE increases with increase in size of data whereas it decreases when the number of symbols are increased. The performance can further be improved by increasing the size of data and considering the appropriate number of symbols.

The PE developed in this chapter can be refined to recognise the spoken words even more efficiently. Classification of the speaker in some predefined classes is also an important problem to deal with. From the existing literature, it has been observed that speaker classification techniques suffer from over-fitting and parameter tuning issues. An efficient tuning of machine learning techniques has been proposed in Chapter 5 that can improve the speaker classification accuracy.

Chapter 5

Machine Learning Based Speaker Classification

This chapter discusses the existing and proposed speaker classification techniques. In the beginning, the motivation behind this research work has been discussed, followed by the feature extraction and selection process. Some well-known competitive Machine Learning (ML) based speaker classification techniques and various performance metrics along with their mathematical formulations and range have also been discussed in this chapter. The proposed techniques have been demonstrated with the help of flowcharts and algorithms. Finally, to evaluate the effectiveness of the proposed techniques, comparative analyses with the existing techniques has been presented using the performance metrics.

5.1 Motivation

Speaker classification techniques are becoming popular day-by-day due to their usage in various real-time applications such as person authentication, recognising persons in a conversation for forensics, and security check, *etc*. Speaker classification deals with classifying the speaker's identity from his/her voice. It is a technique which can accept or reject an identity claim by comparing input speech data. Therefore, two sets of data are required: (i) Training set of speech data consisting of speech files from all the speakers to be classified as valid speakers and (ii) Testing set of speech data consisting of speech files obtained during the testing from the speaker who makes a claim. Based on the testing sample, the speaker can either be marked as an in-domain speaker or out-of-domain speaker. Speaker classification

technique is generally defined as a multi-class problem as we have to recognise who is speaking. A good number of multi-class speaker classification techniques have been designed and implemented so far to recognise the speaker efficiently. From the existing review, it has been found that the existing speaker classification techniques suffer from the over-fitting (Rafiq *et al.*, 2001; Panchal *et al.*, 2011) and parameter tuning issues (Keerthi, 2002; Friedrichs and Igel, 2005).

In this chapter, initially, we have proposed an Ensemble-based Quantum Neural Network (EQNN) to handle the over-fitting issue (Kak, 1995). Beer *et al.* (2020) and Broughton *et al.* (2021) provided the new directions and basics on Quantum Learning. However, it also suffers from parameter tuning issue Therefore, a novel tuning based Support Vector Machine (SVM) has been considered to classify the speakers efficiently. To overcome the parameter tuning issue, a Crossover based Particle Swarm Optimization (CPSO) has been designed to tune the initial parameters of SVM. We have designed CPSO because Particle Swarm Optimization (PSO) may some times get stuck in local optima, and also the performance of PSO depends upon the initially selected particles.

5.2 Feature Extraction and Selection

Data has been collected from different regions of Punjab to capture all the dialectal variations of Punjabi language. The details on the collected data have been presented in Chapter 3. Read speech data of 7 speakers, who had a good command on the Punjabi language, have been selected for this work. Their voices have been recorded at different times and with varying the distance from the mike. 30 voice samples of the one speaker have been considered for this work. Total data used for this work is of 2.5 hours duration.

The feature extraction techniques have been implemented using MATLAB to build the dataset in the form of feature vector along with target class as a given person. These techniques extracted low-level statistical features such as mean, variance, covariance, short-time energy ratio, low short time energy ratio, standard deviation of spectral centroid, spectrum flux, pitch degree, high zero-crossing rate ratio, mode, and median from the speech data (Saeed and Nammous, 2007).

Coefficient of determination (R^2) has been used to evaluate the best and non-redundant features. In order to accomplish this, two threshold values, t_1 and t_2 have been used. In this thesis, we have selected $t_1 = 0.9$ and $t_2 = 0.1$ (Rohart *et al.*, 2017; Jones *et al.*, 2019). Initially, the redundant features have been removed if the value of evaluated R^2, between inter-features is more than the threshold (t_1). Additionally, the

features which have a lower R^2 value with the target class than the threshold (t_2) has also been removed, as these features have a lesser impact on the target class. After applying the feature selection process, we have obtained potential features as mean, variance, co-variance, short-time energy ratio, low short time energy ratio, standard deviation of spectral centroid, spectrum flux, pitch degree, and high zero-crossing rate ratio, for building the speaker classification models. The sample feature vector has been given in Appendix C.1.

5.3 Classification Techniques

There are many ML techniques that can be used to achieve the speaker classification task. In this research work, some well-known competitive speaker classification techniques such as Decision Tree (DT), Random Forest (RF), SVM, and Artificial Neural Network (ANN) have been considered for experimental purpose. We have considered these classifiers for comparative analysis as these classifiers have shown significant results for speaker classification problems (Ge *et al.*, 2017; Liu *et al.*, 2018b; Kanisha *et al.*, 2018).

5.3.1 Decision Tree

The Decision tree or J48 is one of the famous tree-based non-linear machine learning techniques which has been extensively utilized in the literature to classify the speakers (Aljawarneh *et al.*, 2017). It works according to if-then rules that can be used to predict a result based on data. The root node represents a feature having maximum information gain values. Based on the current value of that feature, we can move to the next feature. All nodes in the decision tree represent features except leaf nodes, which represent the target class. Decision trees are easy to implement and are mainly designed to solve supervised problems. A problem is said to be supervised if it comes with target class *i.e.*, labelled data (Ullah *et al.*, 2019).
Although, the decision tree outperforms many machine learning techniques, but suffers from over-fitting and parameter tuning issues (Tuar *et al.*, 2017).

5.3.2 Random Forest

A random forest algorithm is also called as the ensembled decision tree. It generally divides the training data into small chunks. Thereafter, it applies the decision tree on all available chunks individually. Finally, it ensembles them to form a final result

(Ullah *et al.*, 2019). Depending on the type of input data, it either considers majority voting (in case of classification) or an average of the selected decision tree (for regression data). The performance of the random forest depends on the correlation between the decision trees. Random forest needs a minimum correlation between these trees (Belgiu and Drăguţ, 2016).
Although, random forest is able to reduce the over-fitting issue, yet, it requires an efficient tuning of the required initial parameters to build a more efficient ML model.

5.3.3 Support Vector Machine

SVM is another ML technique which has successfully been implemented in the literature to classify the speakers. It is a supervised learning model wherein training data gets assigned to the specific categories and SVM models assigns new examples based on these training models. It's working is based on the support it obtains from the input data elements (*i.e.*, vectors). It initially plots the target class and draws a hyperplane between the two target classes (Olatomiwa *et al.*, 2015). Following that, it expands the line in such a way that it touches the data elements (called as support vectors) (Belgiu and Drăguţ, 2016). Therefore, the performance of SVM depends on how efficiently it divides the given classes. Cortes and Vapnik (1995) designed SVM for efficient prediction of regression data.

5.3.4 Artificial Neural Network

ANN is a connected group of nodes called as artificial neurons. ANNs are the form of computing systems which are motivated by biological neural networks that yield the brain of animals (Xue *et al.*, 2014). They process the information in a similar way in which a human brain does, *i.e.*, they learn from examples and experience. Therefore, ANNs have the power to learn independently.They have a capability to perform better as more data becomes available. The neurons in ANNs are parallel in nature. If there is any failure in neurons, it does not affect the working of overall ANN.
It is used for prediction of the class of an input vector and can be configured to solve many problems (Siniscalchi *et al.*, 2013). ANN has many applications in the area of modeling, diagnosis, classification and pattern recognition. Some characteristics of ANN must be kept in mind before using it such as the choice of model, robustness and learning model. The capabilities of artificial network are data processing, the approximation of functions, regression analysis, controlling and classification (Phan *et al.*, 2000).

5.4 Performance Metrics

This section discusses various performance metrics used in this research work, to evaluate the performance of the existing and the proposed speaker classification techniques.

5.4.1 Accuracy Analysis

Accuracy is a well known quality metric used for evaluating the ratio of total number of correctly classified true positive and true negative classes over total number of classified classes. Confusion matrix (also called as error matrix) is used to perform the accuracy analysis as it gives us the count of exact classification classes as true positive, true negative, false positive and false negative. In this work, accuracy has been measured as correctly classified speakers over total number of classified speakers. The accuracy (A_c) is mathematically defined as given in (5.1).

$$A_c = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \times 100 \tag{5.1}$$

Here, T_p, T_n, F_p, and F_n defines true positive, true negative, false positive, and false negative values, respectively. $A_c \in [0,\ 100]$. A_c approaching towards 100 is desirable for the efficient classification. Accuracy has been considered as the primary performance metric throughout this work to make comparisons between the working of different classification techniques.

5.4.2 F-measure Analysis

If the instances of any class in the input data, either positive of negative, are more, then accuracy plays a less significant role. For such biased data, F-measure analysis is considered to be a more accurate analysis. Therefore, we have also considered the F-measure analysis which evaluates the weighted mean between Precision (p) and Recall (r). Precision, also called as 'positive predictive value', is the ratio of true positive classes among the retrieved classes, *i.e.*, sum of true positive and false positive classes. Recall, also called as 'sensitivity', is the ratio of true positive classes over the total of relevant classes, *i.e.*, sum of true positive and false negative classes Thus, F-measure considers the values of both false positives and negatives in the evaluation. Mathematically, F-measure is defined by $F1\_Score$ as given in

(5.2).

$$F1\_Score = 2 * \frac{r * p}{r + p} \tag{5.2}$$

Here, p can be computed using (5.3) and r can be estimated using (5.4).

$$p = \frac{T_p}{T_p + F_p} \tag{5.3}$$

$$r = \frac{T_p}{T_p + F_n} \tag{5.4}$$

$F1\_Score$ needs to be maximized for the optimal speaker classification.

5.4.3 Specificity Analysis

Specificity calculates the proportion of actual negative classes identified in a problem. It is also called as 'true negative rate'. It defines weather a system is able to identify non-authentic speakers in an efficient manner or not. The more the specificity, the more efficient the system will be. Specificity (S_p) can be mathematically calculated using (5.5).

$$S_p = \frac{T_n}{T_n + F_p} \tag{5.5}$$

5.4.4 Sensitivity Analysis

Sensitivity is another well-known metric to estimate the performance of the speaker classification techniques. It estimates the proportion of actual positives identified in a problem. It is also called as 'true positive rate'. Sensitivity is same as recall used in F-measure. It should be maximized for an optimal classification system. Sensitivity (S_n) can be computed using (5.6).

$$S_n = \frac{T_p}{T_p + F_n} \tag{5.6}$$

5.5 Implementation of the Base Classifiers

This section discusses the implementation and results of the existing speaker classification techniques, namely, DT, RF, SVM and ANN. To evaluate the effectiveness of these speaker classification techniques, a simulation environment has been designed using the MATLAB $2013a$. The overall objective of this section is to compare the performance of the existing techniques by considering various performance

metrics such as accuracy, F-measure, specificity and sensitivity.
A novel data splitting method has been considered in this work. We initially divided the data randomly into small chunks. Now, these chunks are considered for further partitioning. Four strategies have been employed:

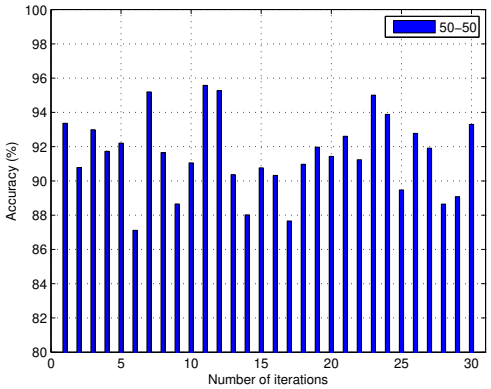
(i) Strategy *a*, wherein 50% of chunks of data are selected randomly for training and remaining 50% of chunks are used for testing,

(ii) Strategy *b*, wherein 60% chunks of data are selected randomly for training and remaining 40% chunks are used for testing,

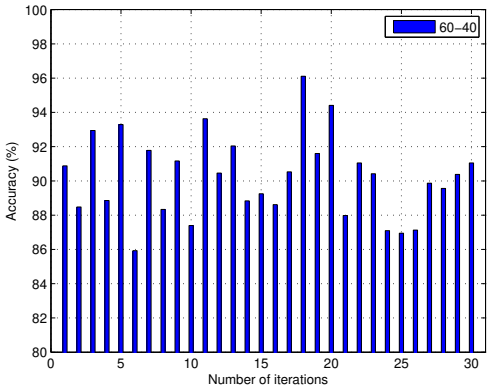
(iii) Strategy *c*, wherein 70% chunks of data are selected randomly for training and remaining 30% chunks are used for testing,

(iv) Strategy *d*, wherein 80% chunks of data are selected randomly for training and remaining 20% chunks are used for testing.

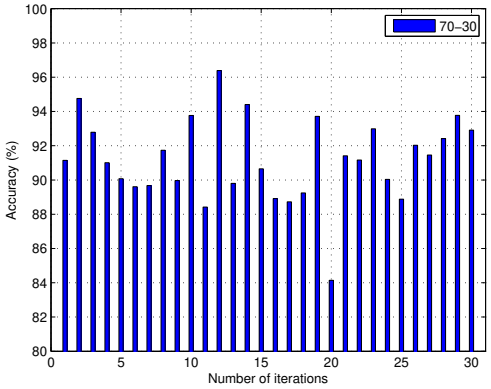
The experiments are repeated 30 times for each of the four strategies mentioned above. It is worth mentioning here that we are dealing with a 7-class problem here as the data from 7 speakers has been considered to implement the models. The repetition count of 30 has also been arrived at, by experimentation. It has been noted that the mean accuracy of models starts converging after 30 repetitions. Figures 5.1(a) to 5.1(p) show the accuracy of all 30 iterations with the 4 strategies applied on four classification techniques, namely, DT, RF, SVM and ANN.
For strategy *a*, the maximum accuracy of 99.4% and the minimum accuracy of 82.4% have been achieved with ANN and RF, respectively. For strategy *b*, the maximum accuracy of 98.9% and the minimum accuracy of 82.4% have been achieved with ANN and SVM, respectively. Furthermore, the maximum accuracy of 99.4% with ANN and the minimum accuracy of 84.1% with DT have been achieved for strategy *c*. In case of strategy *d*, the maximum and the minimum accuracy of 97.8% and 83.0% have been achieved with ANN and SVM, respectively.
For DT, the maximum accuracy of 96.4% among all the strategies has been achieved with strategy *c*, and the minimum accuracy of 84.1% has again been achieved with strategy *c*. For RF, the maximum accuracy of 95.9% and the minimum accuracy of 82.4% have been achieved with strategy *d* and strategy *a*, respectively. In case of SVM, the maximum accuracy of 96.3% with strategy *c* and the minimum accuracy of 82.4% with strategy *b* have been achieved. Lastly, the maximum accuracy of 99.4% with strategy *a* or strategy *c*, and the minimum accuracy of 88.9% with strategy *d* have been achieved in case of ANN.



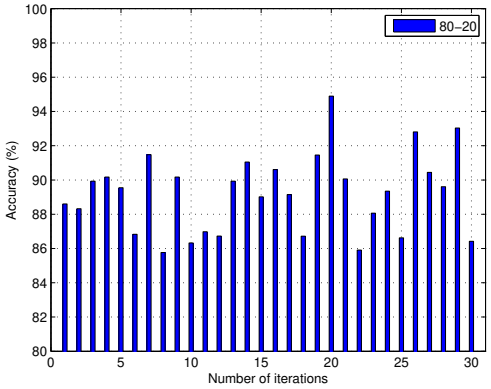
(a) Strategy *a* with DT



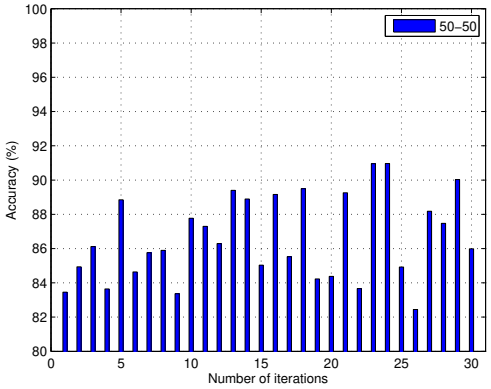
(b) Strategy *b* with DT



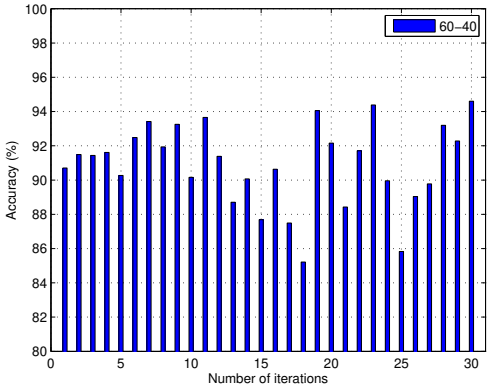
(c) Strategy *c* with DT



(d) Strategy *d* with DT

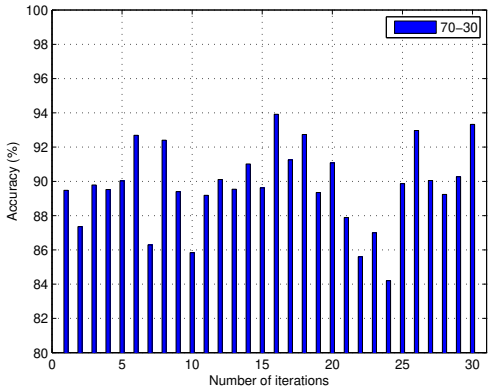


(e) Strategy *a* with RF

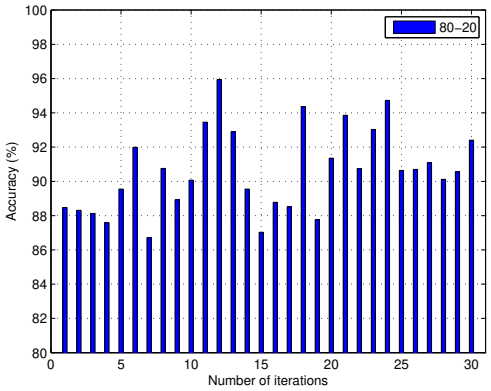


(f) Strategy *a* with RF

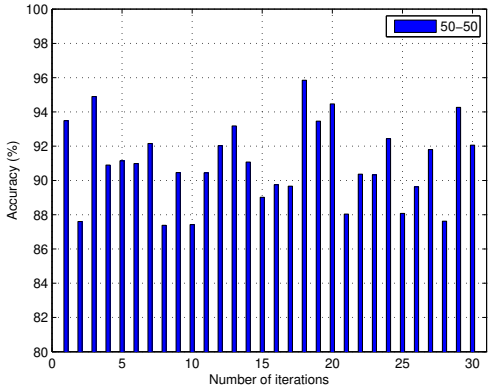
Figure 5.1: The accuracy of all 30 iterations with different data training and testing strategies using DT, RF, SVM and ANN. (a) Strategy *a* with DT, (b) Strategy *b* with DT, (c) Strategy *c* with DT, (d) Strategy *d* with DT, (e) Strategy *a* with RF, (f) Strategy *b* with RF, (g) Strategy *c* with RF, (h) Strategy *d* with RF, (i) Strategy *a* with SVM, (j) Strategy *b* with SVM, (k) Strategy *c* with SVM, (l) Strategy *d* with SVM, (m) Strategy *a* with ANN, (n) Strategy *b* with ANN, (o) Strategy *c* with ANN, and (p) Strategy *d* with ANN.



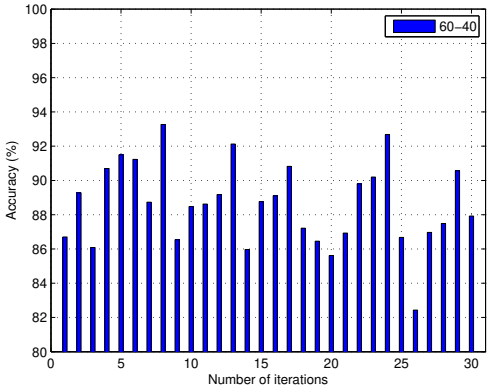
(g) Strategy *c* with RF



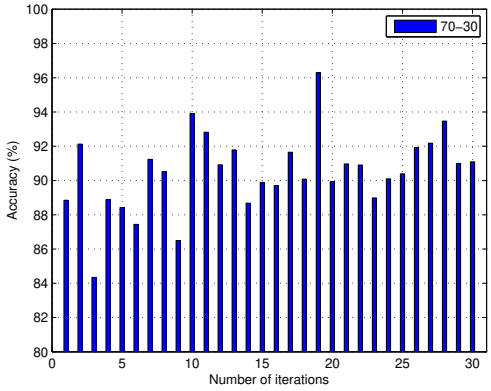
(h) Strategy *d* with RF



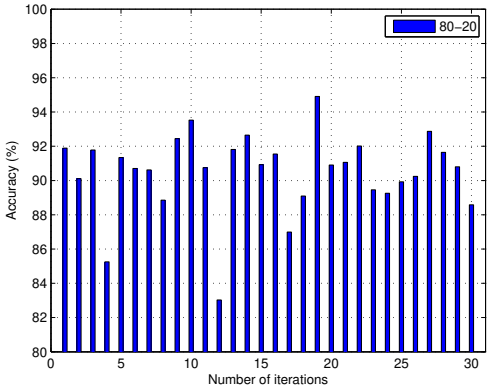
(i) Strategy *a* with SVM



(j) Strategy *b* with SVM

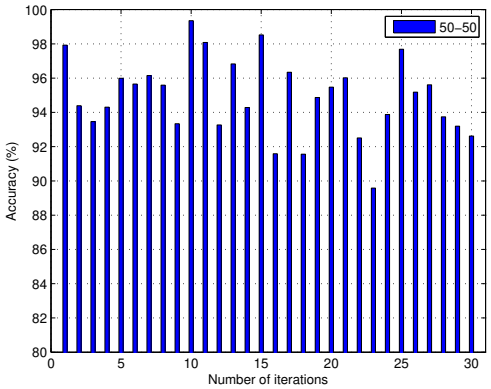


(k) Strategy *c* with SVM

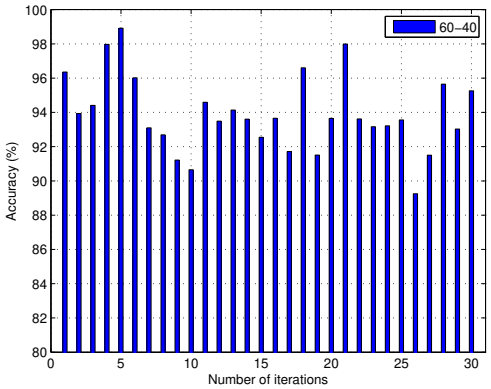


(l) Strategy *d* with SVM

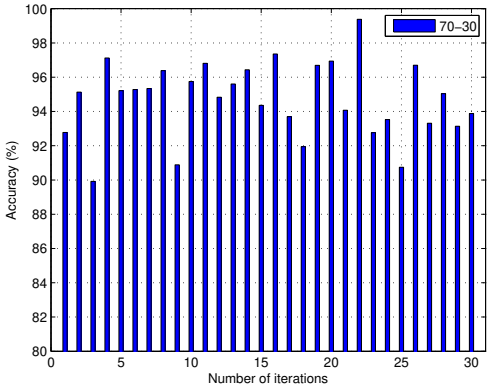
Figure 5.1: Continued.



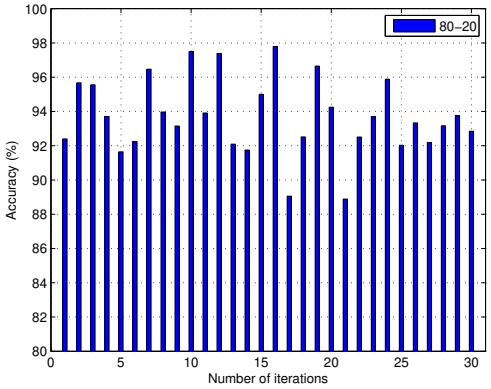
(m) Strategy *a* with ANN



(n) Strategy *b* with ANN



(o) Strategy *c* with ANN



(p) Strategy *d* with ANN

Figure 5.1: Continued.

Now, confusion matrices for all 30 iterations have been used to calculate the mean accuracy and variance for a given model and the specific strategy. Table 5.1 contains the confusion matrices for the 4 models and 4 strategies for the experiments when classification accuracy of the model was maximum. In this table, class i corresponds to i^th speaker; $i = 1, 2, ..., 7$. It is again worth mentioning here that we obtained 480 such matrices, in all, to calculate different performance parameters for the base classifiers.

It can be seen from these confusion matrices that total number of instances with a particular strategy is slightly different in each classifier. The reason behind this fact is that some of the instances have neither been recognised by the classifier as true positive nor as false negative, *i.e.*, a target class has not been assigned to such instances. We call these instances as *deleted instances*. Since the number of *deleted instances* is small, it has a negligible impact on the accuracy of the classifiers.

Table 5.1: Confusion matrices when maximum accuracy was achieved amongst 30 iterations of the models. (a) Strategy *a* with DT, (b) Strategy *b* with DT, (c) Strategy *c* with DT, (d) Strategy *d* with DT, (e) Strategy *a* with RF, (f) Strategy *b* with RF, (g) Strategy *c* with RF, (h) Strategy *d* with RF, (i) Strategy *a* with SVM, (j) Strategy *b* with SVM, (k) Strategy *c* with SVM, (l) Strategy *d* with SVM, (m) Strategy *a* with ANN, (n) Strategy *b* with ANN, (o) Strategy *c* with ANN, and (p) Strategy *d* with ANN.

Actual Class \ Predicted Class	1	2	3	4	5	6	7
1	114	0	0	0	0	0	0
2	0	99	0	0	0	0	0
3	0	0	105	0	0	0	0
4	0	0	1	114	0	0	0
5	0	0	0	0	108	1	0
6	0	0	1	0	0	99	5
7	0	0	0	0	0	4	96

(a) Strategy *a* with DT

Actual Class \ Predicted Class	1	2	3	4	5	6	7
1	98	0	0	0	0	0	0
2	0	69	0	0	0	0	0
3	0	0	84	1	0	0	0
4	0	0	1	80	0	0	0
5	0	0	0	0	87	0	0
6	0	0	0	0	0	96	6
7	0	0	0	0	0	2	74

(b) Strategy *b* with DT

Actual Class \ Predicted Class	1	2	3	4	5	6	7
1	79	0	0	0	0	0	0
2	0	70	0	0	0	0	0
3	0	0	58	0	0	0	0
4	0	0	1	59	0	0	0
5	0	0	0	0	67	1	0
6	0	0	0	0	0	59	2
7	0	0	0	0	0	2	51

(c) Strategy *c* with DT

Actual Class \ Predicted Class	1	2	3	4	5	6	7
1	55	0	0	0	0	0	0
2	0	40	0	0	0	0	0
3	0	0	47	0	0	0	0
4	0	0	0	39	0	0	0
5	0	0	0	0	30	1	0
6	0	0	0	0	0	46	1
7	0	0	0	0	0	0	40

(d) Strategy *d* with DT

Actual Class \ Predicted Class	1	2	3	4	5	6	7
1	118	0	0	0	0	0	0
2	0	112	0	0	0	0	0
3	0	0	91	0	0	0	0
4	0	0	0	107	0	0	0
5	0	0	0	0	113	0	0
6	0	0	0	0	0	106	2
7	0	0	0	0	0	3	95

(e) Strategy *a* with RF

Actual Class \ Predicted Class	1	2	3	4	5	6	7
1	94	0	0	0	0	0	0
2	0	93	0	0	0	0	0
3	0	1	84	0	0	0	0
4	0	0	0	76	0	0	0
5	0	0	0	0	86	0	0
6	0	0	0	0	0	82	2
7	0	0	0	0	0	0	80

(f) Strategy *b* with RF

Table 5.1: Continued.

Actual Class \ Predicted Class	1	2	3	4	5	6	7
1	71	0	0	0	0	0	0
2	0	64	0	0	0	0	0
3	0	0	63	0	0	0	0
4	0	0	0	65	0	0	0
5	0	0	0	0	56	0	0
6	0	0	0	0	0	62	0
7	0	0	0	0	0	2	66

(g) Strategy *c* with RF

Actual Class \ Predicted Class	1	2	3	4	5	6	7
1	45	0	0	0	0	0	0
2	0	36	0	0	0	0	0
3	0	1	37	0	0	0	0
4	0	0	0	31	0	0	0
5	0	0	0	0	57	0	0
6	0	0	0	0	0	48	0
7	0	0	0	0	0	1	43

(h) Strategy *d* with RF

Actual Class \ Predicted Class	1	2	3	4	5	6	7
1	115	0	0	0	0	0	0
2	0	100	0	0	0	0	0
3	1	0	106	0	0	0	0
4	0	0	1	109	0	0	0
5	0	0	0	0	110	3	0
6	0	0	0	0	0	106	6
7	0	0	0	0	0	10	80

(i) Strategy *a* with SVM

Actual Class \ Predicted Class	1	2	3	4	5	6	7
1	97	0	1	0	0	0	0
2	0	76	0	0	0	0	0
3	0	0	81	0	0	0	0
4	0	0	2	76	0	0	0
5	0	0	0	0	104	0	0
6	0	0	0	0	0	85	5
7	0	0	1	0	0	20	50

(j) Strategy *b* with SVM

Actual Class \ Predicted Class	1	2	3	4	5	6	7
1	81	0	1	0	0	0	0
2	0	56	0	0	0	0	0
3	0	0	64	0	0	0	0
4	0	0	0	66	0	0	0
5	0	0	0	0	58	0	0
6	0	0	0	0	0	65	2
7	0	0	0	0	0	23	33

(k) Strategy *c* with SVM

Actual Class \ Predicted Class	1	2	3	4	5	6	7
1	53	0	1	0	0	0	0
2	0	50	0	0	0	0	0
3	0	0	40	0	1	0	0
4	0	0	2	39	0	0	0
5	0	0	0	0	40	0	0
6	0	0	0	0	0	36	4
7	0	0	0	0	0	7	26

(l) Strategy *d* with SVM

Table 5.1: Continued.

Actual Class \ Predicted Class	1	2	3	4	5	6	7
1	122	0	0	0	0	0	0
2	0	103	0	0	0	0	0
3	0	0	97	0	0	0	0
4	0	0	0	108	0	0	0
5	0	0	0	0	111	0	0
6	0	0	0	0	0	104	2
7	0	0	0	0	0	0	100

(m) Strategy *a* with ANN

Actual Class \ Predicted Class	1	2	3	4	5	6	7
1	87	0	0	0	0	0	0
2	0	79	0	0	0	0	0
3	0	1	76	0	0	0	0
4	0	0	0	97	0	0	0
5	0	0	0	0	93	0	0
6	0	0	0	0	3	76	6
7	0	0	0	0	1	1	78

(n) Strategy *b* with ANN

Actual Class \ Predicted Class	1	2	3	4	5	6	7
1	70	0	0	0	0	0	0
2	0	61	0	0	0	0	0
3	0	3	56	0	0	0	0
4	0	0	8	51	3	0	0
5	0	0	0	0	69	0	0
6	0	0	0	0	4	68	3
7	0	0	0	0	0	1	52

(o) Strategy *c* with ANN

Actual Class \ Predicted Class	1	2	3	4	5	6	7
1	47	0	0	0	0	0	0
2	0	44	0	0	0	0	0
3	0	0	34	0	0	0	0
4	0	0	0	53	0	0	0
5	0	0	0	0	40	0	0
6	0	0	0	0	0	34	2
7	0	0	0	0	0	0	45

(p) Strategy *d* with ANN

Tables 5.2 to 5.5 contain the results obtained for the performance parameters. Table 5.2 shows the mean accuracy and variance of existing speaker classification techniques. It has been found that ANN achieves higher accuracy than the competitive speaker classification techniques. It has achieved a maximum accuracy of 99.4% when strategy *a* or strategy *c* is employed.

Table 5.2: Mean accuracy and variance (`dd±d.d`) of 30 iterations for 4 base classifiers and 4 strategies

Classification Technique	Strategy *a*	Strategy *b*	Strategy *c*	Strategy *d*
DT	91.3±5.1	90.2±5.6	91.2±5.7	89.2±5.1
RF	86.6±5.9	90.9±5.6	89.7±5.5	90.6±5.7
SVM	91.0±5.4	88.6±5.7	90.5±5.1	90.5±5.4
ANN	94.9±4.9	93.9±4.8	94.7±4.6	93.7±4.8

Table 5.3 depicts the F-measure of existing speaker classification techniques. It has again been found that ANN achieves larger F-measure values when compared with existing speaker classification techniques.

Table 5.3: Mean F-measure and variance (dd±d.d) of 30 iterations for 4 base classifiers and 4 strategies

Classification Technique	Strategy *a*	Strategy *b*	Strategy *c*	Strategy *d*
DT	0.72±0.12	0.70±0.10	0.72±0.12	0.65±0.13
RF	0.75±0.07	0.67±0.10	0.71±0.12	0.60±0.10
SVM	0.73±0.11	0.78±0.07	0.71±0.14	0.59±0.15
ANN	0.79±0.09	0.81±0.08	0.85±0.07	0.83±0.11

Table 5.4 demonstrates S_p analysis of the existing speaker classification techniques. It has been found that again ANN achieves higher S_p values when compared with other speaker classification techniques.

Table 5.4: Mean specificity and variance (dd±d.d) of 30 iterations for 4 base classifiers and 4 strategies

Classification Technique	Strategy *a*	Strategy *b*	Strategy *c*	Strategy *d*
DT	0.85±0.08	0.86±0.11	0.89±0.08	0.91±0.07
RF	0.87±0.07	0.88±0.08	0.90±0.07	0.92±0.07
SVM	0.88±0.06	0.87±0.09	0.91±0.06	0.93±0.05
ANN	0.87±0.08	0.89±0.08	0.92±0.06	0.94±0.04

Table 5.5 shows sensitivity analysis of the selected competitive speaker classification techniques. It has been found that ANN achieves upto 0.98 sensitivity value with strategy *a*, *b* and *c*. It shows that ANN is able to recognize speakers in more efficient manner compared to other speaker classification techniques.

Table 5.5: Mean sensitivity and variance (dd±d.d) of 30 iterations for 4 base classifiers and 4 strategies

Classification Technique	Strategy *a*	Strategy *b*	Strategy *c*	Strategy *d*
DT	0.92±0.06	0.91±0.07	0.92±0.05	0.90±0.07
RF	0.88±0.08	0.92±0.06	0.91±0.05	0.91±0.06
SVM	0.92±0.05	0.86±0.11	0.91±0.07	0.91±0.06
ANN	0.94±0.04	0.94±0.04	0.94±0.04	0.93±0.05

From Tables 5.2 to 5.5, it has been found that ANN achieves better results in the above stated performance metrics as compared to the other existing speaker classification techniques, namely, DT, RF and SVM. Based on this observation, we have selected ANN technique to remove the over-fitting issue. we have designed an ensembling method for quantum neural networks, that is discussed in the following section.

5.6 Ensemble Based Quantum Neural Network

ANNs have extensively been utilised to design speaker classification models. Although, these models achieve significant performance, yet sometimes due to extreme values in input dataset, ANNs provide unsatisfactory results. Therefore, Quantum Neural Networks (QNNs) are implemented to overcome this issue in the present work.

QNNs are special type of ANNs which belong to a class of feed-forward neural networks. The QNNs work on the principle of exploiting quantum information processing to enhance ANNs. The main step in quantum processing is to re-arrange input data vectors in such a way that outliers or extreme values do not affect the performance of ANNs, to a large extent (McClean *et al.*, 2018).

From real-time implementation of machine learning techniques, it has been found that some stronger machine learning techniques outperform weaker ones and get assigned very high coefficients (assuming we merge standard machine learning techniques with a linear quantum-machine learning technique) (Siniscalchi *et al.*, 2013). In turn, weaker machine learning techniques may obtain near-zero coefficients, since those are learned by taking into account the same training set where the standard machine learning techniques were learned (Wang *et al.*, 2017). It may lead to over-fitting whenever the given machine learning technique does not take place beforehand. However, even a machine learning technique which does not perform well for maximum part of input data may provide good results for a small set of data (*i.e.*, for some particle features in the available data) (Trentin and Gori, 2003). As discussed in earlier sections, the majority of machine learning based speaker classification techniques suffer from over-fitting issues. In this chapter, we have designed an EQNN to overcome the issue of over-fitting. Following steps have been followed to design and implement this EQNN.

(i) The input feature space has been divided as per the strategies *a* to *d* described in previous section.

(ii) Now, QNNs have been implemented on the feature space of strategy *a*.

(iii) Within QNN, cross validation has been performed where 80% of training data has been used for the training of QNN, 10% has been used for validation of training phase and remaining 10% for testing of training phase.

(iv) Steps (ii) and (iii) have been repeated 30 times as we did for base classifiers.

(v) Confusion matrices and other performance metrics have been evaluated on the testing data as per strategy *a* for all the iteration.

(vi) Output of the iteration with best training accuracy has been compared with the actual training data and an error matrix has been generated.

(vii) Actual training classes have been attached with this error matrix.

(viii) A new feature space has been prepared after ensembling the error matrix that we got in step (vii) with the feature space as per strategy *b*.

(ix) Steps (ii) to (vii) are now repeated for strategy *b*.

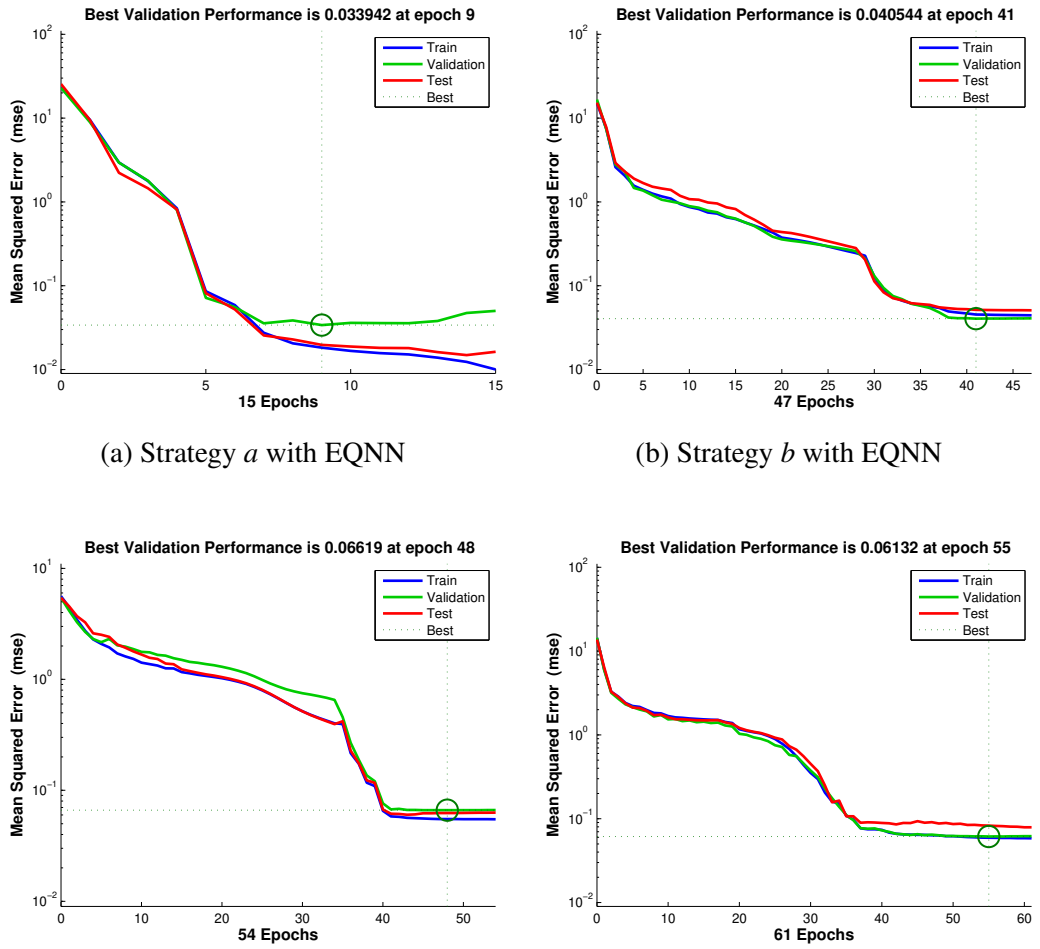
(x) Steps (ii) to (viii) are repeated for strategies *c* and *d*.

(xi) Finally, the performance of EQNN has been calculated as the mean of all the accuracies we got at step (v) for all the four strategies.

5.6.1 Performance Analysis of EQNN

The proposed EQNN has been designed and tested in the same simulation environment as implemented for existing speaker classification techniques. The number of hidden layers are kept random and the target data has been divided using random indices. *Trainlm* network function has been used to update the weight and bias values as per Levenberg-Marquardt optimization algorithm. For networks that contain less number of weights, the Levenberg-Marquardt algorithm have the fastest convergence. Also, *trainlm* is able to obtain lower mean square errors than any of the other training algorithms for neural networks. However, as the number of weights in the network increases, the advantage of *trainlm* decreases (Parmar and Hindoliya, 2011). The default values of training parameters of *Trainlm* have been used and Mean Squared Error (MSE) has been used to evaluate the network's performance. Figures 5.2(a) to 5.2(d) show the validation performance analysis of EQNN with respect to MSE. It has been noted that the best validation performance for strategy

a is 0.03394 at epoch 9 with total number of epochs being 15. For strategy *b*, best validation performance of 0.04054 has been achieved at epoch 41 with total number of epochs being 47. Similarly, for strategies *c* and *d*, best validation performance of 0.06619 and 0.06132 have been found at epochs 48 and 55 with total number of epochs being 54 and 61, respectively. It can be concluded from Figures 5.2(a) and 5.2(d), using validation and test curve, that slight over-fitting of the data has taken place.



(a) Strategy *a* with EQNN

(b) Strategy *b* with EQNN

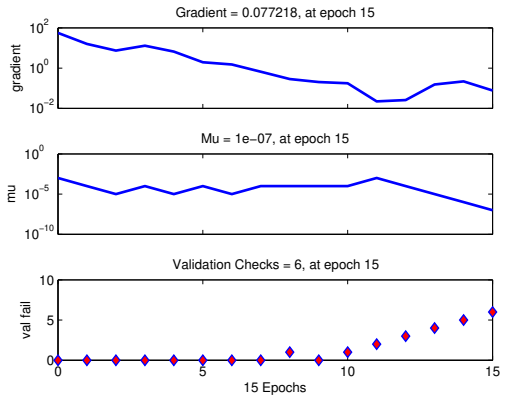
(c) Strategy *c* with EQNN

(d) Strategy *d* with EQNN

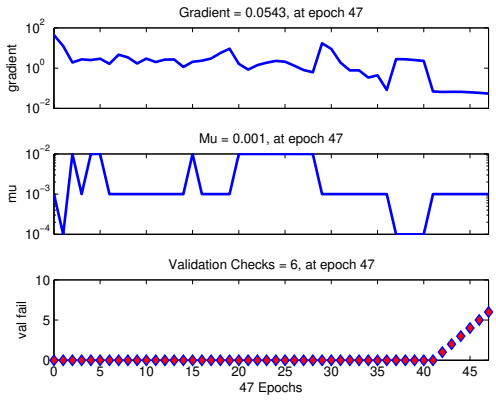
Figure 5.2: The validation performance analysis of EQNN using MSE. (a) Strategy *a* with EQNN, (b) Strategy *b* with EQNN, (c) Strategy *c* with EQNN, and (d) Strategy *d* with EQNN.

Figures 5.3(a) to 5.3(d) indicate how the `gradient` and `mu` (weight changes of neural network) change and also the number of validation checks (`val fail`) during the EQNN training process. Here `gradient` is the value of back-propagation gradient on each iteration, *i.e.*, the local minimum of the performance function, `mu`

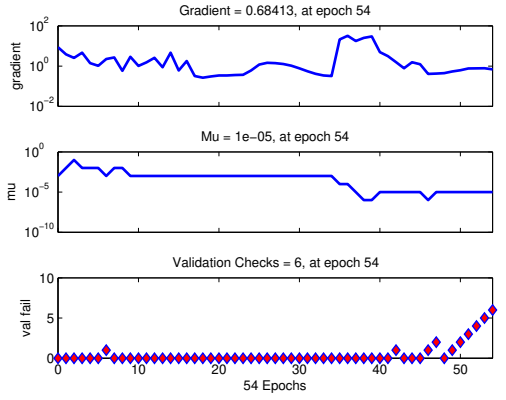
is the control parameter and `val fail` are iterations when validation MSE increased. A lot of fails corresponds to over-fitting. The training algorithm *trainlm* automatically stops training after 6 fails in a row. In the case of EQNN, `val fail` reached 6 for these strategies at epochs 15, 47, 54 and 61, respectively, which depicts that maximum level of training has been reached after these many epochs. The values of `gradient` and `mu` have been found to be 0.07722 and $1e-07$ for strategy *a*; 0.05430 and $1e-03$ for strategy *b*; 0.68413 and $1e-05$ for strategy *c*; and 0.20756 and $1e-05$ for strategy *d*.



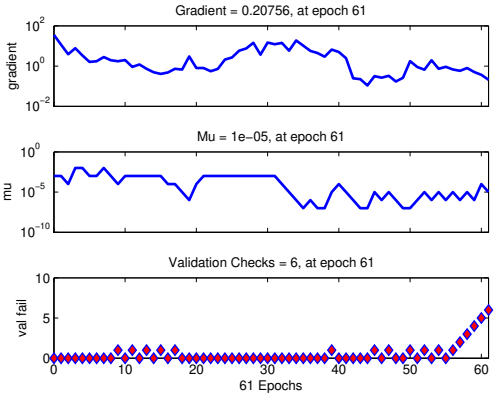
(a) Strategy *a* with EQNN



(b) Strategy *b* with EQNN



(c) Strategy *c* with EQNN

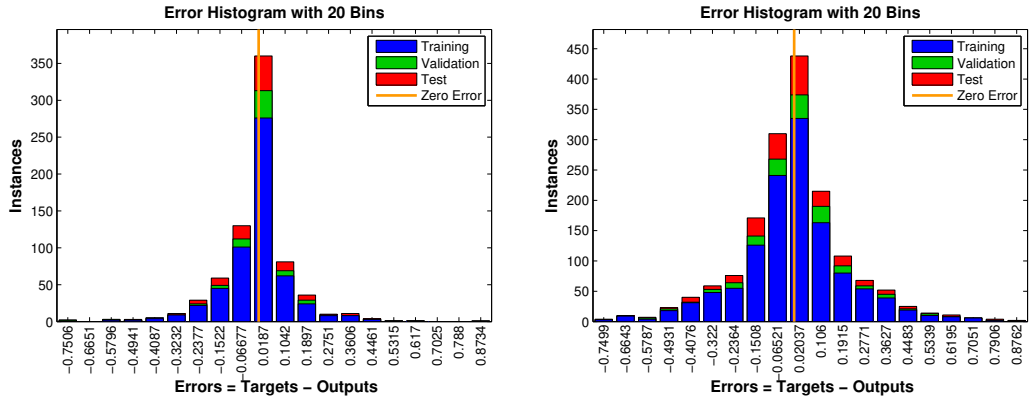


(d) Strategy *d* with EQNN

Figure 5.3: The training state performance of EQNN with respect to `gradient`, `mu` and `val fail`. (a) Strategy *a* with EQNN, (b) Strategy *b* with EQNN, (c) Strategy *c* with EQNN, and (d) Strategy *d* with EQNN.

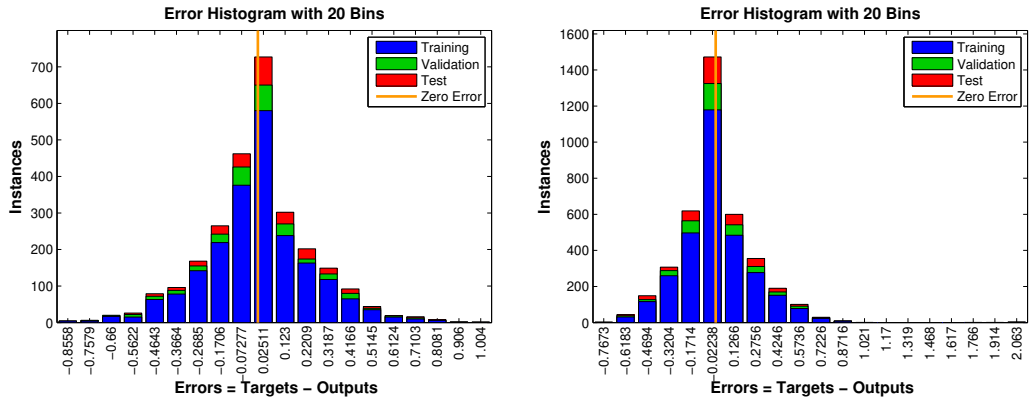
Figures 5.4(a) to 5.4(d) show the obtained error histograms with 20 bins for EQNN when the difference between the actual and the predicted classes has been calculated during the training phase of the network. The blue, green and red bars represent the

training, validation and testing data, respectively. These histograms give an indication of outliers, which are data points where the fit is significantly worse than the majority of data. In the case of strategy *a*, most errors lie between -0.2 and 0.2, and minimum error of -0.01870 has been found at 10^{th} bin. For strategy *b*, most of the errors lie between -0.5 and 0.5, and minimum error of -0.02037 has again been found at 10^{th} bin. In case of strategy *c*, most of the errors fall between -0.6 and 0.5, and minimum error of 0.02511 has been reached at 10^{th} bin. For strategy *d*, most of the errors lie between -0.5 and 0.6, and minimum error of 0.02238 has been found to be at 6^{th} bin.



(a) Strategy *a* with EQNN

(b) Strategy *b* with EQNN



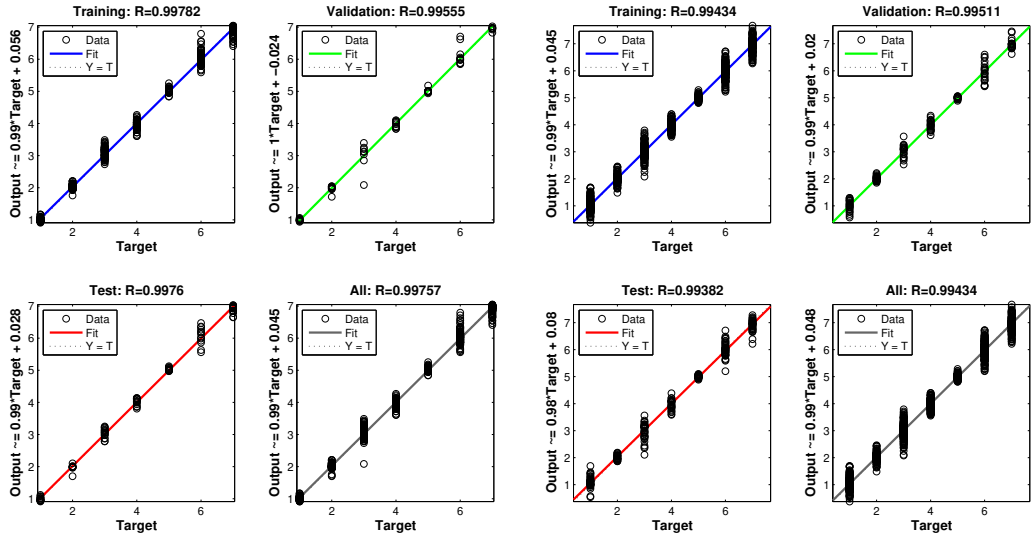
(c) Strategy *c* with EQNN

(d) Strategy *d* with EQNN

Figure 5.4: Error histogram of the EQNN performance with 20 bins. (a) Strategy *a* with EQNN, (b) Strategy *b* with EQNN, (c) Strategy *c* with EQNN, and (d) Strategy *d* with EQNN.

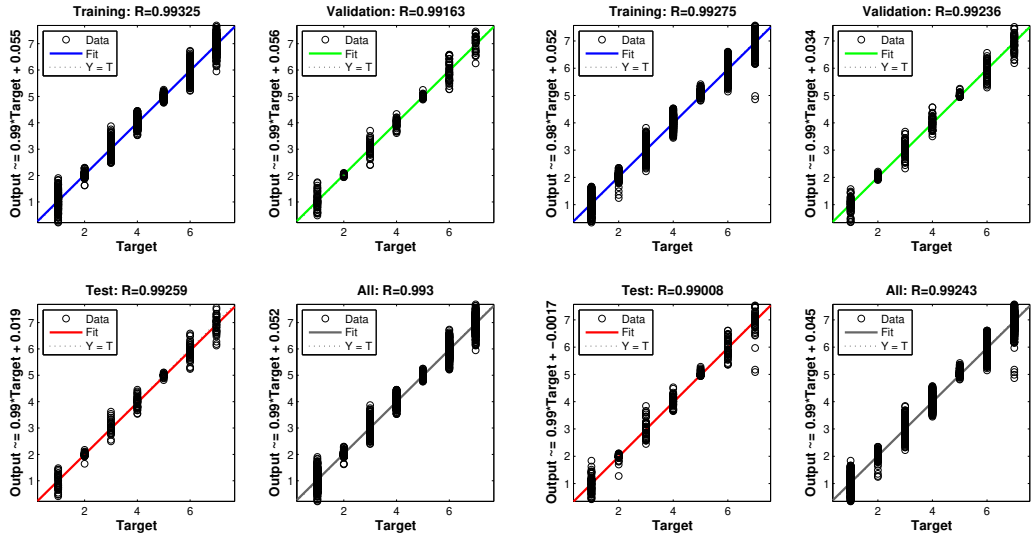
Figures 5.5(a) to 5.5(d) demonstrate the fitness of the EQNN in the form of regression plots which display network outputs with respect to training, testing, validation

and entire dataset, respectively, for the four strategies. For a perfect fit, the data should fall along a 45° line, where the network outputs are equal to the targets.



(a) Strategy *a* with EQNN

(b) Strategy *b* with EQNN



(c) Strategy *c* with EQNN

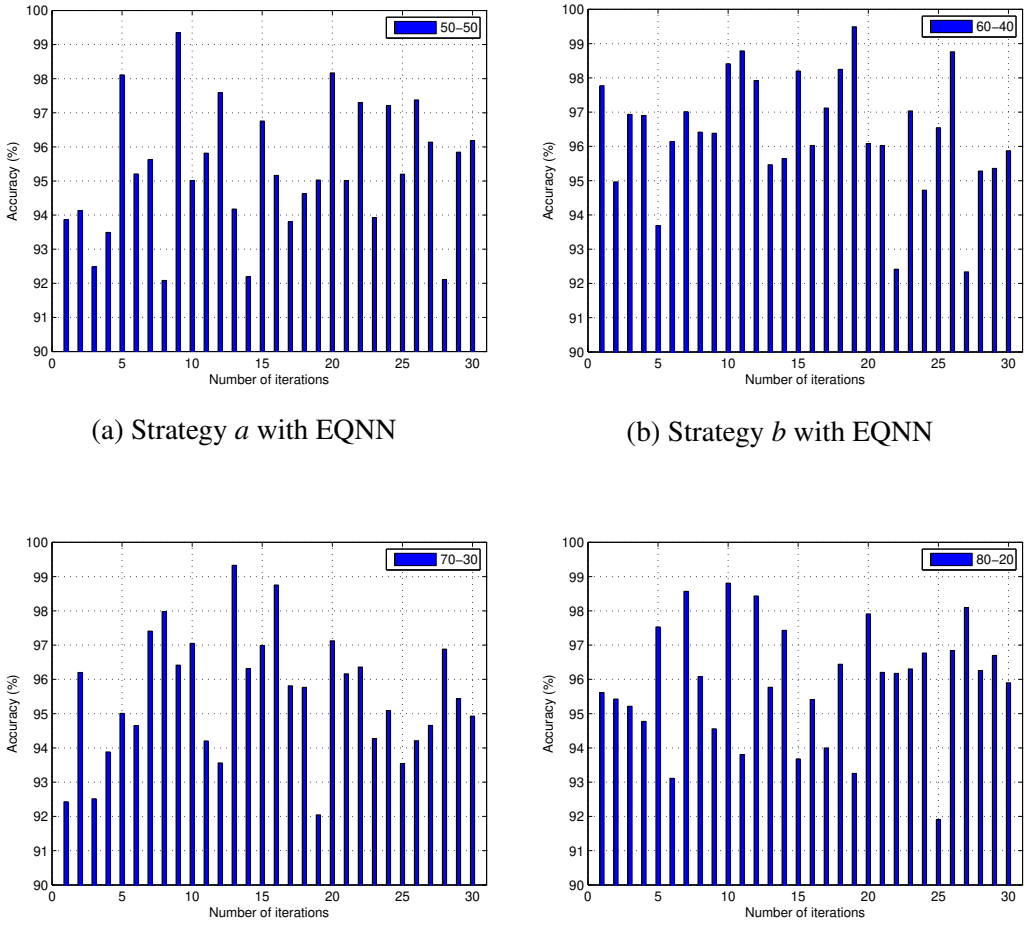
(d) Strategy *d* with EQNN

Figure 5.5: The regression plot used to validate the EQNN performance. (a) Strategy *a* with EQNN, (b) Strategy *b* with EQNN, (c) Strategy *c* with EQNN, and (d) Strategy *d* with EQNN.

It can be concluded from these figures that the EQNN is able to obtain significant prediction results as the data is closely related to the fitness line with R values in

each case for all the strategies approaching 1. It is worth mentioning here that these results have been achieved using to 30 iterations for each strategy. The more we re-train the network with same dataset, the more accurate results it gives after each retraining. This changes the initial weights and biases of the network, and produces an improved network after re-training.

Figures 5.6(a) to 5.6(d) show the accuracy of all 30 iterations with the 4 strategies applied on EQNN. It also includes the minimum and maximum accuracy in all the strategies for EQNN. Also, the mean accuracy and variance of this technique for all 30 iterations and all the strategies have been calculated by using confusion matrices of specific strategy.



(a) Strategy *a* with EQNN

(b) Strategy *b* with EQNN

(c) Strategy *c* with EQNN

(d) Strategy *d* with EQNN

Figure 5.6: The accuracy of all 30 iterations with different data training and testing strategies with EQNN. (a) Strategy *a* with EQNN, (b) Strategy *b* with EQNN, (c) Strategy *c* with EQNN, and (d) Strategy *d* with EQNN.

Table 5.6 contains the confusion matrices while employing EQNN for the 4 strategies when classification accuracy of the model has been maximum. We obtained

120 such matrices, in all, to calculate different performance parameters.

Table 5.6: Confusion matrices for maximum accuracy among 30 iterations of the EQNN. (a) Strategy *a* with EQNN, (b) Strategy *b* with EQNN, (c) Strategy *c* with EQNN, and (d) Strategy *d* with EQNN.

Actual Classes \ Predicted Classes	1	2	3	4	5	6	7
1	123	5	0	0	0	0	0
2	1	95	5	0	0	0	0
3	0	8	84	10	0	0	0
4	0	0	9	74	5	0	0
5	0	0	0	12	84	15	0
6	0	0	0	0	8	111	3
7	0	0	0	0	0	7	88

(a) Strategy *a* with EQNN

Actual Classes \ Predicted Classes	1	2	3	4	5	6	7
1	85	7	0	0	0	0	0
2	1	81	0	0	0	0	0
3	0	0	86	3	0	0	0
4	0	0	1	76	2	0	0
5	0	0	0	0	82	0	0
6	0	0	0	0	10	68	1
7	0	0	0	0	0	8	78

(b) Strategy *b* with EQNN

Actual Classes \ Predicted Classes	1	2	3	4	5	6	7
1	60	3	0	0	0	0	0
2	0	56	0	0	0	0	0
3	0	1	60	3	0	0	0
4	0	0	0	73	0	0	0
5	0	0	0	0	74	0	0
6	0	0	0	0	7	49	1
7	0	0	0	0	0	12	49

(c) Strategy *c* with EQNN

Actual Classes \ Predicted Classes	1	2	3	4	5	6	7
1	32	5	0	0	0	0	0
2	1	37	0	0	0	0	0
3	0	1	53	3	0	0	0
4	0	0	0	40	0	0	0
5	0	0	0	0	42	0	0
6	0	0	0	0	4	41	1
7	0	0	0	0	0	8	30

(d) Strategy *d* with EQNN

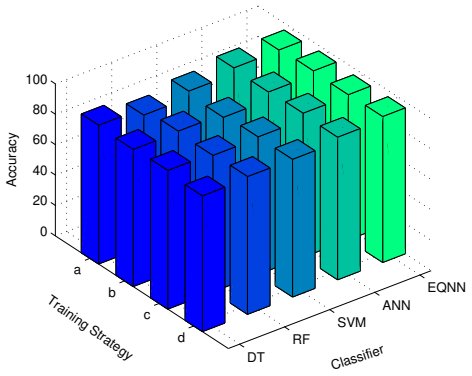
Table 5.7 shows the performance of EQNN on various performance metrics.

Table 5.7: Mean and variance (`dd±d.d`) of the performance metrics with EQNN for 30 iterations and 4 strategies

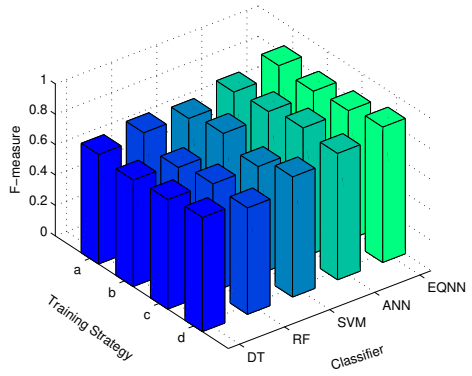
Performance Metric	Strategy *a*	Strategy *b*	Strategy *c*	Strategy *d*
Accuracy	95.3±3.5	96.4±2.9	95.5±3.2	95.9±2.9
F-measure	0.85±0.09	0.83±0.08	0.85±0.09	0.89±0.07
Specificity	0.92±0.05	0.94±0.04	0.95±0.04	0.96±0.03
Sensitivity	0.95±0.03	0.96±0.03	0.97±0.02	0.97±0.02

Figures 5.7(a) to 5.7(d) show the comparison between Accuracy, F-measure, Speci-

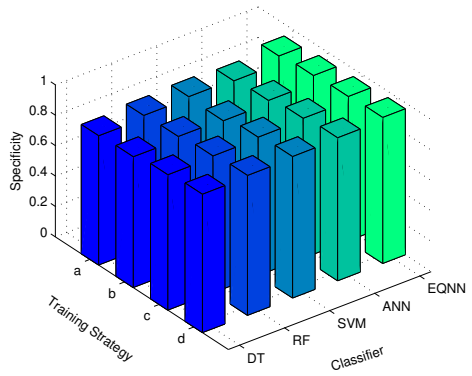
ficity and Sensitivity of all the base classifiers and EQNN for the 4 data splitting strategies. It can be inferred from these figures that the proposed EQNN achieves better results as compared to the base classifiers in all the performance metrics. Mean values of all the performance metrics among 30 iterations of 4 strategies have been plotted in these figures. The best performance has been noted in strategy *c* for the classifiers. From the overall performance of EQNN, it has been observed



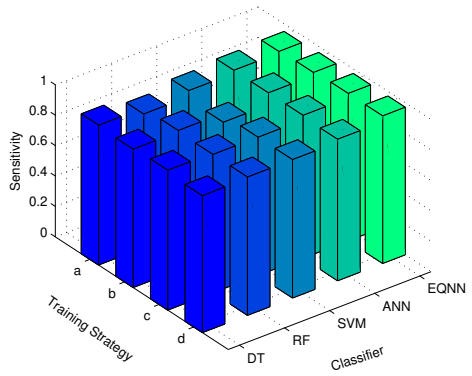
(a) Analysis of accuracy



(b) Analysis of F-measure



(c) Analysis of specificity



(d) Analysis of sensitivity

Figure 5.7: Performance analysis of EQNN over base classifiers with respect to all the performance metrics. (a) Analysis of accuracy, (b) Analysis of F-measure, (c) Analysis of specificity, and (d) Analysis of sensitivity.

that although it performs better in all the aspects than the existing classifiers but it still suffers from slight over-fitting of data as we saw this in the best validation performance of EQNN. As such, parameter tuning is also required to achieve zero validation fails in training of the network. To overcome these issues, we have further designed a new approach using SVM, which is explained in the next section.

5.7 Support Vector Machine Parameters Tuning Using Crossover Based Particle Swarm Optimization

To overcome the issues of parameter tuning and over-fitting, a novel crossover based Particle Swarm Optimization (PSO) is proposed to tune the parameters of support vector machine. PSO is a well-known metaheuristic technique. In PSO, the initial population contains a number of particles, each of which describes the initial parameters of a given solution. These particles move in the problem domain to find the best combination of parameters for a given problem. Each PSO solution updates itself by updating its velocity (V_{id}) and position (P_{id}) by using (5.7) and (5.8), respectively. V_{id} and P_{id} are updated according to the evaluated personal best (Pb_{id}) and global best (Gb_d) values.

$$V_{id}(t+1) = w \times V_{id}(t) + c_1 \times r_1 \times (Pb_{id} - P_{id}(t)) + c_2 \times r_2 \times (Gb_d - P_{id}(t)) \tag{5.7}$$

$$P_{id}(t+1) = P_{id}(t) + V_{id}(t+1) \tag{5.8}$$

Here, i represents particle in d^{th} dimension and r_1, r_2 are the random numbers between 0 and 1. Inertia weight (w), c_1 and c_2 are control parameters. In PSO, c_1 and c_2 are also called as acceleration coefficients.

In the proposed technique, the crossover operator has been used, this operator has an ability to overcome the issue of getting stuck in local optima. The crossover has been applied on the global best and personal best operators to obtain two more solutions which are called as child solutions. After that, fitness of both the children have been evaluated. If one of the obtained child solutions has good fitness than global best then the global best solution is replaced with the respective child solution. Further in this section, Crossover based PSO with SVM (CPSOSVM) has been presented in detail. Figure 5.8 shows the flowchart for the process of optimization of SVM parameters by using PSO. The overall objective is to tune the required initial parameters of SVM. Additionally, the details of the working of PSO based SVM has been shown in Figure 5.9. We now briefly discuss three algorithms that have been used to initialize the population, to calculate the objective function, and for updation of particles.

Algorithm 5.1 outlines the systematic flow of CPSOSVM. In this algorithm, three main steps have been used to generate the initial population through particle en-

coding approach of dividing the dataset into small chunks, update the velocity and position, and evaluate the termination condition, which in our case is 30 number of iterations. Initially, the population size is determined and then, individuals are generated. The number of generated individuals is the population size.

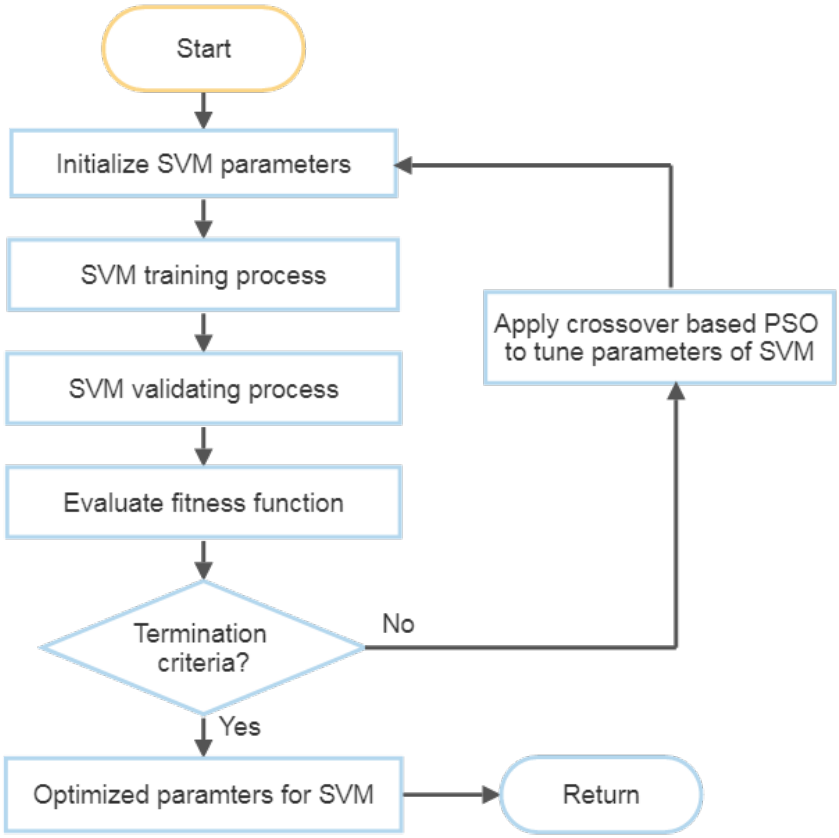


Figure 5.8: Flow of the proposed technique

Algorithm 5.1 Population initialization

$P \leftarrow$ Using proposed particle encoding approach initialize the population;
$Pb_{id} \leftarrow \phi$;
$Gb_d \leftarrow \phi$;
while termination condition not satisfied **do**
  update position and velocity of every particle by using Algorithm 5.3;
  calculate the fitness of every particle;
  update Pb_{id} and Gb_d;
  Apply crossover operator on Pb_{id} and Gb_d and store it into Crossover based global best (Cg_d);
  Finally, update Gb_d by comparing it with Cg_d to obtain best solution.
end while

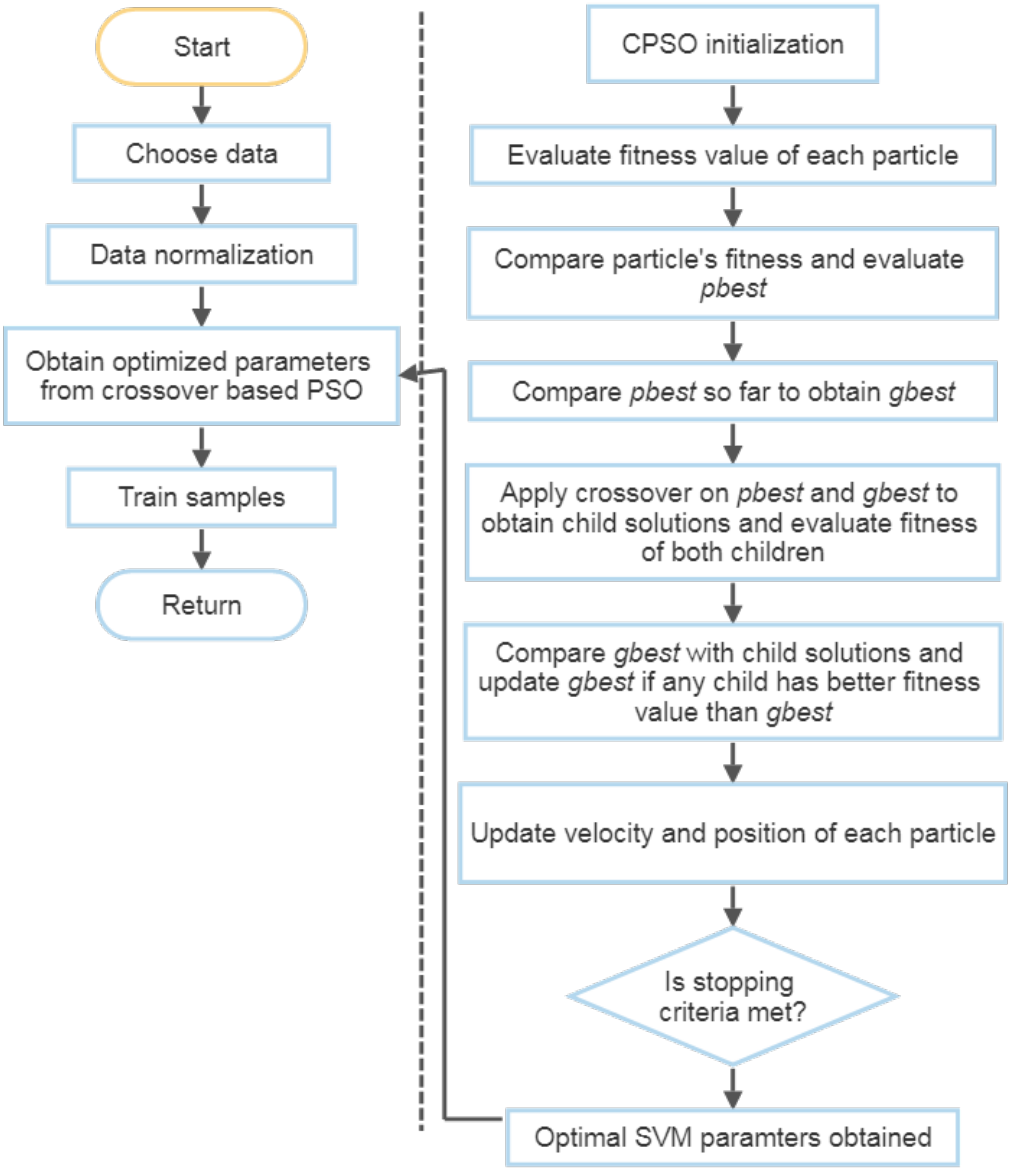


Figure 5.9: Particle swarm optimization based parameter tuning of SVM

In Algorithm 5.2, f_t represents the fitness value. Population P is the dataset in the training set D_{train} as per the data splitting strategy at hand. Training epoch number k is 30 number of iterations, which is also called as termination condition. Also, $batch\_size$ is the total number of target classes, which in this case is 7 and fitness evaluation dataset f_d is the test dataset. To evaluate fitness, each particle solution is mapped to the parameters of CPSOSVM. Thereafter, CPSOSVM is tested on the speaker classification test dataset. Finally, the accuracy has been computed and assigned as fitness value of given particle solution. In Algorithm 5.3, every layer is mapped into particle vector. The acceleration coefficients for every parameter of

Algorithm 5.2 Objective function calculation

Input: Population P, training set D_{train}, training epoch number k, batch size $batch\_size$, fitness evaluation dataset f_d;
Output: Population (P) with fitness;
  for s **in** P **do**
    $i \leftarrow 1$;
    while $i \leq k$ **do**
      Train support vector weights by considering s;
    end while
    $f_t \leftarrow$ Build model on f_d with $batch\_size$ and store it in acc;
    $\mu \leftarrow$ calculate mean of acc
    $f \leftarrow \mu$;
    $P \leftarrow$ Update f of i_d in P;
  end for
  return P

CPSOSVM are implemented as float arrays as shown in (5.9). In (5.9), x and v represent the i^{th} parameters of CPSOSVM, namely, position and its corresponding velocity. Also, Pb_{id} and Gb_d are personal best and global best, respectively. The parameters, w, r_1, and r_2 are same as in traditional PSO given in (5.7).

Algorithm 5.3 Updation of particle using velocity

Input: Individual particle vector i_d, acceleration coefficient array for Pb_{id} c_1, max velocity array v_{max}, inertia weight w, and acceleration coefficient array for Gb_d c_2, ;
Output: Particle vector (i_d);
  for element **in** ind **do**
    $i \leftarrow 0$;
    for $i <$ number of parameters of CPSOSVM **do**
      $x \leftarrow$ the i^{th} byte of the CPSOSVM;
      $(rand_1, rand_2)) \leftarrow$ uniformly generate $rand_1, rand_2$ between [0, 1];
      $v_{new} \leftarrow$ Update velocity according to (5.9);
      $v_{new} \leftarrow$ Apply velocity clamping using v_{max};
      $x_{new} \leftarrow x + v_{new}$
      if $x_{new} > 255$ **then**
        $x_{new} \leftarrow x_{new} - 255$;
      end if
    end for
  end for
  $fitness \leftarrow$ compute updated particle (i_d);
  $(Pb_{id}, Gb_d) \leftarrow$ Update $pbest$ and $gbest$ based upon current $fitness$;
  return i_d

After defining the coefficients, each parameter of CPSOSVM is updated according to evaluated velocity and position. Finally, the parameters of CPSOSVM are evaluated, the new individual is determined, and fitness of updated solution is compared with $gbest$ and $pbest$ to evaluate the updated solutions (v_{new}) as:

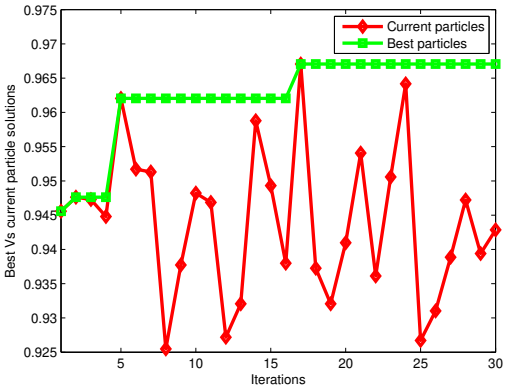
$$v_{new} = w \times v + c_1[i] \times rand_1 \times (Pb_{id} - x) + c_2[i] \times rand_2 \times (Gb_d - x) \quad (5.9)$$

5.7.1 Performance Analysis of CPSOSVM

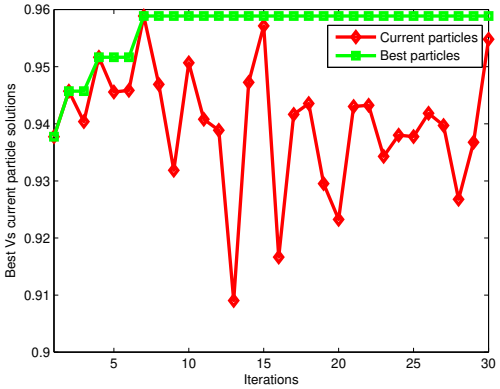
This section discusses the implementation of CPSOSVM. Similar to previous sections, a simulation environment has been designed using the MATLAB $2013a$ to evaluate the effectiveness of the proposed technique. The overall objective of this section is to compare the performance of the existing techniques, EQNN and CPSOSVM by considering same performance metrics, namely, accuracy, F-measure, specificity and sensitivity.

For implementation of CPSOSVM, we have used the acceleration coefficients, $c_1 = 2.0$, $c_2 = 2.0$, and inertia weight $(w)= 0.729$ (Nath *et al.*, 2018; Ghosh *et al.*, 2018)). Maximum number of iterations has been set to 30. Also, the crossover rate of 0.5 with single point crossover has been used. The crossover rate is used to define the probability that how many times crossover will be applied on the obtained solutions. Generally, crossover rate lies within 0 to 1. A value of crossover rate approaching 0 indicates that we want good computational speed and may relax achievable solution. Any value of crossover rate approaching 1 may result in poor computational speed and it does not guarantee that we achieve the optimal solution. Single point crossover means, we initially select two best known solution by using ranking strategy and then combines the first half or first solution with the second half of second selected solution. Similarly, we select the first half of second solution and integrate it with second half of first solution. It results into two child solutions whose performance has been evaluated using fitness function. If any of them has better fitness than best known solution so far, then, it replaces the best known solution so far. Initial value of best particle has been set as zero. Figures 5.10(a) to 5.10(d) show the selection of best particle after comparison with the current particle value with each iteration.

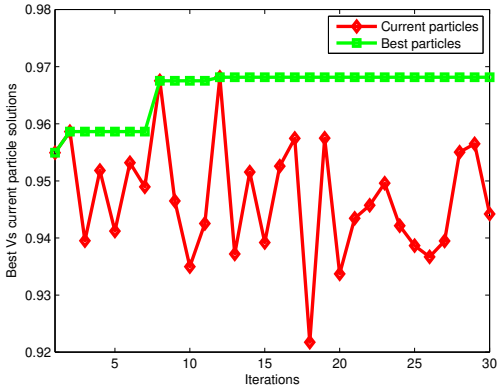
Figures 5.11(a) to 5.11(d) depict the accuracy of 30 iteration for all strategies with CPSOSVM. It has been noted that accuracy of the proposed technique either increased with every iteration or sustained the nearest high value with most of the strategies. This was not the case in existing techniques or EQNN. This increase



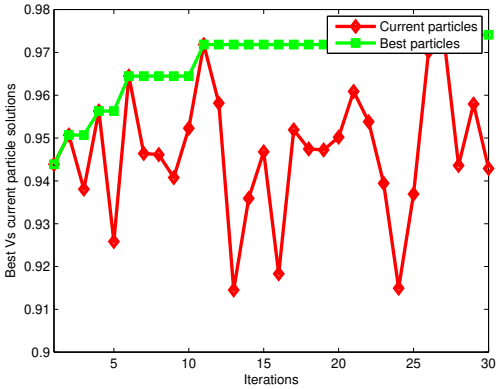
(a) Strategy *a* with CPSOSVM



(b) Strategy *b* with CPSOSVM



(c) Strategy *c* with CPSOSVM



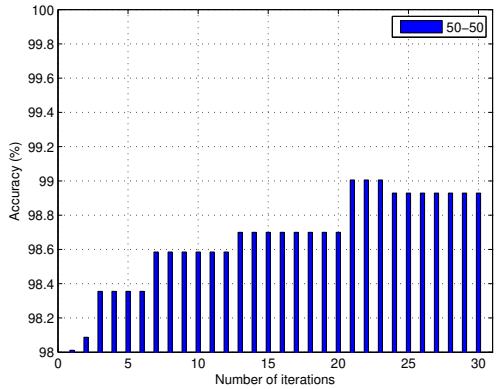
(d) Strategy *d* with CPSOSVM

Figure 5.10: The plot showing selection of best particle after comparison with each iteration's current particle. (a) Strategy *a* with CPSOSVM, (b) Strategy *b* with CPSOSVM, (c) Strategy *c* with CPSOSVM, and (d) Strategy *d* with CPSOSVM.

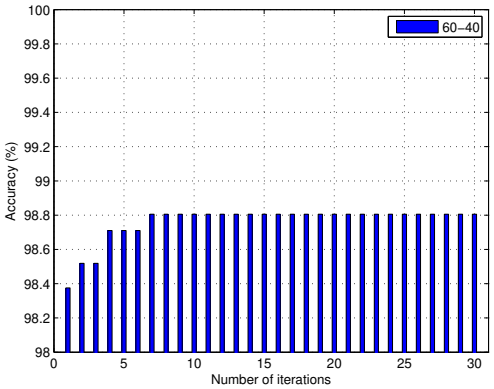
and sustainability is because of tuning of parameters using CPSO. With every iteration, SVM got trained with best parameters selected by CPSO. Also, variance of the accuracies has been sufficiently low for all the strategies when CPSOSVM is used. Table 5.8 contains the obtained confusion matrices of all 7 classes of speakers. It has been found the majority of the obtained classes lie in the true classes, *i.e.*, on the diagonal. Therefore, it leads to good performance results when accuracy, F-score, specificity and sensitivity have been taken into account.

Table 5.9 shows the performance results of CPSOSVM based on these four performance metrics. The performance of the proposed technique has been remarkably better than the existing techniques and EQNN in all the performance metrics.

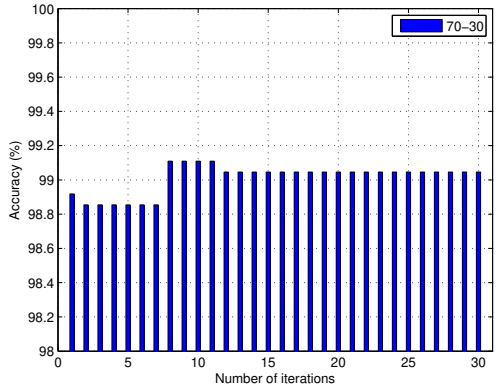
Figures 5.12(a) to 5.12(d) show the comparative analysis between the existing techniques, EQNN and CPSOSVM. In these figures, notched whisker box plot analysis



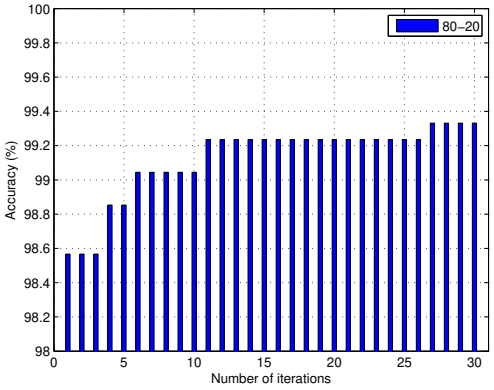
(a) Strategy *a* with CPSOSVM



(b) Strategy *b* with CPSOSVM



(c) Strategy *c* with CPSOSVM



(d) Strategy *d* with CPSOSVM

Figure 5.11: The accuracy of all 30 iterations with different data training and testing strategies with CPSOSVM. (a) Strategy *a* with CPSOSVM, (b) Strategy *b* with CPSOSVM, (c) Strategy *c* with CPSOSVM, and (d) Strategy *d* with CPSOSVM.

has been included for all the performance metrics and mean values obtained for these metrics with the 4 data splitting strategies. The box shows the Inter Quartile Range (IQR). The red line indicates the median of the evaluated values. Notch shows a confidence interval around the median which is (median$+1.57 \times IQR/\sqrt{n}$, median$-1.57 \times IQR/\sqrt{n}$) (McGill *et al.*, 1978). Here, we have considered $n = 4$, which represents the 4 data splitting strategies. If the notch size is small, then the given technique provides consistent results, *i.e.*, with less variation in every experiment and vice versa.

Figure 5.12(a) shows the comparison between the existing speaker classification techniques, EQNN and CPSOSVM in terms of accuracy. It can be concluded from the figures that CPSOSVM gives better accuracy when compared with the existing speaker classification techniques. It has been observed that the proposed CP-

Table 5.8: Confusion matrices for maximum accuracy among 30 iterations of the CPSOSVM (a) Strategy *a* with CPSOSVM, (b) Strategy *b* with CPSOSVM, (c) Strategy *c* with CPSOSVM, and (d) Strategy *d* with CPSOSVM.

Actual Class \ Predicted Class	1	2	3	4	5	6	7
1	115	0	0	0	0	0	0
2	0	85	1	0	0	0	0
3	1	0	100	0	0	0	0
4	0	0	1	104	0	0	0
5	0	0	0	0	118	4	0
6	0	0	1	0	0	114	8
7	0	0	1	0	0	11	83

(a) Strategy *a* with CPSOSVM

Actual Class \ Predicted Class	1	2	3	4	5	6	7
1	92	0	0	0	0	0	0
2	0	79	0	0	0	0	0
3	0	0	89	0	0	2	0
4	0	0	0	78	0	0	0
5	0	0	0	0	83	1	0
6	0	0	0	0	0	86	8
7	0	0	1	0	0	11	68

(b) Strategy *b* with CPSOSVM

Actual Class \ Predicted Class	1	2	3	4	5	6	7
1	72	0	0	0	0	0	0
1	0	56	0	0	0	0	0
1	0	0	59	0	0	0	0
1	0	0	0	59	0	0	0
1	0	0	0	0	68	1	0
1	0	0	0	0	0	62	2
1	0	0	0	0	0	12	58

(c) Strategy *c* with CPSOSVM

Actual Class \ Predicted Class	1	2	3	4	5	6	7
1	48	0	0	0	0	0	0
1	0	46	0	0	0	0	0
1	0	0	35	0	0	0	0
1	0	0	2	46	0	0	0
1	0	0	0	0	46	0	0
1	07	0	0	0	0	38	0
1	0	0	0	0	0	5	33

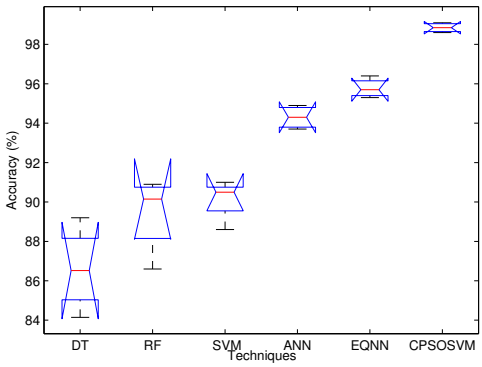
(d) Strategy *d* with CPSOSVM

Table 5.9: Mean and variance (dd±d.d) of the performance metrics with CPSOSVM for 30 iterations and 4 strategies

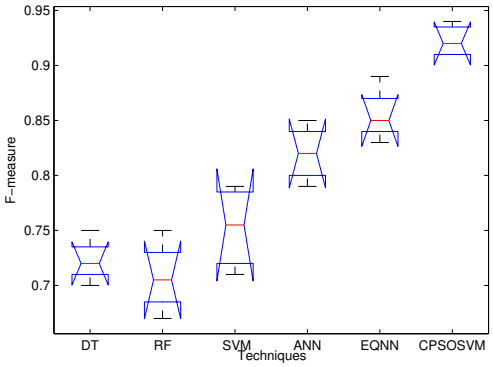
Performance Metric	Strategy *a*	Strategy *b*	Strategy *c*	Strategy *d*
Accuracy	98.6±0.067	98.7±0.010	99.0±0.007	99.1±0.049
F-measure	0.91±0.0009	0.91±0.0008	0.93±0.0005	0.94±0.0002
Specificity	0.94±0.0005	0.94±0.0004	0.96±0.0003	0.96±0.0003
Sensitivity	0.97±0.0002	0.96±0.0003	0.97±0.0002	0.97±0.0002

SOSVM is able to improve the accuracy of the model by 3.0% when compared with EQNN.

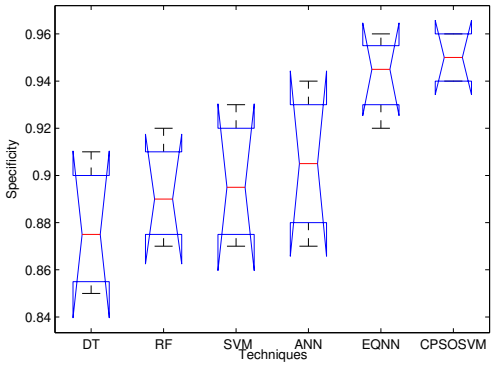
Figure 5.12(b) demonstrates F-measure analysis between CPSOSVM, EQNN and the existing speaker classification techniques. It has been found that the proposed technique provides consistently better F-measure values as compared to the existing techniques. Overall analysis shows that CPSOSVM outperforms EQNN and existing techniques by showing an improvement of 0.07 in F-measure.



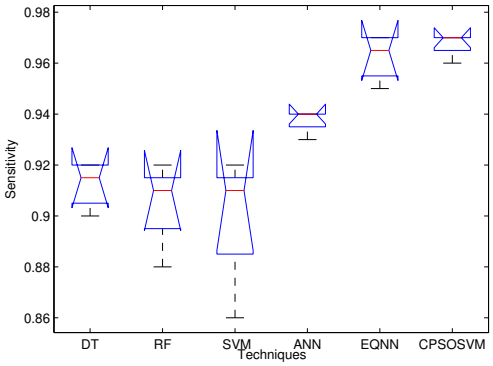
(a) Analysis of accuracy



(b) Analysis of F-measure



(c) Analysis of specificity



(d) Analysis of sensitivity

Figure 5.12: The performance analysis of all 30 iterations with 4 data training and testing strategies with CPSOSVM. (a) Analysis of accuracy, (b) Analysis of F-measure (c) Analysis of specificity, and (d) Analysis of sensitivity.

Figure 5.12(c) shows that CPSOSVM has performed consistently well as the variance in specificity is remarkably smaller as compared to the existing speaker classification techniques. It has been found that CPSOSVM achieved improvement in specificity values by 0.05 which shows CPSOSVM performs better than the existing speaker classification techniques.

Figure 5.12(d) indicates the sensitivity analysis between CPSOSVM and the existing speaker classification techniques. It has again been found that CPSOSVM performed well with respect to sensitivity value as compared to the existing techniques. It has shown an improvement of 0.03 in sensitivity over the existing techniques.

5.8 Summary

In this chapter, a novel speaker classification technique has been proposed. Initially, the existing classification techniques, their issues and various performance metrics have been discussed. A unique data splitting method has also been proposed in this work. This has been followed by the implementation the existing techniques and results have been analysed. Then, EQNN has been proposed which outperformed the existing techniques. But, it had been found that this technique suffers from slight over-fitting and parameter tuning remains an open issue. Therefore, CPSOSVM has then been proposed to tackle these two issues. It has been found that both the proposed speaker classification techniques outperform the competitive machine learning techniques to recognize Punjabi language native speakers. CPSOSVM has shown remarkable improvement in results over EQNN and existing techniques.

Chapter 6

Conclusion and Future Work

This chapter is devoted to conclude the work done in this thesis and also to highlight the significant contribution of this work. This chapter concludes the thesis by explaining the outcome of each chapter. Future directions of the research work have also been presented towards the end of this chapter.

6.1 Conclusion

It has been observed from the literature that speech is the most natural means of communication between humans. Human beings start speaking without any tool or any explicit education. The environment surrounding them helps them to learn the art of speaking. This is the reason that it has been one of the prominent areas of research for many years. Therefore, in this thesis, initially, we have designed a Phonetic Engine (PE) for the Punjabi language which converts the spoken word into set of phonetic symbols.

This PE has been developed using HTK wherein, MFCCs have been used for feature extraction and HMM has been used as a classifier. Data for this work has been collected from different regions of Punjab to cover all the major dialects of the language. Also, data has been collected from the speakers of different age groups. The data collection process has been performed in three modes, namely, read speech mode, lecture speech mode and conversational speech mode. The performance of PE has been evaluated using three different approaches: (i) by increasing the amount of data from 3 hours to 5 hours, (ii) by decreasing the number of symbols from 49 to 29 and (iii) by increasing MFCC dimensions from 12 to 36. An accuracy of 72.3% has been achieved in this work when 5 hours data with 29 symbols and 12 MFCCs

were employed.

Further, the read speech mode data of 7 speakers including 4 females and 3 males has been used to explore text-independent speaker classification. Speaker classification is one of the popular biometric identification techniques, which classifies the speaker's identity by considering the speech of the person. From the existing literature, it has been found that the existing speaker classification techniques suffer from over-fitting and parameter tuning issues. Initially, four existing classifiers, namely, decision tree, random forest, support vector machine and artificial neural network, have been considered as base classifiers. All these base classifiers have been implemented on the data of 7 speakers in MATLAB. The performance of these classifiers has been evaluated using four performance metrics, namely, accuracy, F-measure, specificity and sensitivity. After that, to overcome the over-fitting issue, a novel Ensemble-based quantum neural network (EQNN) has been designed in this thesis. The EQNN based models have been trained using quantum neural network with four data splitting strategies. It selects the error matrix from each trained model for a strategy at hand, and concatenates the matrix with training data of next strategy. In the end, ensembling of the output of one strategy has been done with the input of next strategy. Extensive experiments have been carried out using the proposed technique and existing competitive techniques on speaker recognition data. It has been concluded that the proposed technique outperforms existing speaker recognition techniques in terms of all the performance metrics. It has further been observed from the training performance of the EQNN that it still suffers from slight over-fitting.

The Crossover based Particle Swarm Optimization with Support Vector Machine (CPSOSVM) has also been implemented in this work. In CPSOSVM, Particle Swarm Optimization (PSO) has been used to tune the parameters of Support Vector Machine (SVM). The crossover has been used to avoid the training process getting struck in local optima. It has been applied on the global best and personal best particles to obtain two more solutions, which are called as child solutions. Thereafter, fitness of both child solutions has been evaluated. If one of the obtained child solutions has better fitness than global best then the global best solution has been replaced with the this child solution. Whenever the crossover based particle swarm optimization satisfies stopping criteria, the obtained global best solution is used to select the parameters of support vector machine.

Extensive experiments have again been carried out using the proposed technique and the existing competitive speaker recognition techniques. It has been concluded that the proposed technique outperforms existing speaker recognition techniques in

terms of accuracy, F-measure, specificity and sensitivity, by 3.0%, 0.07, 0.05 and 0.03, respectively. Therefore, the proposed technique is more efficient for real time speaker recognition systems.

6.2 Future Work

The work presented in this thesis can be extended in the following future directions.

(i) This work has been done for the read speech mode data. It can be extended to the lecture speech mode data and the conversational speech mode data. It can also be extended to syllable level recognition.

(ii) The length of silence region in speech can further be explored.

(iii) Various APIs can be developed from the Phonetic Engine that can further be used in different recognition applications and in the development of a search engine for regional languages.

(iv) In this work, the speaker classification strategies have been implemented using the data of 7 speakers. One can work on the dataset containing larger number of speakers.

(v) The performance of classification can possibly be increased by taking more samples of data. One can thus work on increasing the performance of the recognition models.

(vi) Crossover rate and crossover point can further be explored to achieve better performance.

(vii) Our aim was to propose and use the evolutionary techniques for tuning the parameters. However, one can work on the variants of PSO in order to improve the efficiency of the model.

(viii) Further, evolutionary techniques based parameter tuning of EQNN can also be explored in future.

(ix) We have explored only one training algorithm (*trainlm*) while implementing EQNN; Other algorithms can also be used to enhance the performance of the classifiers.

(x) This work can further is explored with the deep learning models and other acoustics models.

List of publications

Conference Publication

- Rupinderdeep Kaur, R.K. Sharma and Parteek Kumar, "Building a Text-to-Speech System for Punjabi Language", 5th International Conference of Advanced Computer Science and Information Technology, pp. 71-87, June, 2017, DOI: 10.5121/csit.2017.70806.

Journal Publications

- Rupinderdeep Kaur, R.K. Sharma and Parteek Kumar, "An efficient speaker recognition using quantum neural network", Modern Physics Letters B, Impact factor: 0.731, DOI: 10.1142/S0217984918503840.
- Rupinderdeep Kaur, R.K. Sharma and Parteek Kumar, "HMM-based phonetic engine for continuous speech of a regional language", Modern Physics Letters B, Impact factor: 0.731, DOI: 10.1142/S0217984919502956.
- Rupinderdeep Kaur, R.K. Sharma and Parteek Kumar, "Speaker classification with support vector machine and crossover based particle swarm optimization", International Journal of Pattern Recognition and Artificial Intelligence, Impact Factor: 1.11. [Accepted]

Appendix A

Appendix A.1

Consonants in Gurmukhi script

ੳ	ਅ	ੲ	ਸ	ਹ
ਕ	ਖ	ਗ	ਘ	ਙ
ਚ	ਛ	ਜ	ਝ	ਞ
ਟ	ਠ	ਡ	ਢ	ਣ
ਤ	ਥ	ਦ	ਧ	ਨ
ਪ	ਫ	ਬ	ਭ	ਮ
ਯ	ਰ	ਲ	ਵ	ੜ

Special consonants in Gurmukhi script

ਸ਼	ਖ਼	ਗ਼	ਜ਼	ਫ਼	ਲ਼

Vowel diacritics in Gurmukhi script

◌ੈ	◌ੈ	◌ਾ	◌ੀ	◌ੂ	◌ੌ	◌ੇ	◌ਿ	◌ੁ

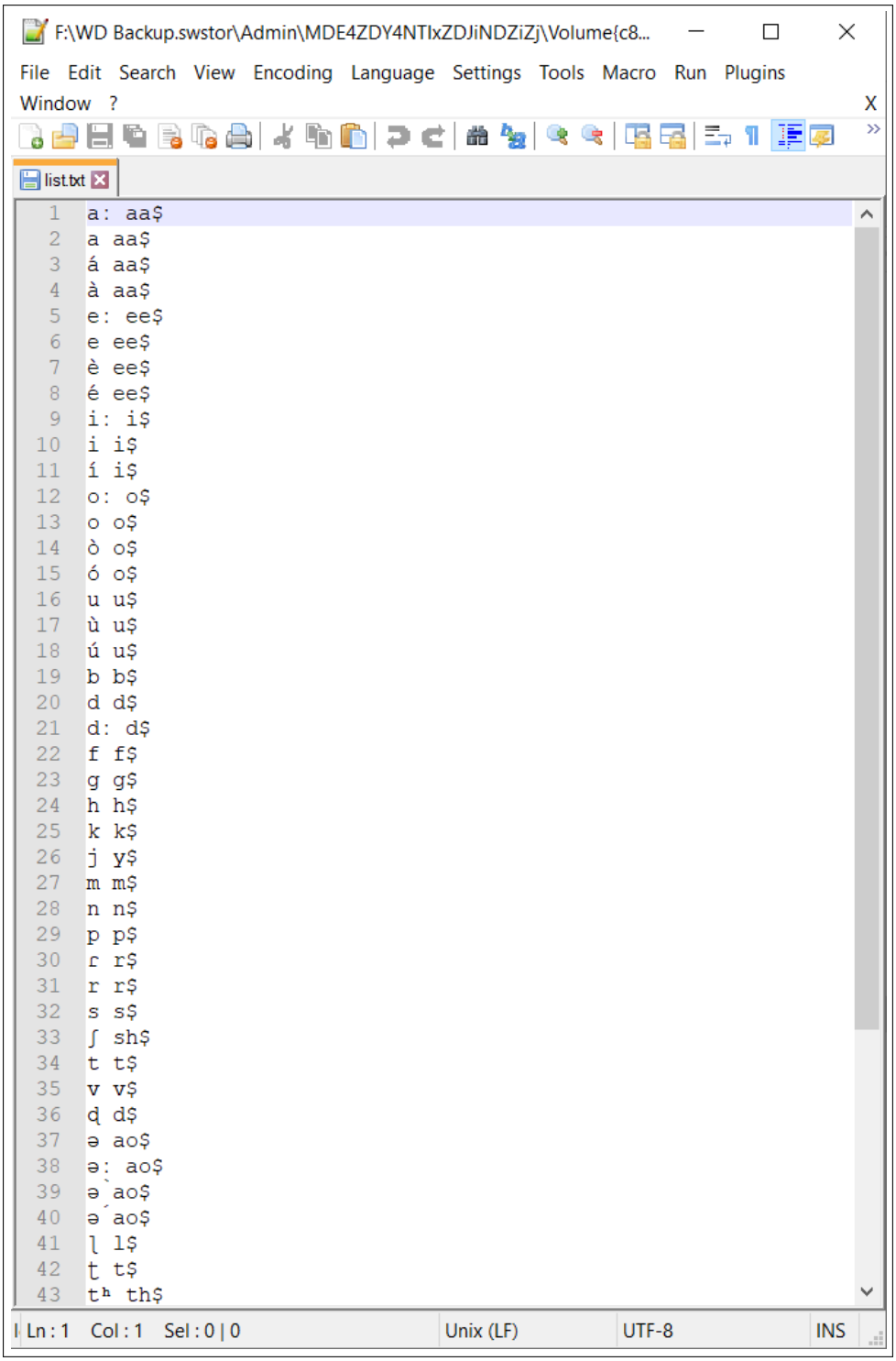
Auxiliary diacritics and subscript letters in Gurmukhi script

◌ਂ	◌ੰ	◌ੱ	◌੍ਵ	◌੍ਹ	◌੍ਰ

Appendix B

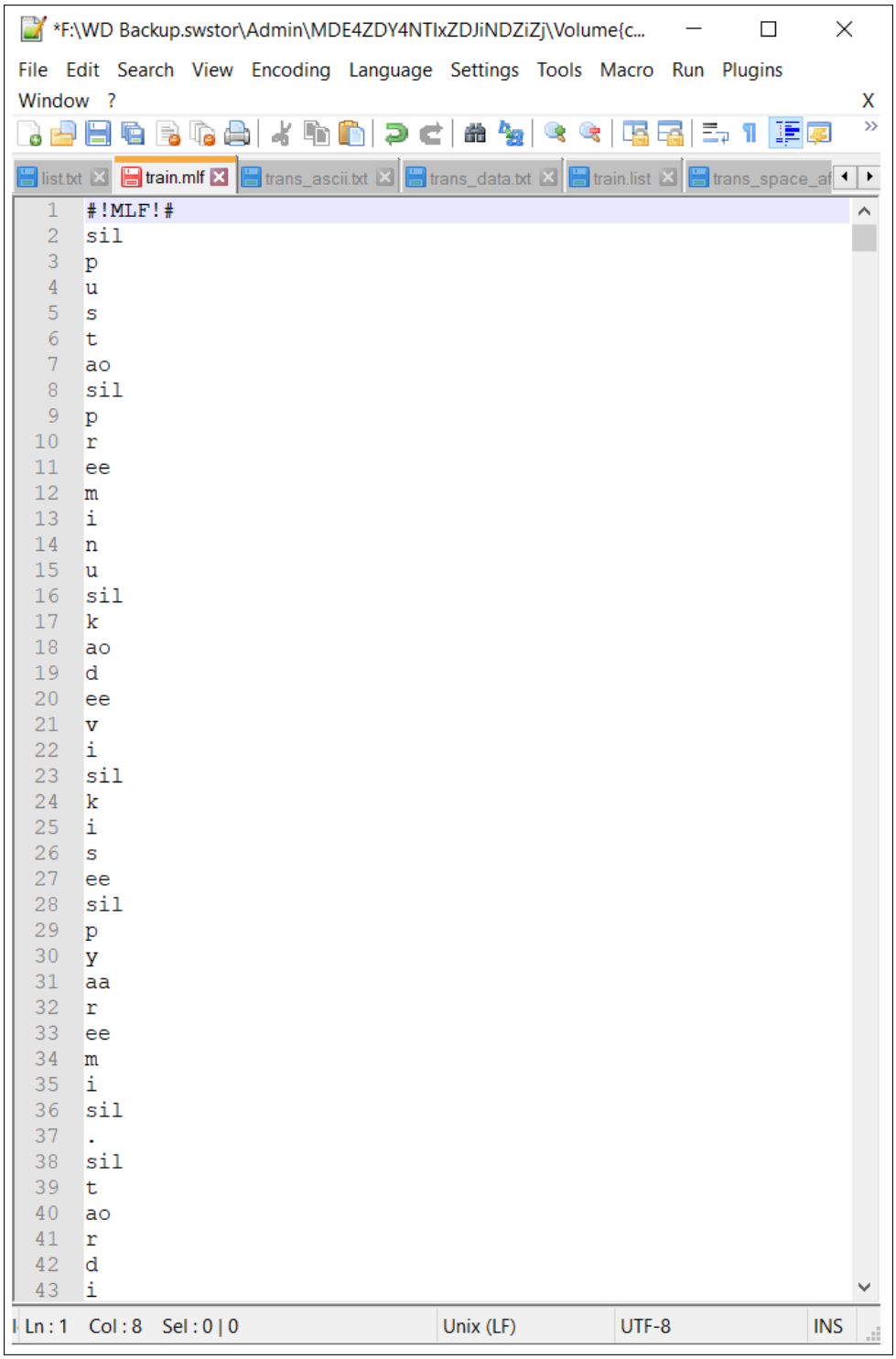
Appendix B.1

Snapshot of *list.txt* used for the development of PE



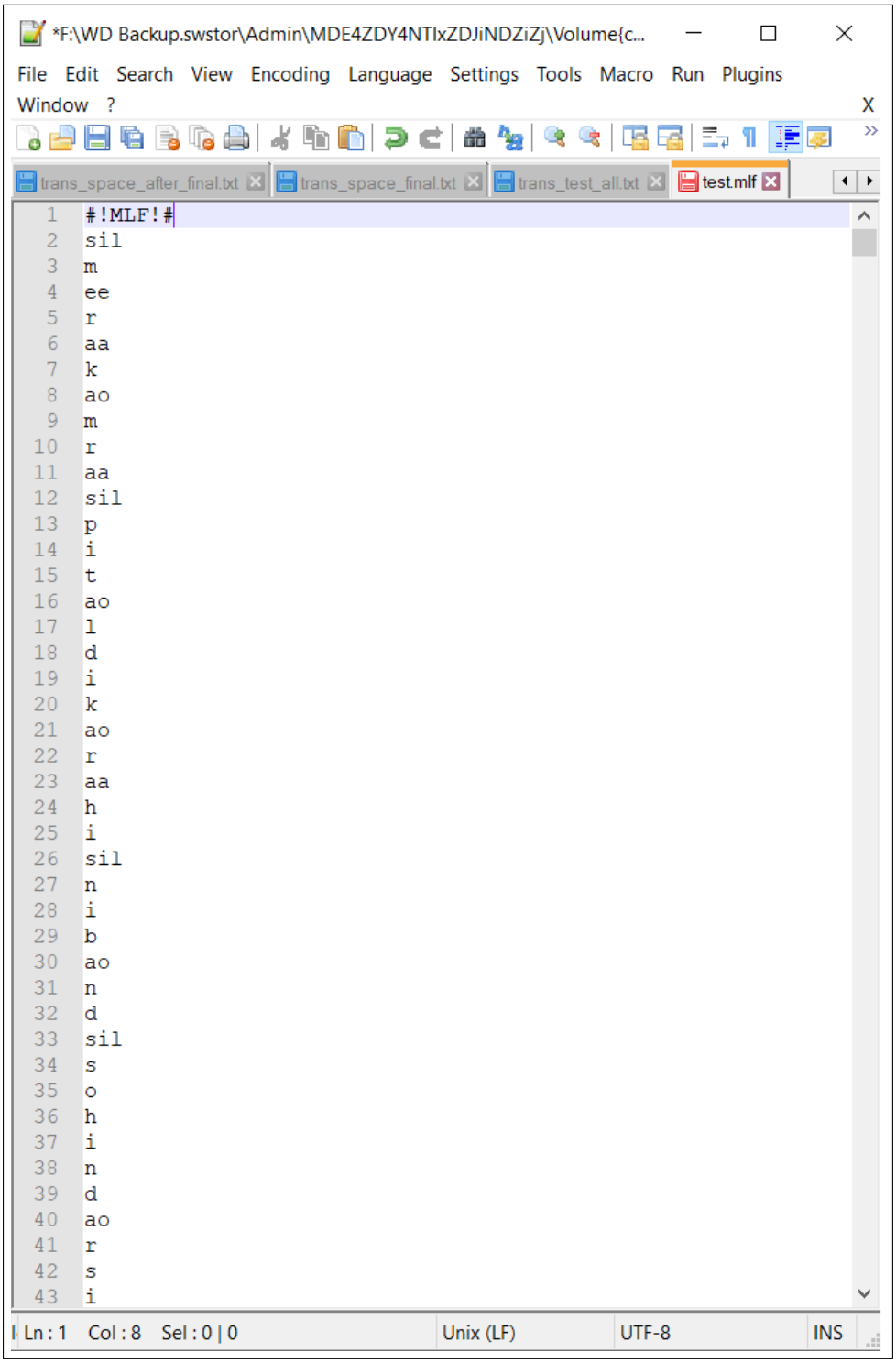
Appendix B.2

Snapshot *train.mlf* used for the development of PE



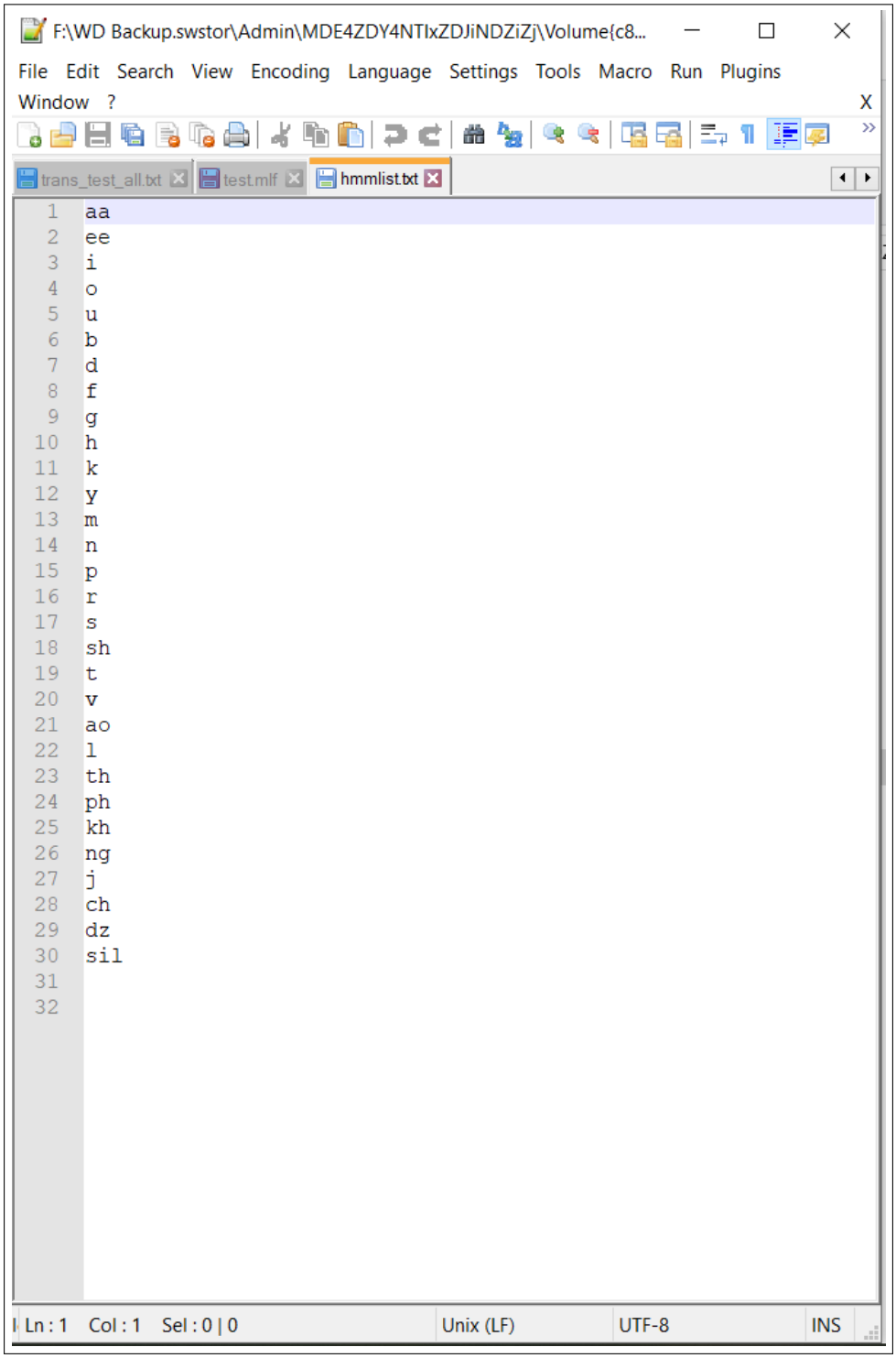
Appendix B.3

Snapshot of *test.mlf* used for the development of PE



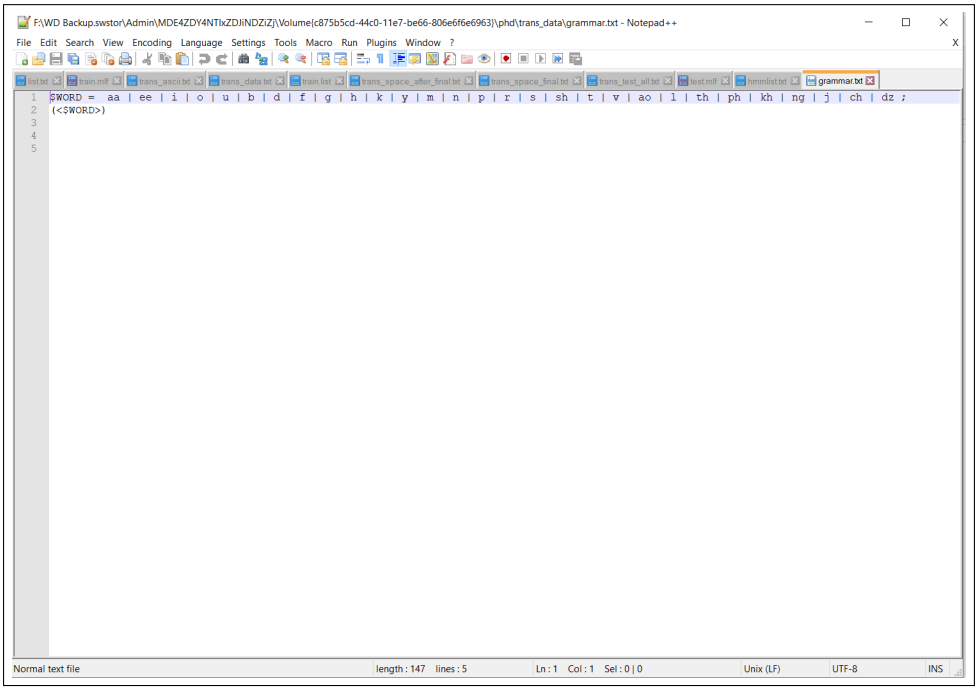
Appendix B.4

Snapshot of *hmmlist.txt* used for the development of PE



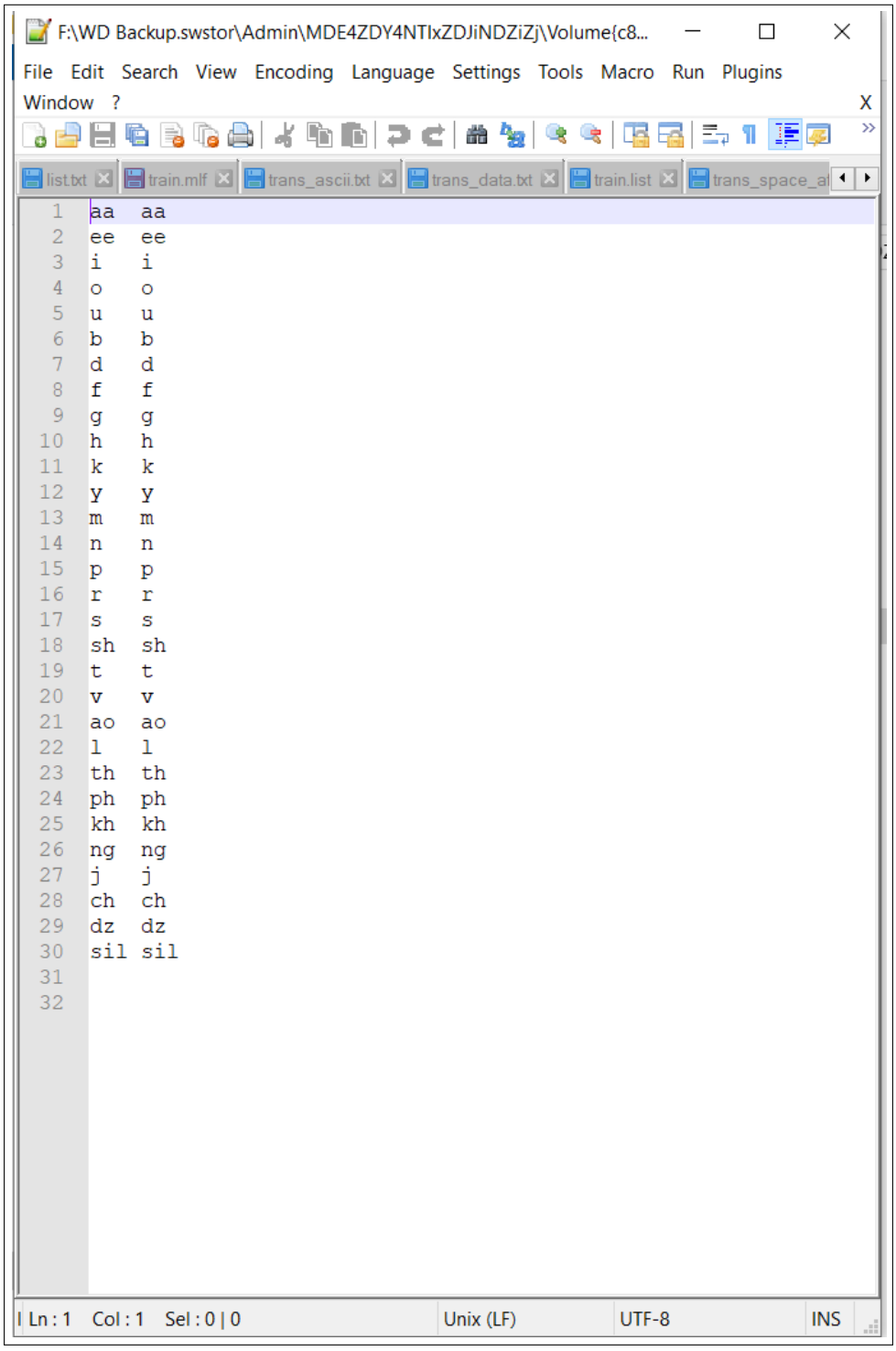
Appendix B.5

Snapshot of *grammar.txt* used for the development of PE



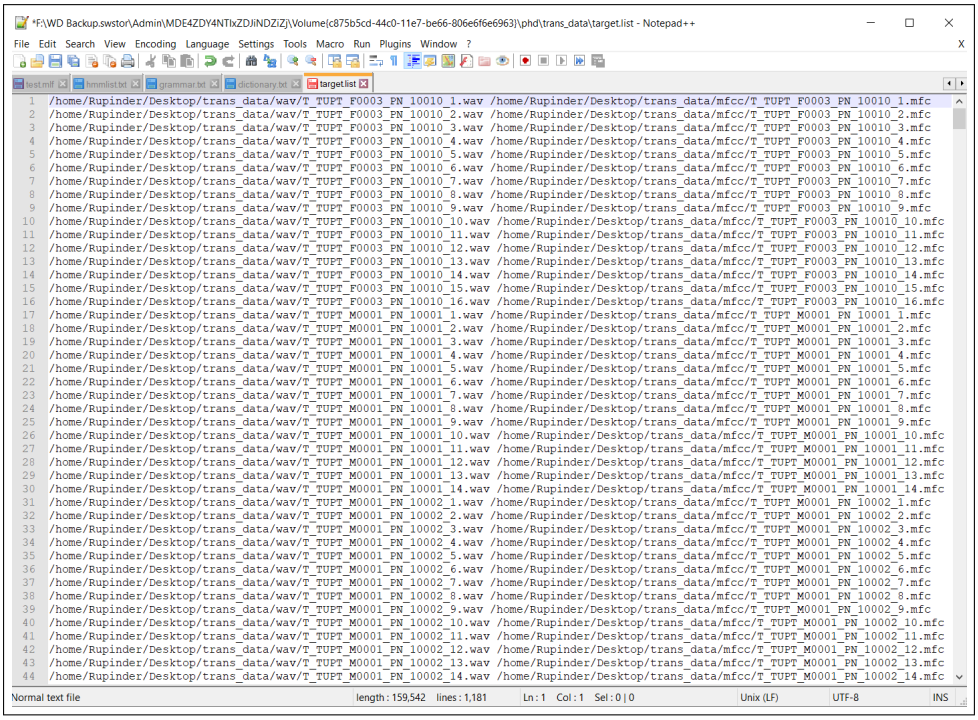
Appendix B.6

Snapshot of *dictionary.txt* used for the development of PE



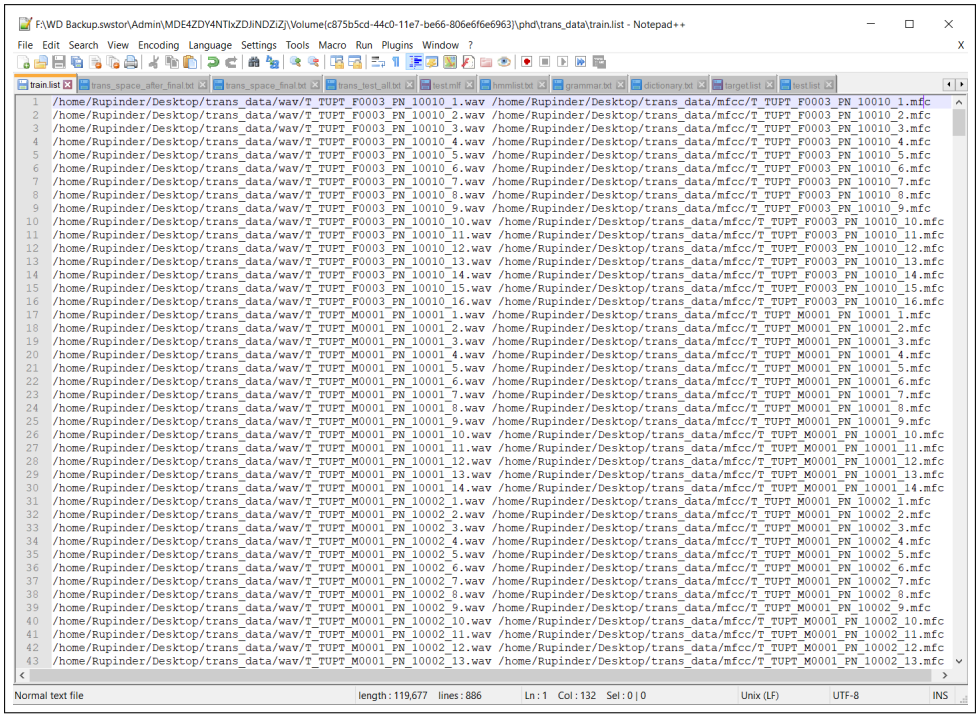
Appendix B.7

Snapshot of *target.list* used for the development of PE



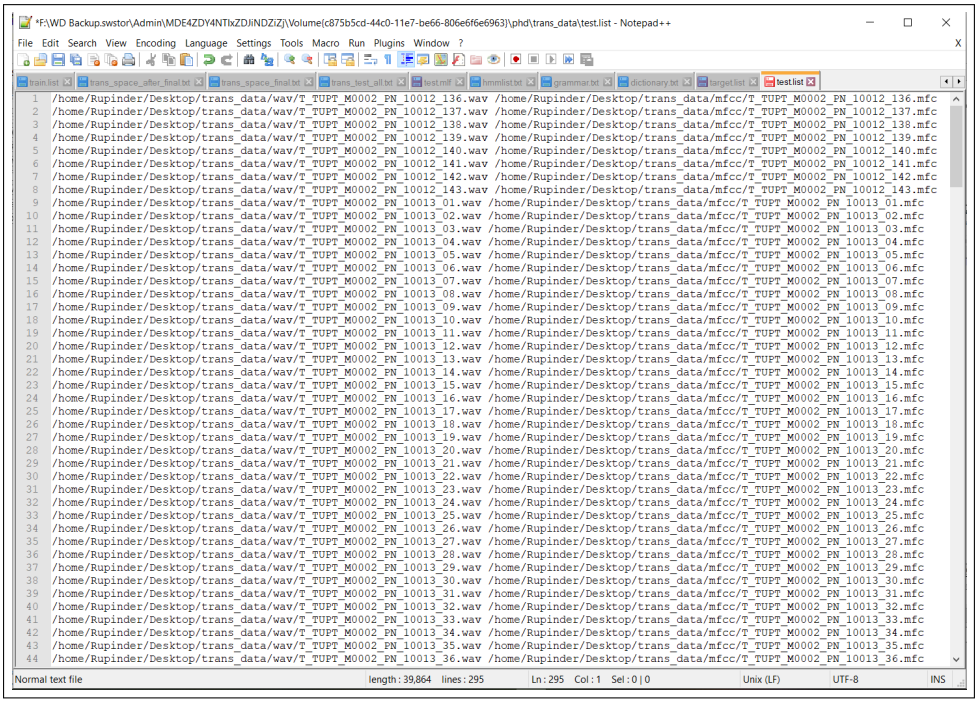
Appendix B.8

Snapshot of *train.list* used for the development of PE



Appendix B.9

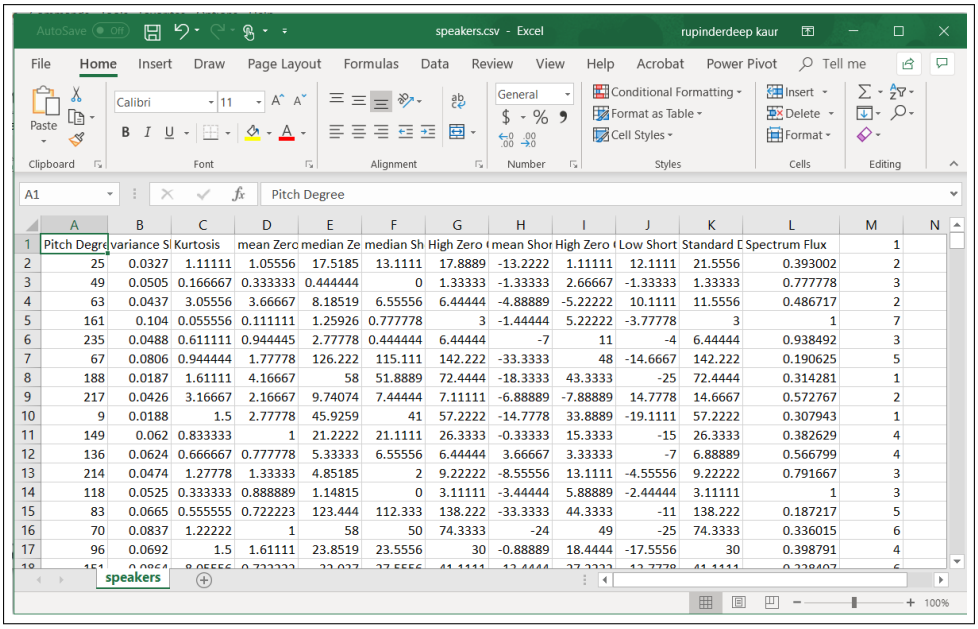
Snapshot of *test.list* used for the development of PE



Appendix C

Appendix C.1

Sample of feature vector used for building the speaker classification models



Pitch Degre	variance S	Kurtosis	mean Zero	median Ze	median Sh	High Zero	mean Shor	High Zero	Low Short	Standard D	Spectrum Flux	
25	0.0327	1.11111	1.05556	17.5185	13.1111	17.8889	-13.2222	1.11111	12.1111	21.5556	0.393002	2
49	0.0505	0.166667	0.333333	0.444444	0	1.33333	-1.33333	2.66667	-1.33333	1.33333	0.777778	3
63	0.0437	3.05556	3.66667	8.18519	6.55556	6.44444	-4.88889	-5.22222	10.1111	11.5556	0.486717	2
161	0.104	0.055556	0.111111	1.25926	0.777778	3	-1.44444	5.22222	-3.77778	3	1	7
235	0.0488	0.611111	0.944445	2.77778	0.444444	6.44444	-7	11	-4	6.44444	0.938492	3
67	0.0806	0.944444	1.77778	126.222	115.111	142.222	-33.3333	48	-14.6667	142.222	0.190625	5
188	0.0187	1.61111	4.16667	58	51.8889	72.4444	-18.3333	43.3333	-25	72.4444	0.314281	1
217	0.0426	3.16667	2.16667	9.74074	7.44444	7.11111	-6.88889	-7.88889	14.7778	14.6667	0.572767	2
9	0.0188	1.5	2.77778	45.9259	41	57.2222	-14.7778	33.8889	-19.1111	57.2222	0.307943	1
149	0.062	0.833333	1	21.2222	21.1111	26.3333	-0.33333	15.3333	-15	26.3333	0.382629	4
136	0.0624	0.666667	0.777778	5.33333	6.55556	6.44444	3.66667	3.33333	-7	6.88889	0.566799	4
214	0.0474	1.27778	1.33333	4.85185	2	9.22222	-8.55556	13.1111	-4.55556	9.22222	0.791667	3
118	0.0525	0.333333	0.888889	1.14815	0	3.11111	-3.44444	5.88889	-2.44444	3.11111	1	3
83	0.0665	0.555555	0.722223	123.444	112.333	138.222	-33.3333	44.3333	-11	138.222	0.187217	5
70	0.0837	1.22222	1	58	50	74.3333	-24	49	-25	74.3333	0.336015	6
96	0.0692	1.5	1.61111	23.8519	23.5556	30	-0.88889	18.4444	-17.5556	30	0.398791	4

References

Abdel-Hamid, O. and Jiang, H. (2013). Fast Speaker Adaptation of Hybrid NN/HMM Model for Speech Recognition Based on Discriminative Learning of Speaker Code. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7942–7946.

Abushariah, A. A., Gunawan, T. S., Khalifa, O. O., and Abushariah, M. A. (2010). English Digits Speech Recognition System Based on Hidden Markov Models. In *IEEE International Conference on Computer and Communication Engineering (ICCCE'10)*, pages 1–5.

Adami, A. G., Mihaescu, R., Reynolds, D. A., and Godfrey, J. J. (2003). Modeling Prosodic Dynamics for Speaker Recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, volume 4, pages IV–788.

Al-Anzi, F. S. and AbuZeina, D. (2017). The Capacity of Mel Frequency Cepstral Coefficients for Speech Recognition. *International Journal of Computer and Information Engineering*, **11**(10), 1149 – 1153.

Al-Qatab, B. A. Q. and Ainon, R. N. (2010). Arabic Speech Recognition Using Hidden Markov Model Toolkit(HTK). In *International Symposium on Information Technology*, volume 2, pages 557–562.

Ali, H., Tran, S. N., Benetos, E., and Garcez, A. S. d. (2018a). Speaker Recognition with Hybrid Features from a Deep Belief Network. *Neural Computing and Applications*, **29**(6), 13–19.

Ali, H., Tran, S. N., Benetos, E., and dAvila Garcez, A. S. (2018b). Speaker recognition with hybrid features from a deep belief network. *Neural Computing and Applications*, **29**.

Aljawarneh, S., Yassein, M. B., and Aljundi, M. (2017). An Enhanced J48 Classification Algorithm for the Anomaly Intrusion Detection Systems. *Cluster Computing*, pages 1–17.

Alotaibi, Y. (2008). Comparative Study of ANN and HMM to Arabic Digits Recognition Systems. *Journal of King Abdulaziz University-Engineering Sciences*, **19**, 43–60.

Amodei, D., Anubhai, R., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Chen, J., Chrzanowski, M., Coates, A., Diamos, G., Elsen, E., Engel, J. H., Fan, L., Fougner, C., Han, T., Hannun, A. Y., Jun, B., LeGresley, P., Lin, L., Narang, S., Ng, A. Y., Ozair, S., Prenger, R., Raiman, J., Satheesh, S., Seetapun, D., Sengupta, S., Wang, Y., Wang, Z., Wang, C., Xiao, B., Yogatama, D., Zhan, J., and Zhu, Z. (2015). Deep speech 2: End-to-end speech recognition in english and mandarin. *CoRR*, **abs/1512.02595**.

Angelini, B., Brugnara, F., Falavigna, D., Giuliani, D., Gretter, R., and Omologo, M. (1994). Speaker Independent Continuous Speech Recognition Using An Acoustic-Phonetic Italian Corpus. In *Third International Conference on Spoken Language Processing*, pages 1391–1394.

Anggraeni, D., Sanjaya, W. S. M., Nurasyidiek, M. Y. S., and Munawwaroh, M. (2018). The Implementation of Speech Recognition using Mel-Frequency Cepstrum Coefficients (MFCC) and Support Vector Machine (SVM) method based on Python to Control Robot Arm. *IOP Conference Series: Materials Science and Engineering*, **288**, 012042.

Audhkhasi, K., Osoba, O., and Kosko, B. (2013). Noisy Hidden Markov Models for Speech Recognition. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6.

Azmi, M. M. and Tolba, H. (2008). Syllable-Based Automatic Arabic Speech Recognition in Different Conditions of Noise. In *2008 9th International Conference on Signal Processing*, pages 601–604.

Bahlmann, C., Haasdonk, B., and Burkhardt, H. (2002). Online Handwriting Recognition with Support Vector Machines - a Kernel Approach. In *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition*, pages 49–54.

Bartlett, S., Kondrak, G., and Cherry, C. (2008). Automatic Syllabification with Structured SVMs for Letter-to-Phoneme Conversion. In *Proceedings of ACL-08: HLT*, pages 568–576, Columbus, Ohio. Association for Computational Linguistics.

Bartlett, S., Kondrak, G., and Cherry, C. (2009). On the Syllabification of Phonemes. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 308–316, Boulder, Colorado. Association for Computational Linguistics.

Beer, K., Bondarenko, D., Farrelly, T., Osborne, T. J., and Salzmann, R. (2020). Training deep quantum neural networks. *Nature Communications*, **11**, 808.

Belgiu, M. and Drăguţ, L. (2016). Random Forest in Remote Sensing: A Review of Applications and Future Directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, **114**, 24–31.

Bhuriyakorn, P., Punyabukkana, P., and Suchato, A. (2008). A Genetic Algorithm-Aided Hidden Markov Model Topology Estimation for Phoneme Recognition of Thai Continuous Speech. In *2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, pages 475–480. IEEE.

Bietti, A., Bach, F., and Cont, A. (2015). An Online EM Algorithm in Hidden (semi-)Markov Models for Audio Segmentation and Clustering. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1881–1885.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA. ACM.

Broughton, M., Verdon, G., McCourt, T., Martinez, A. J., Yoo, J. H., Isakov, S. V., Massey, P., Halavati, R., Niu, M. Y., Zlokapa, A., Peters, E., Lockwood, O., Skolik, A., Jerbi, S., Dunjko, V., Leib, M., Streif, M., Dollen, D. V., Chen, H., Cao, S., Wiersema, R., Huang, H.-Y., McClean, J. R., Babbush, R., Boixo, S., Bacon, D., Ho, A. K., Neven, H., and Mohseni, M. (2021). Tensorflow quantum: A software framework for quantum machine learning.

Brugnara, F., Falavigna, D., and Omologo, M. (1993). Automatic Segmentation and Labeling of Speech Based on Hidden Markov Models. *Speech Communication*, **12**(4), 357–370.

Burges, C. J. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data mining and knowledge discovery*, **2**(2), 121–167.

C., M. V. and Radha, V. (2012). Speaker Independent Isolated Speech Recognition System for Tamil Language using HMM. *Procedia Engineering*, **30**, 1097 – 1102. International Conference on Communication Technology and System Design 2011.

Chavan, R. S. and Sable, G. S. (2013). An Implementation of Text Dependent Speaker Independent Isolated Word Speech Recognition Using HMM. *International Journal Of Engineering Sciences & Research Technology*, **2**(9).

Chen, F.-H. and Howard, H. (2016). An Alternative Model for the Analysis of Detecting Electronic Industries Earnings Management Using Stepwise Regression, Random Forest, and Decision Tree. *Soft Computing*, **20**(5), 1945–1960.

Chen, K. and Salman, A. (2011). Learning Speaker-Specific Characteristics With a Deep Neural Architecture. *IEEE Transactions on Neural Networks*, **22**(11), 1744–1756.

Choudhary, A., Chauhan, M., and Gupta, M. G. (2013). Automatic Speech Recognition System for Isolated and Connected Words of Hindi Language by Using Hidden Markov Model toolkit (HTK). In *International Conference on Emerging Trends in Engineering and Technology*, pages 847–853.

Chung, J. S., Nagrani, A., and Zisserman, A. (2018). VoxCeleb2: Deep Speaker Recognition. In *Proc. Interspeech 2018*, pages 1086–1090.

Cortes, C. and Vapnik, V. (1995). Support-Vector Networks. *Machine learning*, **20**(3), 273–297.

Deekshitha, G., Thennattil, J. J., and Mary, L. (2015). Segmentation of Continuous Speech for Broad Phonetic Engine. In *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pages 1–5.

Dhanjal, S. and Bhatia, S. S. (2013). Development of a Standard Text and Speech Corpus for the Punjabi Language. In *2013 International Conference Oriental*

COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE), pages 1–6.

Dhingra, S. D., Nijhawan, G., and Pandit, P. (2013). Isolated Speech Recognition Using MFCC and DTW. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, **2**(8), 4085–4092.

Dua, M., Aggarwal, R., Kadyan, V., and Dua, S. (2012). Punjabi Automatic Speech Recognition Using HTK. *International Journal of Computer Science Issues (IJCSI)*, **9**(4), 359–364.

Eberhart, R. C. (2007). *Computational Intelligence: Concepts to Implementations*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

El-Alfy, E. M., Thampi, S. M., Takagi, H., Piramuthu, S., and Hanne, T., editors (2015). *Advances in Intelligent Informatics - Proceedings of the Third International Symposium on Intelligent Informatics, ISI 2014, September 24-27, 2014, Greater Noida, Delhi, India*, volume 320 of *Advances in Intelligent Systems and Computing*. Springer.

Elshafei, M., Al-Muhtaseb, H., and Al-Ghamdi, M. (2008). Speaker-Independent Natural Arabic Speech Recognition System. In *The International Conference on Intelligent Systems*.

Fang Sun and Guangrui Hu (1998). Speech Recognition Based on Genetic Algorithm for Training HMM. *Electronics Letters*, **34**(16), 1563–1564.

Friedrichs, F. and Igel, C. (2005). Evolutionary Tuning of Multiple SVM Parameters. *Neurocomputing*, **64**, 107–117.

Fujimura, O. (1975). Syllable as a Unit of Speech Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **23**(1), 82–87.

Gaikwad, S., Gawali, B., Yannawar, P., and Mehrotra, S. (2011). Feature Extraction Using Fusion MFCC for Continuous Marathi Speech Recognition. In *2011 Annual IEEE India Conference*, pages 1–5.

Garimella, S., Mallidi, S. H., and Hermansky, H. (2012). Regularized Auto-Associative Neural Networks for Speaker Verification. *IEEE Signal Processing Letters*, **19**(12), 841–844.

Ge, Z., Iyer, A. N., Cheluvaraja, S., Sundaram, R., and Ganapathiraju, A. (2017). Neural Network Based Speaker Classification and Verification Systems with Enhanced Features. In *2017 Intelligent Systems Conference (IntelliSys)*, pages 1089–1094. IEEE.

Ghahabi, O. and Hernando, J. (2014). Deep Belief Networks for i-vector Based Speaker Recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 1700–1704. IEEE.

Ghosh, S., Nath, S., Biswas, R., Venkateswaran, P., Sing, J. K., and Sarkar, S. K. (2018). PSO Variants and its Comparison with Firefly Algorithm in Solving VLSI Global Routing Problem. In *2018 IEEE Electron Devices Kolkata Conference (EDKCON)*, pages 513–518. IEEE.

Gokgoz, E. and Subasi, A. (2015). Comparison of Decision Tree Algorithms for EMG Signal Classification Using DWT. *Biomedical Signal Processing and Control*, **18**, 138–144.

Gupta, K. and Gupta, D. (2016). An Analysis on LPC, RASTA and MFCC Techniques in Automatic Speech Recognition System. In *2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)*, pages 493–497.

Hart, J. T., Collier, R., and Cohen, A. (1990). *A theory of intonation*, page 68120. Cambridge Studies in Speech Science and Communication. Cambridge University Press.

Heigold, G., Moreno, I., Bengio, S., and Shazeer, N. (2016). End-to-end Text-Dependent Speaker Verification. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 5115–5119. IEEE.

Hiremath, M. and Hiremath, P. S. (2017). 3D Face Recognition Based on Symbolic FDA Using SVM Classifier with Similarity and Dissimilarity Distance Measure. *International Journal of Pattern Recognition and Artificial Intelligence*, **31**(04), 1756006.

Hyassat, H. and Zitar, R. A. (2006). Arabic Speech Recognition Using SPHINX Engine. *International Journal of Speech Technology*, **9**(3-4), 133–150.

Jones, P., Woodgate, S., Doheny, E., Biggs, P., Williams, D., and Holt, C. (2019). Do Changes in Feature Selection Parameters Influence the Classification of Knee Rehabilitation Exercises When Using Body Worn Accelerometer Data? *Osteoarthritis and Cartilage*, **27**, S400.

Kak, S. C. (1995). On quantum neural computing. *Inf. Sci.*, **83**, 143–160.

Kandagal, A. and Udayashankara, V. (2017). Speaker Independent Speech Recognition Using Maximum Likelihood Approach for Isolated Words. *International Journal of Computer Application*, **7**, 72–83.

Kanisha, B., Lokesh, S., Kumar, P. M., Parthasarathy, P., and Chandra Babu, G. (2018). Speech Recognition with Improved Support Vector Machine Using Dual Classifiers and Cross Fitness Validation. *Personal and ubiquitous computing*, **22**(5-6), 1083–1091.

Kapadia, S., Valtchev, V., and Young, S. J. (1993). MMI Training for Continuous Phoneme Recognition on the TIMIT Database. In *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 491–494.

Karamat, N. (2001). Phonemic Inventory of Punjabi.

Karpagavalli, S. and Chandra, E. (2015). Phoneme and Word Based Model for Tamil Speech Recognition Using GMM-HMM. In *2015 International Conference on Advanced Computing and Communication Systems*, pages 1–5.

katspaugh (2012). Wavesurfer Web Address. *[Available Online]:https://wavesurfer-js.org/*. (Accessed 15 July 2012).

Keerthi, S. S. (2002). Efficient Tuning of SVM Hyperparameters Using Radius/Margin Bound and Iterative Algorithms. *IEEE Transactions on Neural Networks*, **13**(5), 1225–1229.

Kennedy, J. and Eberhart, R. (1995). Particle Swarm Optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948.

Khan, J. I. (1998). Dynamic sub-pattern matching with holographic associative memory. In *27th Applied Imagery Pattern Recognition (AIPR), Advances in Computer Assisted Recognition, SPIE*, pages 174–185.

Kim, M., Kim, Y., Yoo, J., Wang, J., and Kim, H. (2017). Regularized Speaker Adaptation of KL-HMM for Dysarthric Speech Recognition. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, **25**(9), 1581–1591.

Koshiba, Y. and Abe, S. (2003). Comparison of L1 and L2 Support Vector Machines. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, volume 3, pages 2054–2059.

Kumar (2010). Comparison of HMM and DTW for Isolated Word Recognition System of Punjabi Language. In I. Bloch and R. M. Cesar, editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 244–252, Berlin, Heidelberg. Springer Berlin Heidelberg.

Kumar, K. and Aggarwal, R. (2011). Hindi Speech Recognition System Using HTK. *International Journal of Computing and Business Research*, **2**(2), 2229–6166.

Kumar, K., Aggarwal, R., and Jain, A. (2012). A Hindi Speech Recognition System for Connected Words Using HTK. *International Journal of Computational Systems Engineering*, **1**, 25 – 32.

Kumar, R. and Singh, M. (2011). Spoken Isolated Word Recognition of Punjabi Language Using Dynamic Time Warp Technique. In C. Singh, G. Singh Lehal, J. Sengupta, D. V. Sharma, and V. Goyal, editors, *Information Systems for Indian Languages*, pages 301–301, Berlin, Heidelberg. Springer Berlin Heidelberg.

Kumar, S. B. S., Rao, K. S., and Pati, D. (2013). Phonetic and Prosodically Rich Transcribed speech corpus in Indian languages: Bengali and Odia. In *2013 International Conference Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*, pages 1–5.

Ladefoged, P. (1982). *A Course in Phonetics*. Harcourt Brace Jovanovich.

Lamel, L. F. and Gauvain, J. . (1992). Experiments on Speaker-Independent Phone Recognition Using BREF. In *Proceedings of ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 557–560.

Lata, S., Verma, P., and Arora, S. (2015). Identification of Prosodic Features of Punjabi for Enhancing the Pronunciation Lexicon Specification (Pls) for Voice Browsing. *International Journal on Natural Language Computing*, **4**, 61–77.

Latif, S., Rana, R., Khalifa, S., Jurdak, R., Qadir, J., and Schuller, B. (2020). Deep representation learning in speech processing: Challenges, recent advances, and future trends.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, **521**, 436–44.

Lee, K.-F. and Hon, H.-W. (1989). Speaker-Independent Phone Recognition Using Hidden Markov Models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **37**(11), 1641–1648.

Lee, K.-F., Hon, H.-W., Hwang, M.-Y., Mahajan, S., and Reddy, R. (1989). The SPHINX Speech Recognition System. In *1989 International Conference on Acoustics, Speech, and Signal Processing (ICASSP '89)*, pages 445 – 448.

Lee, K. F., Hon, H. W., and Reddy, R. (1990). An Overview of the SPHINX Speech Recognition System. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **38**(1), 35–45.

Leggetter, C. J. and Woodland, P. C. (1995). Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models. *Computer speech & language*, **9**(2), 171–185.

Li, L., Zhao, Y., Jiang, D., Zhang, Y., Wang, F., Gonzalez, I., Valentin, E., and Sahli, H. (2013). Hybrid Deep Neural Network-Hidden Markov Model (DNN-HMM) Based Speech Emotion Recognition. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pages 312–317.

Lippmann, R. P. (1997). Speech Recognition by Machines and Humans. *Speech communication*, **22**(1), 1–15.

Liu, Z., Wu, Z., Li, T., Li, J., and Shen, C. (2018a). GMM and CNN Hybrid Method for Short Utterance Speaker Recognition. *IEEE Transactions on Industrial Informatics*, **14(7)**, 3244–3252.

Liu, Z.-T., Wu, M., Cao, W.-H., Mao, J.-W., Xu, J.-P., and Tan, G.-Z. (2018b). Speech Emotion Recognition Based on Feature Selection and Extreme Learning Machine Decision Tree. *Neurocomputing*, **273**, 271–280.

Malde, K. D., Vachhani, B. B., Madhavi, M. C., Chhayani, N. H., and Patil, H. A. (2013). Development of Speech Corpora in Gujarati and Marathi for Phonetic Transcription. In *2013 International Conference Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*, pages 1–6.

Manjunath, K. E. and Rao, K. S. (2014). Automatic Phonetic Transcription for Read, Extempore and Conversation Speech for an Indian Language: Bengali. In *2014 Twentieth National Conference on Communications (NCC)*, pages 1–6.

Manjunath, K. E., Rao, K. S., and Pati, D. (2013). Development of Phonetic Engine for Indian Languages: Bengali and Oriya. In *2013 International Conference Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*, pages 1–6.

Mankala, S. R., Bojja, S. R., Ramaiah, V. S., and Rao, R. R. (2014). Automatic Speech Processing Using HTK for Telugu Language. *International Journal of Advances in Engineering & Technology*, **6**(6), 2572.

Mari, J. ., Fohr, D., and Junqua, J. . (1996). A Second-order HMM for High Performance Word and Phoneme-based Continuous Speech Recognition. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 1, pages 435–438.

Martinez, J., Perez, H., Escamilla, E., and Suzuki, M. M. (2012). Speaker Recognition Using Mel Frequency Cepstral Coefficients (MFCC) and Vector Quantization (VQ) Techniques. In *CONIELECOMP 2012, 22nd International Conference on Electrical Communications and Computers*, pages 248–251.

Mary, L. (2011). *Extraction and Representation of Prosody for Speaker, Speech and Language Recognition*. Springer Publishing Company, Incorporated.

Mary, L. (2018). *Extraction of Prosody for Automatic Speaker, Language, Emotion and Speech Recognition*. Springer.

Mary, L., Antony, A. P., Babu, B. P., and Prasanna, S. R. (2018). Automatic Syllabification of Speech Signal Using Short Time Energy and Vowel Onset Points. *International Journal of Speech Technology*, **21**(3), 571–579.

McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R., and Neven, H. (2018). Barren Plateaus in Quantum Neural Network Training Landscapes. *Nature communications*, **9**(1), 4812.

McGill, R., Tukey, J. W., and Larsen, W. A. (1978). Variations of box plots. *The American Statistician*, **32**(1), 12–16.

Memon, S., Jokhio, I. A., Arisar, S. H., Lech, M., and Maddage, N. (2012). Delta-MFCC Features and Information Theoretic Expectation Maximization Based Text-Independent Speaker Verification System. *IETE Journal of Research*, **58**(1), 5–12.

Mistry, P., Neagu, D., Trundle, P. R., and Vessey, J. D. (2016). Using Random Forest and Decision Tree Models for a new Vehicle Prediction Approach in Computational Toxicology. *Soft Computing*, **20**(8), 2967–2979.

Mu, R. and Zeng, X. (2019). A review of deep learning research. *KSII Transactions on Internet and Information Systems*, **13**, 1738–1764.

Muda, L., Begam, M., and Elamvazuthi, I. (2010). Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques. *CoRR*, **abs/1003.4083**.

Nagarajan, T., Murthy, H., and Rajesh, m. (2003). Segmentation of Speech into Syllable-like Units. In *EUROSPEECH-2003*, pages 2894–2896.

Nandyala, S. P. and Kumar, T. K. (2014). Hybrid HMM/DTW Based Speech Recognition with Kernel Adaptive Filtering Method. *International Journal on Computational Sciences and Applications*, **4(1)**, 11–21.

Nath, S., Sing, J. K., and Sarkar, S. K. (2018). Performance Comparison of PSO and Its New Variants in the Context of VLSI Global Routing. In *Particle Swarm Optimization with Applications*, chapter 5, page 61. BoD–Books on Demand.

Nooteboom, S. (1997). The Prosody of Speech: Melody and Rhythm. In *The Handbook of Phonetic Sciences, Nr. 5 in Blackwell Handbooks in Linguistics, chap*, pages 640–673.

Novoa, J., Wuth, J., Escudero, J. P., Fredes, J., Mahu, R., and Yoma, N. B. (2018). DNN-HMM Based Automatic Speech Recognition for HRI Scenarios. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '18, pages 150–159, New York, NY, USA. ACM.

Nwe, T. L., Foo, S. W., and De Silva, L. C. (2003). Speech Emotion Recognition Using Hidden Markov Models. *Speech communication*, **41**(4), 603–623.

Olatomiwa, L., Mekhilef, S., Shamshirband, S., Mohammadi, K., Petković, D., and Sudheer, C. (2015). A Support Vector Machine–Firefly Algorithm-Based Model for Global Solar Radiation Prediction. *Solar Energy*, **115**, 632–644.

Panchal, G., Ganatra, A., Shah, P., and Panchal, D. (2011). Determination of Over-Learning and Over-Fitting Problem in Back Propagation Neural Network. *International Journal on Soft Computing*, **2**(2), 40–51.

Parmar, H. and Hindoliya, D. (2011). Artificial Neural Network Based Modelling of Desiccant Wheel. *Energy and Buildings*, **43**(12), 3505 – 3513.

Passy, P. (2005). IPA Web Address. *http://www.langsci.ucl.ac.uk/ipa/fullchart.html*. (Accessed 23 July 2012).

Patil, H. A., Madhavi, M. C., Malde, K. D., and Vachhani, B. B. (2012). Phonetic Transcription of Fricatives and Plosives for Gujarati and Marathi Languages. In *2012 International Conference on Asian Language Processing*, pages 177–180.

Phan, F., Micheli-Tzanakou, E., and Sideman, S. (2000). Speaker Identification Using Neural Networks and Wavelets. *IEEE Engineering in Medicine and Biology Magazine*, **19**(1), 92–101.

Pruthi, T., Saksena, S., and Das, P. K. (2002). Swaranjali: Isolated Word Recognition for Hindi Language using VQ and HMM.

Rabiner, L. and Schmidt, C. (1980). Application of Dynamic Time Warping to Connected Digit Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **28**(4), 377–388.

Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, **77**(2), 257–286.

Rabiner, L. R., Wilpon, J. G., and Soong, F. K. (1989). High Performance Connected Digit Recognition Using Hidden Markov Models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **37**(8), 1214–1225.

Rafiq, M., Bugmann, G., and Easterbrook, D. (2001). Neural Network Design for Engineering Applications. *Computers & Structures*, **79**(17), 1541–1552.

Ravinder, K. (2010). Comparison of HMM and DTW for Isolated Word Recognition System of Punjabi Language. In I. Bloch and R. M. Cesar, editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 244–252, Berlin, Heidelberg. Springer Berlin Heidelberg.

Richardson, F., Reynolds, D., and Dehak, N. (2015). Deep Neural Network Approaches to Speaker and Language Recognition. *IEEE Signal Processing Letters*, **22**(10), 1671–1675.

Rohart, F., Gautier, B., Singh, A., and Lê Cao, K.-A. (2017). MixOmics: An R Package for Omics Feature Selection and Multiple Data Integration. *PLoS computational biology*, **13**(11), e1005752.

Saeed, K. and Nammous, M. K. (2007). A speech-and-speaker identification system: Feature extraction, description, and classification of speech-signal image. *IEEE Transactions on Industrial Electronics*, **54**(2), 887–897.

Sahoo, S. K., Choubisa, T., and Prasanna, S. R. M. (2012). Multimodal Biometric Person Authentication : A Review. *IETE Technical Review*, **29**(1), 54–75.

Saini, P., Kaur, P., and Dua, M. (2013). Hindi Automatic Speech Recognition Using HTK. *International Journal of Engineering Trends and Technology (IJETT)*, **4**(6), 2223–2229.

Saleem, N. and Khattak, M. I. (2018). Regularized Sparse Decomposition Model for Speech Enhancement via Convex Distortion Measure. *Modern Physics Letters B*, **32(32)**, 1850262.

Sandipan Mandal, Biswajit Das, and Pabitra Mitra (2010). Shruti-II: A Vernacular Speech Recognition System in Bengali and an Application for Visually Impaired Community. In *2010 IEEE Students Technology Symposium (TechSym)*, pages 229–233.

Sarma, B., Sarma, M., and R M Prasanna, S. (2014a). Semi-Automatic Segmentation and Marking of Pitch Contours for Prosodic Analysis. *Lecture Notes in Electrical Engineering*, **347**, 127–137.

Sarma, B., Sarma, M., and R M Prasanna, S. (2014b). Semi-Automatic Syllable Labelling for Assamese Language Using HMM and Vowel Onset-Offset Points. *Lecture Notes in Electrical Engineering*, **347**, 139–147.

Sarma, B. D., Sarma, M., Sarma, M., and Mahadeva Prasanna, S. R. (2013). Development of Assamese Phonetic Engine: Some Issues. In *2013 Annual IEEE India Conference (INDICON)*, pages 1–6.

Sarma, H., Saharia, N., and Sharma, U. (2014c). Development of Assamese Speech Corpus and Automatic Transcription Using HTK. In S. M. Thampi, A. Gelbukh, and J. Mukhopadhyay, editors, *Advances in Signal Processing and Intelligent Recognition Systems*, pages 119–132, Cham. Springer International Publishing.

Sarma, H., Saharia, N., and Sharma, U. (2017). Development and Analysis of Speech Recognition Systems for Assamese Language Using HTK. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, **17**(1), 7.

Satori, H., Harti, M., and Chenfour, N. (2007). Introduction to Arabic Speech Recognition Using CMUSphinx System. *CoRR*, **abs/0704.2083**.

Satori, H., Hiyassat, H., Haiti, M., and Chenfour, N. (2009). Investigation Arabic Speech Recognition Using CMU Sphinx System. *International Arab Journal of Information Technology (IAJIT)*, **6**(2), 186–190.

Sha, F. and Saul, L. K. (2007). Comparison of Large Margin Training to Other Discriminative Methods for Phonetic Recognition by Hidden Markov Models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–313. IEEE.

Shaikhina, T., Lowe, D., Daga, S., Briggs, D., Higgins, R., and Khovanova, N. (2017). Decision Tree and Random Forest Models for Outcome Prediction in Antibody Incompatible Kidney Transplantation. *Biomedical Signal Processing and Control*, **52**, 456–462.

Sharma, P. and Rajpoot, Kiran, A. (2013). Automatic Identification of Silence, Unvoiced and Voiced Chunks in Speech. *Computer Science and Information Technology*, **3**, 87–96.

Sheh, A. and Ellis, D. P. W. (2003). Chord Segmentation and Recognition Using EM-Trained Hidden Markov Models. In *ISMIR*.

Shen, C.-P., Lin, J.-W., Lin, F.-S., Lam, A. Y.-Y., Chen, W., Zhou, W., Sung, H.-Y., Kao, Y.-H., Chiu, M.-J., Leu, F.-Y., *et al.* (2017). GA-SVM Modeling of Multiclass Seizure Detector in Epilepsy Analysis System Using Cloud Computing. *Soft Computing*, **21**(8), 2139–2149.

Shet Shirodkar, N. (2016). *Speech to Text Recognition Using Hidden Markov Model Toolkit*. Ph.D. thesis, GOA University.

Shi, W., Zhang, X., Zou, X., and Han, W. (2017). Deep Neural Network and Noise Classification-Based Speech Enhancement. *Modern Physics Letters B*, **31**(19-21), 1740096.

Shridhara, M. V., Banahatti, B. K., Narthan, L., Karjigi, V., and Kumaraswamy, R. (2013). Development of Kannada Speech Corpus for Prosodically Guided Phonetic Search Engine. In *2013 International Conference Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*, pages 1–6.

Singh, P. and Lehal, G. (2011). Corpus Based Statistical Analysis of Punjabi Syllables for Preparation of Punjabi Speech Database. *International Journal of Intelligent Computing Research*, **2**, 124–128.

Siniscalchi, S. M., Li, J., and Lee, C. (2013). Hermitian Polynomial for Speaker Adaptation of Connectionist Speech Recognition Systems. *IEEE Transactions on Audio, Speech, and Language Processing*, **21**(10), 2152–2161.

Snyder, D., Ghahremani, P., Povey, D., Garcia-Romero, D., Carmiel, Y., and Khudanpur, S. (2016). Deep Neural Network-Based Speaker Embeddings for End-to-end Speaker Verification. In *Spoken Language Technology Workshop (SLT), 2016 IEEE*, pages 165–170. IEEE.

Solomonoff, A., Campbell, W. M., and Boardman, I. (2005). Advances in Channel Compensation for SVM Speaker Recognition. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, volume 1, pages I–629. IEEE.

Sreejith, A., Mary, L., K. S., R., Joseph, A., and Augustine, A. (2013). Automatic Prosodic Labeling and Broad Class Phonetic Engine for Malayalam. In *International Conference on Control Communication and Computing, ICCC 2013*, pages 522–526.

Ssarma, M. K., Gajurel, A., Pokhrel, A., and Joshi, B. (2017). HMM Based Isolated Word Nepali Speech Recognition. In *2017 International Conference on Machine Learning and Cybernetics (ICMLC)*, volume 1, pages 71–76.

Sukawattanavijit, C., Chen, J., and Zhang, H. (2017). GA-SVM algorithm for Improving Land-Cover Classification Using SAR and Optical Remote Sensing Data. *IEEE Geoscience and Remote Sensing Letters*, **14**(3), 284–288.

Sunil Kumar, S. and Sreenivasa Rao, K. (2016). Voice/non-voice detection using phase of zero frequency filtered speech signal. *Speech Communication*, **81**, 90–103. Phase-Aware Signal Processing in Speech Communication.

Suryawanshi, U. J. and Ganorkar, S. (2014). Hardware Implementation of Speech Recognition Using MFCC and Euclidean Distance. *International journal of advanced research in electrical, electronics and instrumentation engineering*, **3(8)**, 11248–11254.

Sztahó, D., Szaszák, G., and Beke, A. (2019). Deep learning methods in speaker recognition: a review.

Tailor, J. H. and Shah, D. B. (2016). Speech Recognition System Architecture for Gujarati Language. *International Journal of Computer Applications*, **138(12)**, 28–31.

Thalengala, A., Shama, K., and Mangalore, M. (2018). Performance Analysis of Isolated Speech Recognition System using Kannada Speech Database. *Pertanika Journal of Science and Technology*, **26**(4), 1849–1866.

Thennattil, J. and Mary, L. (2014). Implementation of Automatic Segmentation of Speech Signal for Phonetic Engine in Malayalam. *International Journal of Engineering and Technical Research*, **2**, 295–298.

Thennattil, J. J. and Mary, L. (2016). Phonetic Engine for Continuous Speech in Malayalam. *IETE Journal of Research*, **62**(5), 679–685.

Tirumala, S. S. and Shahamiri, S. R. (2016). A review on deep learning approaches in speaker identification. In *ICSPS 2016*.

Tiwari, V. (2010). MFCC and its Applications in Speaker Recognition. *International journal on emerging technologies*, **1**(1), 19–22.

Trang, H., Tran Hoang Loc, and Huynh Bui Hoang Nam (2014). Proposed Combination of PCA and MFCC Feature Extraction in Speech Recognition System. In *International Conference on Advanced Technologies for Communications (ATC 2014)*, pages 697–702.

Trentin, E. and Gori, M. (2003). Robust Combination of Neural Networks and Hidden Markov Models for Speech Recognition. *IEEE Transactions on Neural Networks*, **14**(6), 1519–1531.

Tripathy, S., Baranwal, N., and Nandi, G. C. (2013). A MFCC Based Hindi Speech Recognition Technique Using HTK Toolkit. In *2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013)*, pages 539–544.

Tu, S. (2015). Derivation of Baum-Welch Algorithm for Hidden Markov Models.

Tuar, T., Gantar, K., Koblar, V., enko, B., and Filipi, B. (2017). A study of overfitting in optimization of a manufacturing quality control procedure. *Applied Soft Computing*, **59**, 77–87.

Ullah, I., Raza, B., Malik, A. K., Imran, M., Islam, S. U., and Kim, S. W. (2019). A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector. *IEEE Access*, **7**, 60134–60149.

Vachhani, B. B. and Patil, H. A. (2013). Use of PLP Cepstral Features for Phonetic Segmentation. In *2013 International Conference on Asian Language Processing*, pages 143–146.

Wang, J.-C., Chin, Y.-H., Hsieh, W.-C., Lin, C.-H., Chen, Y.-R., and Siahaan, E. (2015). Speaker Identification with Whispered Speech for the Access Control System. *IEEE Transactions on Automation Science and Engineering*, **12**(4), 1191–1199.

Wang, Y., Du, J., Dai, L., and Lee, C. (2017). A Gender Mixture Detection Approach to Unsupervised Single-Channel Speech Separation Based on Deep Neural Networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, **25**(7), 1535–1546.

Woodland, P. C. and Povey, D. (2002). Large Scale Discriminative Training of Hidden Markov Models for Speech Recognition. *Computer Speech & Language*, **16**(1), 25–47.

Woodland, P. C., Leggetter, C. J., Odell, J. J., Valtchev, V., and Young, S. J. (1995). The 1994 HTK Large Vocabulary Speech Recognition System. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 73–76.

Xi, J., Liang, R., and Fei, X. (2017). An Algorithm of Improving Speech Emotional Perception for Hearing Aid. *Modern Physics Letters B*, **31**(19-21), 1740094.

Xu, L., Yang, Z., and Sun, L. (2016). Simplification of I-Vector Extraction for Speaker Identification. *Chinese Journal of Electronics*, **25**(6), 1121–1126.

Xue, S., Abdel-Hamid, O., Jiang, H., Dai, L., and Liu, Q. (2014). Fast Adaptation of Deep Neural Network Based on Discriminant Codes for Speech Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, **22**(12), 1713–1725.

You, C. H., Lee, K. A., and Li, H. (2009). An SVM Kernel with GMM-supervector Based on the Bhattacharyya Distance for Speaker Recognition. *IEEE Signal processing letters*, **16**(1), 49–52.

Young, S. (1999). Acoustic Modelling for Large Vocabulary Continuous Speech Recognition. In *Computational Models of Speech Pattern Processing*, pages 18–39. Springer.

Young, S. (2001). The HTK Book (for HTK version 3.1). *http://htk. eng. cam. ac. uk*.

Young, S. and Young, S. (1994). The HTK Hidden Markov Model Toolkit: Design and Philosophy. *Entropic Cambridge Research Laboratory, Ltd*, **2**, 2–44.

Yu, D. and Li, J. (2017). Recent progresses in deep learning based acoustic models.

Zhang, M. (2017). Application of BP Neural Network in Acoustic Wave Measurement System. *Modern Physics Letters B*, **31**(19-21), 1740052.

Zhang, T. and Chen, W. (2016). LMD Based Features for the Automatic Seizure Detection of EEG Signals Using SVM. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, **25**(8), 1100–1108.