

On-device Mobile Speech Recognition

Mohammed Kyari Mustafa

A thesis submitted in partial fulfilment of the
requirements of Nottingham Trent University for the
degree of Doctor of Philosophy

May 2016

Acknowledgement

My profound gratitude to Allah (S.W.T) for giving me life, to see this project through from the beginning to the end, and all blessings bestowed on me throughout my life.

My deepest gratitude to the Petroleum Technology Development Fund (PTDF), for their full sponsorship of this research right from the beginning to the end.

I will like to acknowledge my Director of studies, Dr. Tony Allen. For no amount of words I use can quantify the amount of support he has given me throughout my studies. I enjoyed full unconditional support from him. It is also worthy of note, that contrary to the general norm of students aligning to their supervisors. He was magnanimous enough to adjust to me. I will also like to acknowledge my second supervisor, Dr. Kofi Appiah, who despite the stage he joined the project, got involved and offered so much support. He was always available at any given time I needed help and for that I am exceptionally grateful. I do acknowledge the support of my independent assessor Dr. Demosthenes Koutsogeorgis, who always kept the balance throughout my study.

I will love to extend gratitude to my parents. Dr. Shettima Mustafa and Hajiya Aishatu Mustafa. They have showered me with love, support in every aspect of this project and my life. For this I am eternally grateful. My heartfelt gratitude to my brothers and sisters for putting up with me.

I will like to thank my colleagues and friends at Nottingham Trent University for their support in the last four years. I am very grateful. Two are worthy of mention, because their support was overwhelming (Dr. Tariq Tashan and Dr. Mahmud Shertil).

Mohammed Kyari Mustafa

October 2015

Abstract

Despite many years of research, Speech Recognition remains an active area of research in Artificial Intelligence. Currently, the most common commercial application of this technology on mobile devices uses a wireless client – server approach to meet the computational and memory demands of the speech recognition process. Unfortunately, such an approach is unlikely to remain viable when fully applied over the approximately 7.22 Billion mobile phones currently in circulation. In this thesis we present an On – Device Speech recognition system. Such a system has the potential to completely eliminate the wireless client-server bottleneck.

For the Voice Activity Detection part of this work, this thesis presents two novel algorithms used to detect speech activity within an audio signal. The first algorithm is based on the Log Linear Predictive Cepstral Coefficients Residual signal. These LLPCCRS feature vectors were then classified into voice signal and non-voice signal segments using a modified K-means clustering algorithm. This VAD algorithm is shown to provide a better performance as compared to a conventional energy frame analysis based approach. The second algorithm developed is based on the Linear Predictive Cepstral Coefficients. This algorithm uses the frames within the speech signal with the minimum and maximum standard deviation, as candidates for a linear cross correlation against the rest of the frames within the audio signal. The cross correlated frames are then classified using the same modified K-means clustering algorithm. The resulting output provides a cluster for Speech frames and another cluster for Non – speech frames. This novel application of the linear cross correlation technique to linear predictive cepstral coefficients feature vectors provides a fast computation method for use on the mobile platform; as shown by the results presented in this thesis.

The Speech recognition part of this thesis presents two novel Neural Network approaches to mobile Speech recognition. Firstly, a recurrent neural networks architecture is developed to accommodate the output of the VAD stage. Specifically, an Echo State Network (ESN) is used for phoneme level recognition. The drawbacks and advantages of this method are explained further within the thesis. Secondly, a dynamic Multi-Layer Perceptron approach is developed. This builds on the drawbacks of the ESN and provides a dynamic way of handling speech signal length variabilities within its architecture. This novel Dynamic Multi-Layer Perceptron uses both the Linear Predictive Cepstral Coefficients (LPC) and the Mel Frequency Cepstral Coefficients (MFCC) as input features. A speaker dependent approach is presented using the Centre for spoken Language and Understanding (CSLU) database. The results show a very distinct behaviour from conventional speech recognition approaches because the LPC shows performance figures very close to the MFCC. A speaker independent system, using the standard TIMIT dataset, is then implemented on the dynamic MLP for further confirmation of this. In this mode of operation the MFCC outperforms the LPC. Finally, all the results, with emphasis on the computation time of both these novel neural network approaches are compared directly to a conventional hidden Markov model on the CSLU and TIMIT standard datasets.

Copyright Statement

This work is the intellectual property of the author. You may copy up to 5% of this work for private study, or personal, non-commercial research. Any re-use of the information contained within this document should be fully referenced, quoting the author, title, university, degree level and pagination. Queries or requests for any other use, or if a more substantial copy is required, should be directed in the first instance to the owner(s) of the Intellectual Property Rights.

Table of Content

Acknowledgement	i
Abstract	ii
Copyright Statement	iv
Table of Content	v
List of Figures	viii
List of Tables	x
List of Abbreviations	xii
List of Publications	xiii
Chapter One - Introduction	1
1. Introduction.....	1
1.1 Speech Recognition Process	3
1.2 Speech Recognition On Mobile Platforms	4
1.2.1 The Client-Server Approach Challenges	5
1.2.2 On-Device Implementation Challenges	5
1.2.3 Overcoming Challenges.....	6
1.3 Research Motivation	6
1.4 Research Questions.....	7
1.5 Research Aim and Objectives.....	7
1.6 Research Methodology	9
1.7 Framework	10
1.8 Thesis Outline	10
Chapter Two – Background Information and Experimental Framework	12
2. Introduction.....	12
2.1 Speech Signal.....	13
2.1.1 Speech Features	14
2.1.2 Time Domain Features.....	14
2.1.3 Spectral Domain Features	15
2.2 Phoneme and Word Segmentation Techniques	18
2.2.1 Self-Organising Maps	18
2.2.2 K-means	20
2.3 Phoneme and Word Classification Techniques	21

2.3.1 Multi-Layer Perceptron (Feed-Forward Architecture)	22
2.3.2 Echo State Networks	26
2.3.3 Hidden Markov Model.....	27
2.4 Speech Database	29
2.5 Mobile Devices	31
2.6 Summary.....	32
Chapter Three – Literature Review	33
3. Introduction.....	33
3.1 Voice Activity Detection	33
3.2 Speech Recognizer.....	43
3.2.1 Feed Forward Multi-Layer Perceptron in Speech Recognition	43
3.2.2 Echo State Network Applications to Speech Recognition.....	48
3.2.3 Hidden Markov Model Speech Recognition.....	50
3.2.4 Pocket-Sphinx On-Device Speech Recognition Systems	51
3.3 Summary.....	52
Chapter Four – Mobile Voice Activity Detection.....	54
4. Introduction.....	54
4.1 Why Voice Activity Detection?.....	55
4.2 Other Voice Activity Detection Experiments	56
4.2.1 Speech Signal Pre-Processing.....	57
4.2.2 Short Time Energy Voice Activity Detection Experiments	58
4.2.3 Linear Predictive Coding Residual (Prediction Error) Voice Activity Detection Experiments.....	61
4.2.4 Raw Short Time Energy Voice Activity Detection versus Raw Prediction Error Voice Activity Detection Results	62
4.2.5 Linear Cross Correlation Voice Activity Detection Experiments	64
4.2.6 Linear Cross Correlation Results	68
4.3 Novel Voice Activity Detection Experiments	71
4.3.1 First Novel Voice Activity Detection Algorithm.....	71
4.3.2 Prediction Error K-means Voice Activity Detection Algorithm Results.....	75
4.3.3 Second Novel Voice Activity Detection Algorithm	78

4.3.4 Linear Predictive Coding K-means on Standard Deviation Results	84
4.4 Summary	86
Chapter Five – Mobile Speech Recognition	90
5. Introduction	90
5.1 Speech Recognition	91
5.2 Databases	92
5.3 Application of Speech Signal to Speech Recognition	93
5.4 Echo State Recognizer	94
5.5 Echo State Network Experiments and Results	103
5.6 Proposed Novel Dynamic Multi-Layer Perceptron	108
5.7 Comparison Hidden Markov Model	116
5.8 Conclusion	118
5.9 Summary	121
Chapter Six – Conclusion and Future Work	122
6. Introduction	122
6.1 Choice of Linear Prediction Coefficients	123
6.2 Choice of Voice Activity Detection	123
6.3 K-means Voice Activity Detection Choice	124
6.4 Artificial Neural Network Classifier Choice	125
6.5 Contributions to Knowledge	126
6.6 Mobile Speech Recognition Framework	127
6.7 Recommendations	129
6.8 Future Work	130
6.8.1 Further Investigation of Speech Production Mechanism	130
6.8.2 Advancing Into Continuous Speech Recognition	131
6.8.3 Adoption of Dynamic Multi-Layer Perceptron	131
6.8.4 Further Work on Echo State Networks	131
6.8.5 Dynamic Multi-Layer Perceptron in Other Domains	132
References	133
Appendix 1 Other Voice Activity Detection based Experiment results	146
Appendix 2 Results of two Hidden Layer Dynamic Multi-Layer Perceptron	149

List of Figures

- Figure 1.1 Speech recognition process
- Figure 1.2 Adopted Framework
- Figure 2.1 An example of a (3 x 4) SOM structure
- Figure 2.2 An example of a 5x4x3 MLP with connections
- Figure 2.3 A standard logistic function
- Figure 2.4 An ESN network showing output weights update
- Figure 2.5 A three state left to right HMM
- Figure 4.1 Former part of framework
- Figure 4.2 Time domain speech signal containing the digits 1 - 9
- Figure 4.3 Periodicity plot
- Figure 4.4 Raw Powered STE plot of speech signal of 5.541 seconds
- Figure 4.5 Raw Power Prediction Error signal of 5.541 seconds speech signal
- Figure 4.6 Half side DFT Spectrum of 5.541 seconds speech signal
- Figure 4.7 64 order PE of 5.541 seconds speech signal
- Figure 4.8 DFT Cross Correlation Plot of speech signal of 5.541 seconds
- Figure 4.9 LPC Cross Correlation Plot of speech signal of 5.541 seconds
- Figure 4.10 PE Cross Correlation Plot of speech signal of 5.541 seconds
- Figure 4.11 LCC LPC using MSD plot
- Figure 4.12 Flipped figure 4.11

Figure 4.13	Framework Outcome of Chapter 4
Figure 5.1	Latter Part of Framework
Figure 5.2	A plot of digit 1 from CSLU
Figure 5.3	A plot of digit 1 from TIDIGITS
Figure 5.4	ESN Architecture
Figure 5.5	Dynamic MLP structure
Figure 5.6	Framework Outcome Chapter 5
Figure 6.1	Framework showing Computation Times
Figure A1	21 Order LPC Cross Correlation for speech signal with 44330 samples (346 frames)

List of Tables

Table 2.1	TIDIGITS Speaker details
Table 4.1	Manual borders of time domain signal in figure 1
Table 4.2	STE Algorithm Pseudo code
Table 4.3	Output evaluation table for Raw powered experiments
Table 4.4	Final results of STE Vs PE of 10 CSLU database speakers
Table 4.5	LCC algorithm computation Pseudo code
Table 4.6	LCC computation times in seconds on HTC mobile device
Table 4.7	Prediction Error Extraction
Table 4.8	K-means computation pseudo
Table 4.9	PE K-means Recognition Results for CSLU 160 digits (10 speakers)
Table 4.10	K-means PE vs. Raw PE vs. STE
Table 4.11	Computation Time of 44330 samples (5.54125 secs) speech signal
Table 4.12	Feature Extraction + Algorithm computation times
Table 4.13	Pseudo Implementation of LCC code
Table 4.14	K-means clustering implementation pseudo
Table 4.15	10 Initial Samples for parameter optimisation
Table 4.16	10 samples from the algorithm in section 4.3.1
Table 4.17	Linear Cross Correlation Computation Times

Table 4.18	Computation time for all VAD algorithms including feature extraction times in seconds
Table 4.19	Final Performance of all VAD algorithms
Table 5.1	Digit Phoneme labels
Table 5.2	Echo State Network Forward Pass
Table 5.3	Manually concatenated MFCC ESN phoneme (frame) recognition results for training data
Table 5.4	Automatic 17 output SOM used for ESN with MFCC and LPC on training data
Table 5.5	Pseudo code of a forward pass of the Dynamic MLP
Table 5.6	Results of the final set of experiments conducted using the Dynamic MLP
Table 5.7	Dynamic MLP best performances
Table 5.8	Hidden Markov Model results for MFCC word recognition
Table 5.9	Average Computation times of ESN, DMLP and HMM on Samsung Note 3
Table 5.10	Comparison between best performance Dynamic MLP and HMM
Table 6.1	Performance for CSLU for the Two features
Table 6.2	Comparison between best performance Dynamic MLP and HMM over 143.4 seconds of speech data
Table A1	Parameter modifications for Raw STE algorithm
Table A11	Independent samples test for second novel algorithm
Table A2	2 Hidden Layer results for Dynamic MLP

List of Abbreviations

ANN	Artificial Neural Networks
ASR	Automatic Speech recognition
CMU	Carnegie Melon University
CSLU	Center for Spoken Language and Understanding
DFT	Discrete Fourier Transform
DMLP	Dynamic Multi-Layer Perceptron
ED	Euclidean Distance
ESN	Echo State Networks
FFT	Fast Fourier Transform
GMM	Gaussian Mixture Models
HMM	Hidden Markov Models
LCC	Linear Cross Correlation
LPC	Linear Predictive Coding Cepstral Coefficients
MFCC	Mel Frequency Cepstral Coefficients
MLP	Multi-Layer Perceptron
MSD	Minimum Standard Deviation
NIST	National Institute of Standards and Technology
NN	Neural Networks
PC	Personal Computer
PE	Prediction Error
RAM	Read Access Memory
RNN	Recurrent Neural Network
SD	Standard Deviation
SOM	Kohonen Self-Organising Maps
STE	Short Time Energy
TIDIGITS	Texas Instruments Digits
TIMIT	Texas Instruments and Massachusetts Institute of Technology
VAD	Voice Activity Detection

List of Publications

Conference Publication

1. M. K. Mustafa, T. Allen, and L. Evett, A Review of Voice Activity Detection Techniques for On-Device Isolated Digit Recognition on Mobile Devices, in M. Bramer, M. Petridis (eds.), Research and Development in Intelligent Systems XXXI, Springer International Publishing, 2014.
2. M. K. Mustafa, T. Allen, and K. Appiah. A novel K-means Voice Activity Detection Algorithm using Linear Cross correlation on the standard deviation of Linear Predictive Coding, in M. Bramer, M. Petridis (eds.), Research and Development in Intelligent Systems XXXII, Springer International Publishing, 2015. (Best Application Paper).

Journal Publication (Invitation)

As a result of the conference submission of paper 2. An invitation was extended for a journal publication in “**German Journal on Artificial Intelligence - Organ des Fachbereichs "Künstliche Intelligenz" der Gesellschaft für Informatik e.V.**”. It is intended, that the novel speech recognition algorithm (Dynamic MLP) developed will be submitted to this journal.

Chapter One

Introduction

1. Introduction

Speech recognition is a major area of research in Artificial intelligence. This technology is very relevant in the field of control, where a specific voice command is issued to a device, which then performs the specified task. This is particularly appropriate for people who have physical difficulties with using conventional keyboard input and for all users of remote telephone-based information system services. Speech recognition involves an individual saying a particular set of words, and then the system automatically attempting to recognize the spoken utterances.

In recent years, with the advent of SIRI etc, the application of speech recognition in our everyday life has moved from being a fantasy to an everyday reality. Today, communication with these man-made devices seems normal, and is even moving from being just speech recognition to a two-way “conversation” between the machine and the speaker. Given the visions offered by recent movies such as Star trek, Iron man, Avengers etc.; this trend is only likely to continue with people eventually being able to speak to cars, robots and even smart homes.

Speech recognition can be classified into two parts, speaker dependent and speaker independent [1]. In a speaker dependent speech recognition system, the system is designed to recognize speech from a particular user. In a speaker independent recognition system, the system is designed to recognize speech from any user who uses the system. Both speaker dependent and independent systems are further divided into isolated word and continuous

speech recognition. By their nature, isolated word recognition systems tend to use small, bespoke vocabularies to achieve high recognition rates [2]. Continuous speech recognition systems, on the other hand, attempts to use larger, natural language level, vocabularies which carry with them a significantly reduced recognition rate.

The most recent and popular application of this technology is on smart phones, which have brought about the upsurge in the popularity of this technology. SIRI from Apple, Talk to Google from Google and, the most recent to join the competition, Cortana from Microsoft are the fore-front application of Speech recognition on mobile devices.

The advancements in speech recognition research and application has brought about the adaptation of speech recognition as a tool for a variety of applications such as topic identification [3], source cell-phone identification [4], industrial control using microcontrollers [5] and many other applications. However, the underlying principle of speech recognition has not changed irrespective of the task it is used for. The main principle behind speech recognition is based on a pattern matching framework, which takes spoken utterances and expresses them as sequences of stochastic patterns [6]. The pattern matching frameworks are based on different models and include probabilistic models such as the Gaussian Mixture Models(GMM)[7,8], Hidden Markov Models(HMM)[9,10,11,12] and statistical models such as Artificial Neural Networks (ANN)[6,13,14,15,16] and more recently, hybrid approaches that combine both probabilistic and statistical models[17].

1.1. Speech Recognition Process

The speech recognition process can be broken down into two stages;

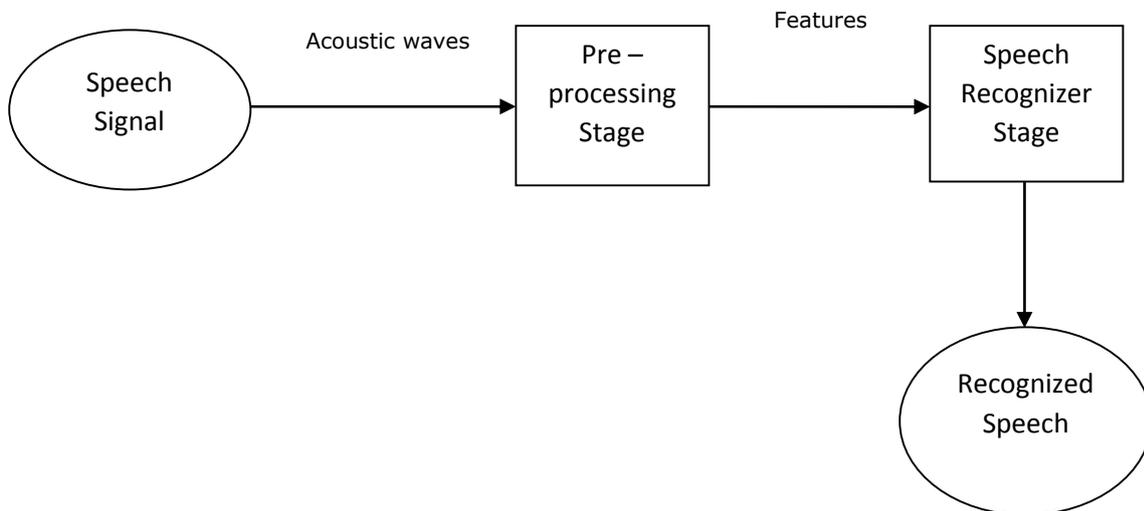


Figure 1.1. Speech recognition process

Figure 1.1 tries to capture the process of speech recognition.

- **Pre- Processing stage:** In all the approaches used for speech recognition, there is a significant amount of work done prior to the speech recognition stage because an efficient automatic speech segmentation pre-processor can reduce the computational load and power consumption of a target device [18]. At the same time an optimal choice of signal parameterisation can lead to a better recognition rate [19]. There are a variety of pre-processing techniques that can be applied to the speech signal. Both feature extraction and voice activity detection (VAD) can be considered as a pre-processing stage to speech recognition. In feature extraction, the recorded speech signal is processed to extract the relevant speech features. These features are then often passed to a VAD stage before being further passed on to the speech recognition stage. It is at

this pre-processing stage that the input format of the classification method is determined. In the speech recognition literature, there are two basic directions in speech recognition feature choices. These are the Discrete Fourier Transform (DFT) approach and the Linear Predictive Cepstral (LPC) approach [19]. Any processing done on the speech signal to extract features and modify them are predominantly by-products of one of these two mentioned approaches.

- **Speech Recognizer Stage:** The extracted speech features are then passed on to this stage for recognition. A classification model is used for the classification of the respective speech features. These features are used to train the recogniser in order to create a module that can recognise unknown speech samples. The common speech recognisers used are probabilistic models such as Gaussian mixture models [7] and hidden Markov models [9], and statistical models which are based on neural networks structures [6,14].

1.2. Speech Recognition on Mobile Platforms

In the studies of speech recognition, there is a huge trend towards developing features and algorithms that produce high performance figures rather than on the practicality of implementation or the computational resources they demand. Computational resources play a crucial role in mobile device based speech recognition systems; which can be classified into two basic approaches. The first is the client – server approach, which is the predominant approach of implementation. **SIRI**, **Google Talk** and **Cortana** are all based on the client – server implementation. The second implementation tries to implement the speech recognition system as a standalone system on the mobile device, the most common example of this being the **Pocketsphinx** by CMU [20]. However, there are disadvantages and limitations with regards to both implementations;

1.2.1. The Client – Server Approach Challenges

- The efficient assignment of Automatic Speech Recognition (ASR) decoders to computing resources in network based server deployment (the efficiency at peak loads, when a large number of ASR decoders are used in multi - user scenarios) [21].
- The implementation of acoustic adaptation and normalization algorithms within a client – server framework [21].
- The requirement for 24x7 connectivity to a server (requiring some form of internet connection) also determines the functionality of the speech recognition.
- Bandwidth of the connection to the server also determines how fast the system works.
- Energy consumption of wireless communications between client and server.

1.2.2. On - Device Implementation Challenges

- Memory on the mobile platform is limited as compared to computer platform. Though there have been recent improvements in the memory capacity for mobile devices, this is still insignificant compared to a modern computer platform.
- Computational speed on the mobile platform is also limited as compared to a computer based device. This hampers the processing time required to complete a speech recognition process.
- Power is limited on mobile devices. Speech recognition process requires a good deal of operation and as such this affects the memory and computational demand on the mobile platform, which in turn drain the power of the mobile device.

- The only currently available on-device speech recognition engine [20] is a restricted version of a research engine that was developed for PC based speech recognition purposes. As such it is not optimised for mobile platforms.

1.2.3. Overcoming challenges

As technology improves, it is expected that the identified on-device challenges will be easier to overcome than the equivalent client-server approach challenges. There is a well-established route for technological improvements in the processing power, memory capacity and power consumption of mobile devices. Thus, the research & development of an on-device speech recognition system that optimally addresses the challenges of current mobile devices would be a timely addition to this future technical development.

1.3. Research Motivation

The technological boom currently being experienced on the mobile platform is as substantial as that on the computer based level in the latter part of the last century. The number of mobile phone users has been rapidly rising since the early 2000s, with current figures at 7.22 billion and an annual prediction of 6.10% [22]. Along with the increase in the number of users, comes a higher demand on services that the mobile phones can offer. Speech recognition is one such service that can be applied to a wide range of applications. The current client-server speech recognition methods (SIRI, Google and Cortana) have known issues regarding scalability, bandwidth and connectivity etc. On-device speech recognition would help address these issues. Unfortunately, the only attempted implementation to date [20] is a downscaled version of a PC based approach.

The application of an on-device based speech recognition system is also economically viable for countries where internet services and/or server farms are not consistently available. The other unique economic advantage is that a system can be built into the mobile device

which can then be easily integrated to other devices for which a dual communication system over choice of connectivity is avoided (connections to two or more networks).

With mobile devices being so much a part of our daily lives, the potential of on-device speech recognition offers endless possibilities. An on-device based speech recognition system can be developed for the control of our daily activities. For example, a mobile based speech recognition system can be used to control different systems in the house of the end user. There is no limit to the amount of devices humans have in their houses. The prospect of being able to synchronise all these devices into a mobile based speech recognition system is appealing to say the least. This can save a lot of monitoring required for the different devices by integrating them on the mobile device. After integration, controlling these systems by simple voice commands to do whatever task required. The prospects of mobile devices use cannot be quantified but it will be a rewarding venture to delve into this booming technological field.

1.4. Research Questions

1. What are the optimum voice activity detection, feature extraction and speech recognition techniques that can be used to provide for On-device mobile speech recognition, bearing in mind mobile device constraints.
2. How do the overall on-device speech recognition performance figures compare to existing state-of-the-art client-server speech recognition systems.

1.5. Research Aim and Objectives

The Aim of this research is to develop an efficient and effective on-device implementation of isolated speech recognition that takes into account the constraints of the mobile platform. This aim can be achieved through these following objectives;

- **Feature Extraction:** The different speech features that can be extracted should be investigated with a view to their viability with regards to mobile devices implementation and how they affect the choice of the VAD techniques to be developed for the speech recognition process. Both Mel frequency cepstral coefficients (MFCC) which are the predominant choice of speech features [4,17,23] and the LPC features are to be tested in this regard to make an informed choice between the two feature sets.
- **Voice activity detection (VAD):** Existing VAD algorithms should be explored with regards to their viabilities for mobile based speech recognition. If need be, then develop novel VAD algorithms that can cater for the inadequacies of the mobile platform in terms of processing power and memory. In addition, the need to accommodate speech pause detection when recording audio signals should be considered. This is to try and conserve power consumption on the mobile device because extra computation has an effect on the power consumption of mobile devices. The VAD algorithm should be fast and efficient enough to compliment the mobile speech recognition process.
- **Speech recognition:** The different methods of speech recognition should be explored with a view to accommodating the outcome of the VAD stage. Neural network based word and phoneme recognition approaches would be explored to produce an effective and fast speech recogniser that is able to accommodate the segments of the speech signal that is adjudged to have speech from the VAD stage. The speech recogniser to be developed also has to accommodate the speech signal length variability. This is a common characteristic of speech because different words have different utterance times. This is in addition to the speech recogniser developed being fast enough to work in a real-time use of speech recognition. A comparison with a baseline hidden

Markov model system should also be attempted to provide a comparative view of the results obtained.

1.6. Research Methodology

To be able to achieve the outlined aim of this research, two important components have to be considered. These components are the mobile devices that are to be used as the target implementations, as well as the speech databases to be used.

The mobile devices to be used for this research have to be mobile devices that are currently available in the market and which give an averaged range of the current mobile device computational, memory and power. These devices should not be modified to fit into the speech recognition process. On the contrary, the speech recognition process to be developed needs to conform to the mobile devices. The particular mobile devices selected and used for this research are identified and details are given in chapter 2 section 2.5.

The speech databases to be used must be in a format that can be easily processed on the mobile devices without any modification. The sampling rates of the audio files must equally be in a format easily transmittable over mobile telecommunications channels. The databases to be used should also be readily available databases within the literature. This gives room for direct comparison to other techniques available within the literature. In addition, the databases to be adopted have to cover a good range of speakers. These speakers must include male, female and children (boys and girls) in order to cover the gender of speakers. These speakers must also be collected over a wider geographical area in order to be cover a good range of speaking accents associated with different speakers. The full details of the particular databases used for this research are given in chapter 2 section 2.4.

1.7. Framework

A framework was developed and adopted for the work undertaken in this thesis. This framework is designed to provide efficient and effective speech recognition on the mobile device. The work done in this research falls into different subsections of this framework.

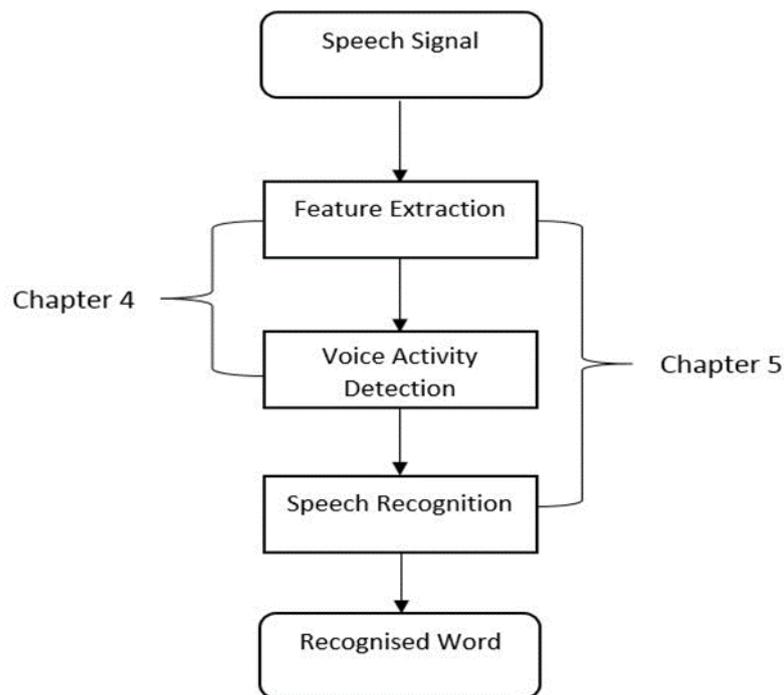


Figure 1.2 Adopted Framework

As the framework in Figure 1.2 shows, the work undertaken to achieve efficient and effective on-device speech recognition on the mobile device are contained in chapters 4 and 5. The work done in chapter 4 is based on the pre-processing stage of speech recognition and the work done in chapter 5 is the speech recognition stage.

1.8. Thesis Outline

The rest of the thesis is structured as follows: Chapter 2 details the background theories and experimental infrastructure of the thesis. This provides the background information for the respective techniques adopted for use in the rest of the thesis including the Kohonen Self organising map, the K-means algorithm, Echo State Networks (ESN), Multi-layer perceptron

(MLP) and finally HMMs. Chapter 3 provides a literature review, summarising the different attempts in the literature in trying to address the problems of VAD and neural network based speech recognition. This chapter also provides an insight into the available HMM based on-device speech recognition system for mobiles. Chapter 4 presents the VAD work of this thesis. Both DFT and LPC speech features are investigated for use within VAD algorithms and the two novel VAD algorithms developed during the work of this thesis are presented. Chapter 5 describes the speech recognition stage experiments. The feature sets chosen and database used are given in this chapter together with the experimental investigations performed with an ESN based phoneme classifier. A novel DMLP word recogniser is also presented in this chapter and the results of its computation times and performance, as compared to the standard HMM technique are given. Chapter 6 gives a conclusion of the entire thesis as well as specifying the framework to be adopted for future on-device based speech recognition systems. This chapter also provides future work recommendations with regards to the outcome of the respective experiments.

Chapter Two

Background Information and Experimental Framework

2. Introduction

This chapter presents the theoretical background to the work undertaken in this thesis and the experimental infrastructures used in this thesis. This chapter is different from chapter 3 (Literature Review) in that it presents the theoretical background information about the techniques, mobile devices and databases used, whilst chapter three presents the different implementations of these techniques within the literature. The chapter is broken down into different sections. Section 2.1 gives the theoretical architecture and understanding of the human speech signal as well as the respective features that can be extracted from this signal. Section 2.2 then gives an insight into the techniques used to segment the speech signal into silence, word and phoneme regions. Techniques described include the Kohonen self-organising maps and the standard K-means algorithm. Section 2.3 presents a more detailed background explanation of the theories and concepts of the phoneme and word classification techniques used to convert frames of feature vectors into phoneme/word symbols. The techniques discussed include feed forward Multi-layer perceptron neural network, echo state neural networks and hidden Markov model concepts, together with a brief example of the problems that can be solved with them. Section 2.4 is focused on the two databases used for this research. The details as to the content of these databases are given in this section. Section

2.5 gives the specifications of the two mobile devices used for this research. Finally, Section 2.6 gives a brief summary of this chapter.

2.1 Speech Signal

Speech or rather sound is the creation and propagation of vibration or waves through a medium. In the case of human speech production; the vibrations are created by the glottis and the resonances of the vocal tract within the air generated from the lungs. It can also be said to be formed by convolving the excitation signal from the lungs with the filter response of the vocal tract [19].

There are three different ways in which speech signal can be excited;

- **Voiced Excitation:** The air pressure forces the glottis to open and close periodically, thereby generating a periodic pulse signal. This is used for vowels and consonants.
- **Unvoiced Excitation:** The glottis is open and the air passes through a narrow passage in the throat or mouth. This results in a turbulence which generates a noise signal. This is predominantly used for fricative sounds, examples include the letters “F” and “V”.
- **Transient Excitation:** The mouth is closed initially. This raises the air pressure and a sudden opening of the mouth creates an explosive burst. This type of excitation is mostly used for plosive sounds such as the letters “P” or “B” [24].

Words are then sequences of these three excitation modes. The shape of the speech signal is determined by the shape created by the throat, tongue, teeth and lips (vocal tract) [24, 25]. The nasal cavity can also be used to change the shape of the speech signal. The vocal tract is said to maintain a fixed set of characteristics over periods of 10 – 30msecs depending on the speakers speaking rate [25].

For a speech recognizer the most valuable set of information is contained in the spectral shape of the speech signal as it changes in time [26].

2.1.1 Speech Features

Recognition is achieved by extracting different features from the speech signal which are then used for recognition. The set of available speech signal features can be divided into two groups; the time domain features and the spectral/frequency domain features. To extract these features; the speech signal is first framed using a window, the Hamming and Hanning windows are the predominant window used in speech recognition [24]. The windowing is done in a way that each frame is within the 10 – 30msecs time frame in which the vocal tract maintains a fixed set of characteristics.

2.1.2 Time Domain Features

- a) **Short Time Energy Analysis (STE)**; this is computed by splitting the speech signal into frames of N samples and then computing the total squared values of the samples within each frame. The splitting of the signal is achieved by using a suitable window function to split the signal into desired frames [25]. The equation for the STE is given by;

$$E_n = \sum_{m=-\infty}^{\infty} (x[m]w[n-m])^2 \quad (2.1)$$

where E_n is the Energy of the nth frame of the speech signal and $x[m]$ is the amplitude of the speech signal in the time domain. $w[n-m]$ is the window within which the speech samples exist.

The STE feature is said to be very good for detecting word boundaries [27]. Many experiments have shown that the addition of energy information, as another model feature, improves system information and as such helps in voice activity detection [27, 28].

- b) **Pitch Period;** this is the distance between the ripples of the sound wave. The Pitch period of a signal is the time interval on which the signal repeats itself. This can be estimated for voiced speech because the pitch period is zero for unvoiced speech. The pitch period can be calculated in different ways for voiced speech [25].

2.1.3 Spectral Domain Features

- a) **Discrete Fourier Transform (DFT);** Due to the sinusoidal fidelity of sinusoids, it is relatively easy to decompose complex signals into them and process them in the frequency domain. The DFT basically decomposes the speech signal into a set of Cosine and Sine waves with basis function of each being given by;

$$ck(i) = \cos\left(\frac{2\pi k(i)}{N}\right) \quad (2.2)$$

$$sk(i) = \sin\left(\frac{2\pi k(i)}{N}\right) \quad (2.3)$$

where $k(i)$ is the sample within the speech signal, $ck[i]$ is the Cosine wave representation of $k[i]$ and $sk[i]$ is the Sine wave representation of $k[i]$.

This then synthesizes the sine and cosine components of the complex signal to give a DFT Spectrum. The DFT spectrum is said to contain the raw information about a speech signal [29]. Due to the computational load of computing DFTs, a faster technique was implemented by Cooley – Tukey [30]. This is known as the Fast Fourier Transform (FFT) and it is used to compute the DFT in a fast manner. The

DFT spectrum for each frame contains two parts, the second part is a symmetrical version of the first part.

- **Linear Predictive Cepstral Coding (LPC);** The main idea behind this method of feature extraction is that a speech sample is approximated as a linear combination of past speech samples [31, 32]. Linear prediction is mostly used in low bit rate transmission or storage of speech [25]. The LPC features are obtained by minimizing the sum of squared differences between the actual speech samples and the linearly predicted ones. The LPC process has also become the predominant technique for estimating the basic speech parameters such as formants, spectra, vocal tract area functions etc.[24]. The prediction of a speech sample $x[n]$ is given in equation 2.4 as;

$$\tilde{x}(n) = \sum_{i=1}^p \alpha_i x(n - i) \quad (2.4)$$

where $\tilde{x}(n)$ is the sample being predicted, $x(n - i)$ is the previous sample, P is the order and α_i is the prediction coefficient.

- c) **Prediction Error (PE);** The prediction error is a by-product signal of the LPC process. The prediction error decreases with the increase in P (order). This Prediction error is a very good approximation of the excitation source [25]. Based on the ascertained assumptions above, the prediction error will be large for voiced speech at the beginning of each pitch period. This feature can then be used to estimate the pitch period by subtracting between the large samples of the error signal. This can be used to identify pitch periods within samples [25].

$$e(n) = x(n) - \tilde{x}(n) = x(n) - \sum_{i=1}^p \alpha_i x(n - i) \quad (2.5)$$

where $e(n)$ is the error of the sample n and $x(n)$ is the time domain sample from the signal. $\tilde{x}(n)$, $x(n-1)$ and α_i are as defined in equation 2.4.

- d) **Mel Frequency Cepstral Coefficients(MFCC)**; MFCCs are based on the Mel scale, which is linearly distributed within the first 1000 Hz or Mels and is then logarithmically distributed above 1000Hz [33]. Basically the MFCC is computed by taking a linearly spaced frequency scale signal and then multiplying it with a set of triangular band pass filters [33]. Most speech recognition systems use the Mel frequency cepstral coefficients and its first and sometimes second derivative, as the input feature vectors of the speech signal, in order to better reflect the dynamic changes of the speech [24]. The equation for converting the frequency component to their Mel scale equivalent is given as;

$$Mel(f) = 1127 \log \left(1 + \frac{f}{700} \right) \quad (2.6)$$

where f is the frequency component to be converted to its corresponding Mel value.

The different speech features explained above have all been used for different aspects of speech recognition. However, MFCC features are the most commonly used features for speech recognition, while the other features are used for different aspects of speech processing. The LPC features have been adopted for speech recognition although their performance is not as good as the MFCC features. The poorer performance of the LPC features for speech recognition is due to the linear prediction analysis, which assumes the vocal tract to be a linear model. As such it adopts a linear set of analysis thereby making the features linear. However, this is argued not to be the case [34]. Whilst MFCC features are

based on the Mel scale, which was developed based on the human perception of sound and as such this makes it a better reflection of the actual speech content within a frame [35].

2.2 Phoneme and Word segmentation techniques

Phonemes are the basic units of a word. This allows different speech recognition systems to adopt a phoneme level segmentation approach to the recognition of different words. This involves segmentation of a given input word into the respective phonemes such that the speech recogniser is only required to recognize the limited set of different phonemes within the word rather than having to recognise all the different individual words. After recognizing the component phonemes, they can be synthesized to form the word.

The alternative word segmentation approach involves segmenting a given audio signal into the respective words contained in that audio signal. For example, an audio signal containing the digits 1 – 9 can be segmented into the respective digits of 1,2,3.....9. The individual segmented digits are then passed on to the speech recogniser. In this case, the recognizer is required to recognise the entire word thereby avoiding the synthetisation process of phoneme recognition.

Both these segmentation approaches have been adopted in this research. A self-organising map has been used for phoneme segmentation [36, 37, 38] while a K-means algorithm was adopted for word segmentation [39,40]. These two techniques are explained in the next section.

2.2.1 Self – Organising Map

A Kohonen self-organising map (SOM) is a type of neural network structure that offers unsupervised learning of temporal patterns presented to it as inputs. It is an abstract mathematical model of a topographic model [41]. In general, a Kohonen SOM can be viewed as a feed – forward neural network consisting of an input layer and the computational layer. The computational layer is structured as a map with rows and columns corresponding to the

desired classification output. As such a SOM can have multiple dimensions with modifications to the rows and columns. Each neuron in the computational layer is fully connected to the input layer.

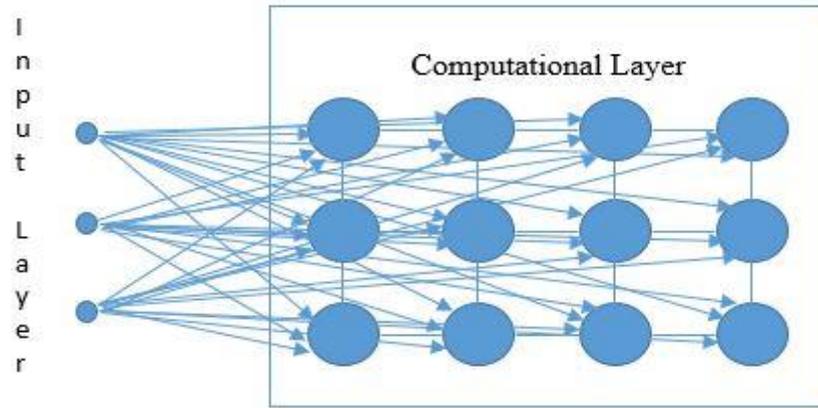


Figure 2.1 - An example of a (3 x 4) SOM structure.

In figure 2.1 a SOM structure can be seen with connections between the input and a 3 x 4 computational layer, which is also referred to as the output layer or map grid. The clustering steps involved in training a SOM to classify an input pattern can be summarised as;

- i) Selection of random weights between the input layer and output layer.
- ii) Using a Euclidean distance measure, the distance between an input pattern presented to the network in the input layer and the respective weight vectors corresponding to the neurons in the output layer are calculated. Equation 2.7 gives the Euclidean distance measure between two variables.

$$D(X, Y) = \sqrt{\sum_{i=1}^N (x_i - y_i)^2} \quad (2.7)$$

Where $\mathbf{X} = x_1, x_2, \dots, x_N$ is the input vector, $\mathbf{Y} = y_1, y_2, \dots, y_N$ is the output vector with N being the size of the vector.

- iii) The neuron with the minimum distance is subsequently regarded as the winner neuron and the input vector gets assigned into that cluster.
- iv) After the selection and assignment in step III. An update is performed on the weights of the winner neuron and subsequently all the other weight vectors of the neurons within a specific neighbourhood of the winner neuron

Steps ii to iv are repeated for every input pattern until the specified number of iteration is reached. After training, each neuron in the output layer or map is a representative of a set of training patterns. An unknown pattern is then simply classified by determining which output layer node has weights with the closest match to the unknown input pattern. The labelled associated with this winner neuron is the class of the unknown input pattern.

2.2.2 K-means

The K-means algorithm is a fast and efficient method of data classification. The K-means algorithm does not use weights in its calculations, rather it is used to cluster data based on the desired number of clusters. There are K_N clusters in the K-means algorithm. K-means assigns feature vectors to these clusters by the minimum distance assignment principle using the same Euclidean distance measure given in equation 2.7.

$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - c_j\|^2 \quad (2.8)$$

where $\|x_i^{(j)} - c_j\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j [42].

Steps involved in training the K-means clustering include;

- Choose initial cluster centers (randomly)

- Assign each sample vector to a cluster with the minimum Euclidean distance measure between the vector and the centroid of that cluster using equation 2.8.
- Compute new average as new center for each cluster
- If any center has changed or the error function changes dramatically, then go to step 2, else terminate [43].

At the end of the algorithm, the entire data would have been classified into one of the desired clusters K_N as specified.

There are similarities between the K-means and SOM. They are both data classification algorithms and as such are very good candidates for the phoneme or word segmentation task. While the SOM uses the weights attached to a particular neuron to learn the input data corresponding to it, the K-means algorithm uses the centroid of the respective clusters as the learned classification criterion using the Euclidean distance. Both algorithms adapt an iterative approach to the classification of the input. A set number of iteration is applied until the maximum number of iteration is reached, or when there are no more assignments of the input data into the different classes.

Once trained, an unknown input pattern is then classified by determining which cluster center has the smallest Euclidean distance to the unknown input pattern. The label associated with the nearest cluster is the class to which the unknown pattern is attached.

2.3 Phoneme and Word Classification Techniques

After an effective segmentation of the speech signal. A classifier is used to classify the respective segmented phonemes or words. Different techniques are used for this task which includes Neural Networks [38] and Hidden Markov Models [9,10]. A more detailed description of these techniques is given in the following sections.

2.3.1 Multi – Layer Perceptron (Feed – Forward Architecture)

A multi-layer perceptron (MLP) is a type of artificial neural network (ANN) with connections moving in one direction, from the input to the output [44]. MLPs are models designed to adapt to static training data. MLPs are arranged in different layers, with each layer having different number of nodes. Each node in a preceding layer is connected to every node in the succeeding layer, with the output of a node from the preceding layer being the input to the nodes of the succeeding layer. Figure 2.2 gives a simple (5x4x3) MLP structure, with 5 nodes in the input layer, 4 nodes in the hidden layer and 3 nodes in the output layer.

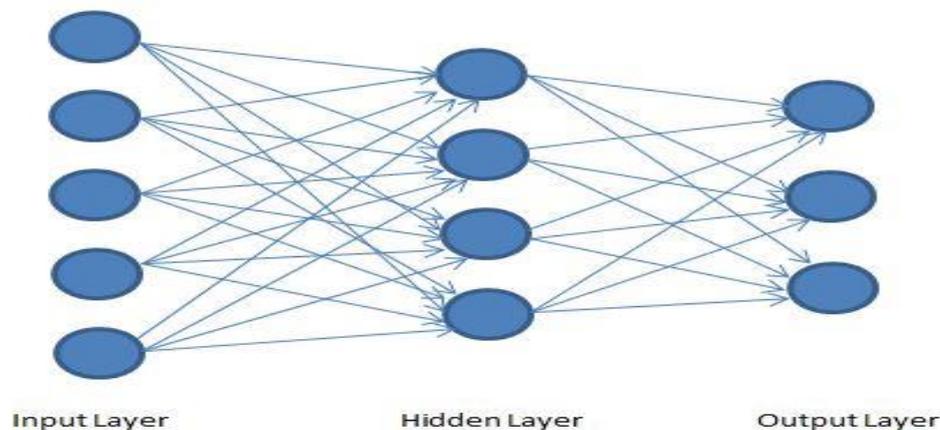


Figure 2.2 – An example of a 5x4x3 MLP with connections

In an MLP, connections between the different layers are established using weights. Each input node has weights connecting it to all the nodes in the succeeding layer, and also each node in the hidden layer has weights connecting it to all the nodes in the output layer. The number of weights between the input and hidden is a product of the number of nodes in the input and the number of nodes in the hidden. Subsequently the number of weights between the hidden and output is also a product of the number of nodes in the hidden layer and the number of nodes in the output layer. A node in the succeeding layer is activated using an activation function which is applied on the summation of the weights connected to that node. Equations 2.9 and 2.10 show the basic functions employed in each node:

$$Net_i = W_{i_0} + \sum_{j \in Pred(i)} (W_{ij}a_j) \quad (2.9)$$

$$a_i = f_{\log}(Net_i) \quad (2.10)$$

where a_j is the output of the preceding layer neuron, W_{ij} is the weight connecting the two neurons and W_{i_0} is the bias of the neuron. These are summed for all J in the preceding layer for which the connection $i \rightarrow j$ exists. a_i in equation 2.10 is the network output of the i^{th} neuron after applying an activation function. In the case of equation 2.10 it is a log activation function.

Different activation functions can be used for MLPs depending on the task an MLP is created to handle. The activation of a neuron is strictly based on its inputs as well as the activation function used. The activation function given in equation 2.10 is a logistic activation function. This is a common function used within MLPs where the output is required to be between the range of 0 and 1. Equation 2.11 shows the log activation function;

$$f(x) = \frac{1}{1 + e^{-\frac{x}{Q}}} \quad (2.11)$$

where x is the input of the activation function, in this case the network output in equation 2.9 and Q is the temperature of the neuron. This function tends to slightly change with an increase in temperature. Figure 2.3 shows the standard logistic function behaviour with $Q = 1$.

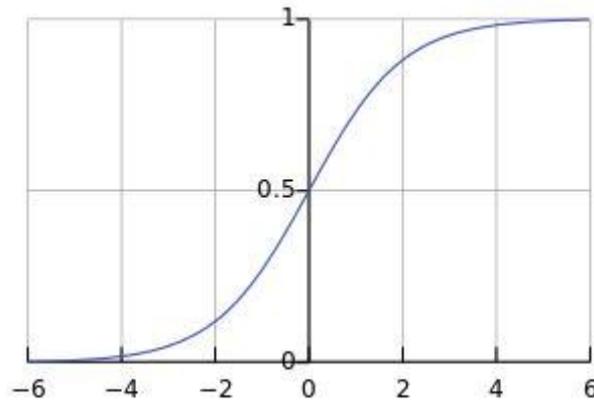


Figure 2.3 – A standard logistic function

Training of an MLP can be achieved using different techniques. The most common of such techniques is the back propagation algorithm. This algorithm is designed to minimize the mean squared error of the network. The training of an MLP is done in iterations over the entire training set. A full iteration that runs through the entire training set is called an epoch. The network is continuously trained in epochs until a desired epoch or a set minimum mean squared error value is achieved. The network then stops training. Training of an MLP is performed on the weights of the network. The weights are trained in each epoch until the output of the network becomes as close to the target output as desired or possible; given the training data.

The steps of the back propagation algorithm are:

- i) Calculate the output of the network after feeding an input pattern all the way to the output using equation 2.9 and 2.10 for each neuron.
- ii) Use equation 2.12 to calculate the error at the output of the network for each output neuron and then propagate that error backwards to the previous layers.

The output layer error is given by;

$$error_o = out_o(1 - out_o)(do_o - out_o) \quad (2.12)$$

where the error for the o^{th} neuron is $error_o$, out_o is the actual output of the network and do_o is the desired output of the network. This error is propagated backwards to the hidden layer. The error calculation for the hidden layer nodes is given as;

$$error_j = out_j(1 - out_j) \sum error_o W(t)_{jo} \quad (2.13)$$

where the error for the j^{th} neuron in the hidden layer is $error_j$, the output of the j^{th} neuron is out_j and $W(t)_{jo}$ is the weight connecting the j^{th} neuron and the o^{th} neuron in the output layer.

The last step in the back propagation algorithm is an update of the weights. The weights in the different layers are updated using the errors calculated in step II. The update of the weights are achieved using equations 2.14 and 2.15.

$$W(t + 1)_{jo} = W(t)_{jo} + \beta error_o out_j + \tau [W(t)_{jo} - W(t - 1)_{jo}] \quad (2.14)$$

$$W(t + 1)_{ij} = W(t)_{ij} + \beta error_j out_i + \tau [W(t)_{ij} - W(t - 1)_{ij}] \quad (2.15)$$

where $W(t + 1)_{jo}$ is the new weight connecting the j^{th} neuron in the preceding layer and the o^{th} neuron in the succeeding layer, $W(t)_{jo}$ is the current weight and $W(t - 1)_{jo}$ is the previous weight. τ is the momentum factor, β is the learning rate, $error_o$ is the error of the o^{th} neuron calculated in equation 2.12 and out_j is the output of the j^{th} neuron of the previous layer. The calculation of the new weights for the hidden layer is given in equation 2.15.

These steps are performed for each neuron and weight in the network, for every input pattern presented to the network. The network is trained until a desired epoch is achieved or the set minimum squared error is achieved [45].

After training, the MLP can be used to classify an unknown pattern by simply passing it through the network from input to output. The network output then represents the class vector of the unknown input.

2.3.2 Echo State Networks

An Echo State Network (ESN) is a form of recurrent neural network. Recurrent neural networks (RNN) represent a very powerful generic tool, integrating both large dynamical memory and highly adaptable computational capabilities [46]. RNNs are neural network structures that implement feedback connections. ESNs are practical, conceptually simple and relatively easy to implement. ESNs replace the feedback signal with a nonlinear response signal and then combine a desired output signal by a trainable linear combination of the response signals [47]. As compared to other RNNs structures, ESNs are computationally inexpensive to train. ESNs do not train the input or internal reservoir weights; they only train the output weights. This is one of the main reasons for its fast training. Figure 2.5 gives a brief view of such a process;

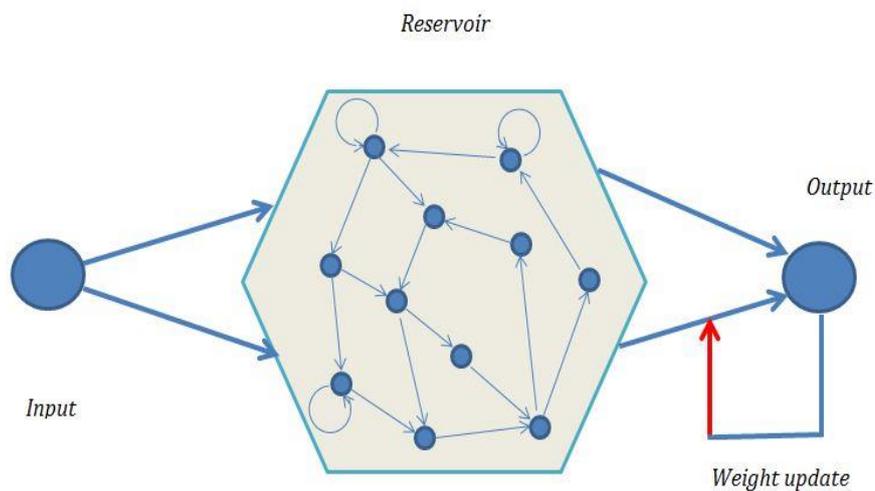


Figure 2.4 - An ESN network showing output weight updates

These type of neural networks are different from the networks prescribed before (MLPs and SOMs). This is because ESNs adopt a reservoir as the hidden unit and the nodes have some form of feedback associated with them, which are a representation of a part of the input sequence over time. This allows these networks to exhibit temporal learning within the reservoir layer. There are different architectural set-ups for ESN and a more detailed description of the network architecture adopted and used for experimentation in this thesis is given in chapter 5.

2.3.3 Hidden Markov Model

Hidden Markov Models (HMM) are a probabilistic technique that are regularly used for pattern recognition tasks [48,49]. As such, the mobile HMM speech recognition implemented in this work was used as a baseline technique against which the novel neural network approaches could be compared. HMMs are based on the concept of Markov Chains which represent random sequential process that undergo transitions from one state to another. The transitions of states in a Markov chain depends on the current state. The concept of a “hidden” Markov model is based on situations where the underlying Markov chains or Markov process is not readily identified. In general, a HMM classification can be regarded as the determination of the most suitable Markov chain to fit the observed signal. The process or chain is defined by the respective states in it and the transitions between one state in the chain to the next. HMMs consist of two components;

- i) State/Transitions; These specify the number of states in the Markov chain and their transition. This can be simply understood as the ordering of the states in the model.
- ii) Probability Distributions; The probability distribution of the states in a HMM define the distribution of the states observable. This should not be misunderstood as the transition, whereas the transition defines the ordering of the states, the probability

distribution defines the rules for the Markov chain to follow while doing these transitions.

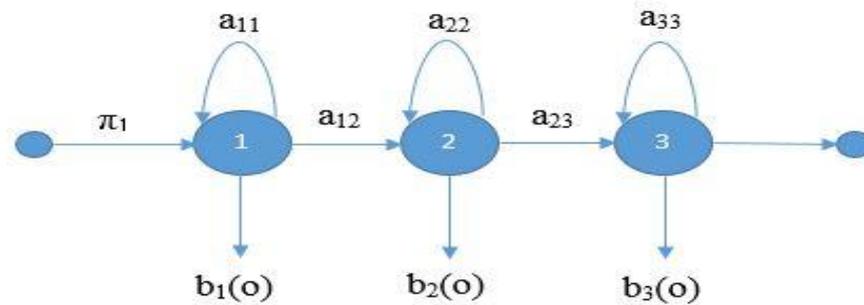


Figure 2.5 - A three state left to right HMM

A HMM can be defined mathematically as;

$$\lambda = (A, B, \pi) \tag{2.16}$$

where λ is the HMM, A represents the state transitions matrix, B is the observation probability matrix and π is the initial state of the Markov chain, which is simply the probability that a state is the initial state in the chain. This can be seen in figure 2.5

The transition matrix A is the transition matrix containing the transition probabilities for each state in the model. The total sum of the transitions for a chain of observation symbols is 1 (A is row stochastic). A percentage of it is distributed for each state. An example of this transition can be when walking from point 1 to 2. The number of states involved in the walking model can be regarded to as 2 (right leg R and left leg L). The transitions are R→L and L→R and the transition matrix would be (0.5 0.5) for A_{RL} and also (0.5 0.5) for A_{LR} .

There are 3 well defined problems that a HMM can be used to solved;

- i) Given the HMM model $\lambda = (A, B, \pi)$ and a sequence of observations O. Find $P(O|\lambda)$.

That is to determine the likelihood of the observation sequence O given the HMM λ .

- ii) Given the HMM model $\lambda = (A, B, \pi)$ and an observation sequence O . Find the optimal sequence for the underlying Markov process. This is used to determine the best possible ordering of states in a HMM (optimization).
- iii) Given an observation sequence O with its dimensions N (number of states in the sequence) and M (number of observation symbols). Create the HMM $\lambda = (A, B, \pi)$ that maximizes the probability of the observation sequence O .

These 3 problems can be used hand in hand. For example in the case of pattern recognition applications; problem iii) is used to train the system and then problem i) is used to find or rather test the system for the probability of that observation sequence [50].

2.4. Speech Database

Speech recognition problems have been studied and investigated using different databases. The particular type of database employed is mostly dependent on the type of speech recognition system or engine being built. To name a few commonly used databases available, there is the Texas Instrument and Massachusetts Institute of Technology (TIMIT) databases, Centre for Spoken Language Understanding (CSLU), Oregon Graduate Institute School of Science and Engineering database and National Institute of Standards and Technology (NIST) database. Some of the mentioned organisations have a wide range of speech databases, with each database having a different set up, with a different number of speakers and a different vocabulary set from the other databases. Due to the constraint of availability and finances with procuring databases; the work presented in this thesis is based on two databases, the CSLU database and the TIMIT TIDIGITS database.

- a) CSLU Database: This is a commercially available speaker verification database from the Centre for Spoken Language Understanding, Oregon Graduate Institute of Science and Technology Research Centre in Spoken Language Understanding. This database contains a total of 91 speakers. 20 of these speakers (11 females and 9 males) were used in this

work. The audio samples were recorded over a digital telephone line with a sampling frequency of 8 kHz to produce 8 – bit u-law files. The data is recorded in two sessions as a combined file containing the spoken English digits 1 – 9. The respective digits from these samples were extracted into the same sampling frequency of 8 kHz, 16-bit wave format. This resulted in a total of 2880 single digit files (20 speakers x 16 sessions x 9 digits in the files). The voice activity detection algorithms developed used the raw set of audio files to perform the voice activity detection. The 2880 extracted digits were also used for the recognition experiments. These were broken down into training and testing sets, with a random selection of 360 digits being used for testing and the remaining 2520 being used for training. Further details of this database can be found on the website[51]

- b) TIMIT TIDIGITS Database: The Texas Instrument and Massachusetts Institute of technology Digits database is one out of the several databases available for speech recognition from this organisation. The TIDIGITS database is a Speaker Independent connected digit corpus. The data was collected in a quiet environment and digitized at 20 kHz. This is a dialectically balanced database; that accounts for different accents over the continental US. The entire country was subdivided into 21 regions and data was captured for the different regions. There are a total of 326 speakers (111 male, 114 females, 50 boys and 51 girls), which are further divided into a set of training and testing sets. The training data has 163 speakers (55 males, 57 females, 25 boys and 26 girls) and the testing data also has 163 speakers (56 males, 57 females, 25 boys and 25 girls). Each speaker utters the digits 1 – 9 in two sessions. A summary of these details are given in table 2.1 for an easier understanding.

TIDIGITS Database Details		
Speakers	Digits	No. of Speakers
Male	1 to 9	111
Female	1 to 9	114
Boy	1 to 9	50
Girl	1 to 9	51
Total No. of Speakers		326

Table 2.1 - TIDIGITS Speaker details

The recognition experiments employed the database as it is. With no modifications to the data. More information about this database and other databases from this organisation can be found from their website [52].

2.5. Mobile devices

Two mobile devices were used for testing the different algorithms to determine their use for voice activity detection and speech recognition purposes. The chosen mobile development platform for this research was the Android mobile platform from google. The chosen mobile devices used in this research are;

- a) HTC Desire; this is a mobile phone released in March 2010 by High-Tech Computer Corporation. At current times, this mobile phone can be considered to be average in terms of processing power and memory. The specifications for this mobile phone are;
 - 576 MB RAM
 - 1 GHz Scorpion Processor
- b) Samsung Galaxy Note 3; the Samsung Galaxy Note 3 is a mobile phone made by Samsung and released to the market in September 2013. Though a little old at the time of writing, it still possesses a very powerful processing power as compared to the HTC. The Samsung Note 3's specifications are;
 - 3GB RAM
 - 2.3GHz Quad –core Krait 400

Both these mobile devices serve as good test platforms as their computational capabilities offer very distinct test scenarios. A good approximation can be made for the respective algorithms over the range of processing power that these mobile phones offer. This gives a good picture as to the performance of the algorithms for other mobile phones that fall within the computational capability range of these two, taking the HTC as the lower end of the spectrum and the Samsung Note 3 as the high end of the spectrum.

Both mobile devices cannot in anyway be compared to PC/Laptop based systems in terms of computational and memory capabilities. Even though the Samsung Note 3 is the more powerful of the two; it is still a fraction when compared to the computational and memory capabilities of a PC. While computational and memory capabilities of some PCs can be increased, this is not the case with mobile devices. At best, only the storage memory can be increased in some cases.

2.6. Summary

In this chapter the respective experimental infrastructures used for this research have been demonstrated. This infrastructure summary captures the full range of techniques developed in this research from the novel voice activity detection algorithms implemented, all the way to the neural network based speech recognition classifiers used in this thesis. This chapter also describes the speech signal and all the relevant features extracted from it as well as the specific mobile devices used in the experimental sections of this research. As will be shown in the subsequent chapters, it is the development of speech recognition algorithms that can accommodate the memory and processor limitations of mobile devices that are the rationale for this work.

Chapter Three

Literature Review

3. Introduction

This chapter presents a literature review of the speech recognition process with emphasis on the use of the different techniques and databases presented in chapter 2, with regards to their particular application for use on mobile platforms. Section 3.1 details the voice activity detection literature and techniques available. Section 3.2 then gives descriptions of the classifier techniques applied as the speech recognizers. Sub-sections 3.2.1 and 3.2.2 describes multi-layer perceptron and echo state networks phoneme/word classifiers. Whilst section 3.2.3 provides information about the baseline hidden Markov model used for comparison. Section 3.2.4 concludes with a description of the popular on-device sphinx implementation. Finally, section 3.3 is a summary of the entire chapter.

3.1 Voice Activity Detection

Voice Activity Detection (VAD) in audio sources is a classic problem in speech recognition. A precise VAD technique reduces the computation cost and response time of speech recognition systems by only passing detected speech frames into the recognition algorithm for processing [53]. VAD is a common pre-processing stage for speech recognition related activities such as speech coding, speech recognition, hands free telephony and echo cancellation. In addition, VAD can also be used to identify Voice, within an audio signal captured in noisy environments in order to provide good recognition performance [54]. In Global System for Mobile (GSM) based systems, VAD has been used to conserve battery power by truncating transmissions when speech pause is detected [55].

VAD based algorithms are divided into two parts;

- Acoustic Feature Extraction
- Decision Module

The first part normally deals with feature extraction from the audio signal. Different feature subsets can be extracted using different references of speech production and spectral adaptation. Some of the feature subsets are standard whilst others involve author specific signal decompositions. The decision module part is where a discriminative set of classifiers or sophisticated algorithms are developed to perform the task of clustering, segmentation or classification based on the different labels chosen by the respective authors for use in their papers.

In [53] the choice of speech features used in the VAD algorithm is the linear predictive coding (LPC) and residual signal (prediction error). This paper also uses the TIDIGITS corpus and Noise 92 databases for evaluating the algorithms developed. A new sub feature set called the prediction error variance, was extracted and subsequently fused with the LPC features as a feature for the VAD algorithm. An auto regression of 10, and a smothering order of 20 are used to extract the features. The prediction error variance is based on the actual residual signal. However, this is the variance of the original residual signal which is obtained by calculating the total mean squared error of the linear predictive model. The linear predictive model plus prediction error variance shows better numerical boundary detection when compared to the original residual error signal with regards to their start and end points. The TIDIGITS used were the digits 1 – 9. The algorithm was applied to the individual digits.

In [54], the cumulants of the harmonics of the prediction error is used in a VAD algorithm. Specifically the 4th order (Kurtosis) is preferred over the 3rd order (Skewness) and these are both normalized by the variance of the signal. The system developed is based on an

embedded system for use in a smart environment. This involves the use of the VAD algorithm developed in this paper to detect activities of human interaction and conversations within this environment. A bespoke test dataset of 45 minutes was created using different types of speakers and this was divided into 30 files of equal length with an 8 kHz sampling rate and frames of 256 samples. The Kurtosis cumulant is used in combination with the autocorrelation of the LPC. The choice of autocorrelation is influenced by the robustness of all cumulants of order above 2 being 0 for Gaussian distributed signals. This makes them robust to noise especially wideband noise. This combined feature vector had a general error rate of 9.5% compared to the standard energy profile feature with 13.3% and the Kurtosis alone with 16.3%.

A time frequency analysis of speech itself provides a better extraction algorithm as illustrated by [55]. The application of a Discrete Wavelet Transform (DTW), which is based on a windowed frequency representation of a signal, is proposed. This technique adopts a variable regional analysis within the windows which makes it suitable for dynamic analysis of the speech signal. By allowing adaptation of different speech lengths for analysis, the DTW technique can be deduced as a multi-resolution analysis technique. The signal is then further analysed using the Teager energy operator; which is a measure of the energy within the signal. It is argued that this is important because it enhances the discriminability between speech and noise. These are both combined to produce an auto correlated signal decomposed into 4 sub-bands and then synthesized to form a feature subset called the Speech Activity envelope. An adaptive threshold is applied in the decision module to produce the speech start and end points. The Noise 92 database is used in conjunction with a database that combines Mandarin and English audio signals. Speech is processed in a window of 256 samples with a 64 sample overlap. An average performance of 95.22% was achieved. However, this system

does not adopt a reconstruction of the original wavelet and it is based on an on-line implementation.

An entirely different frame analysis is introduced in [18] where the entropy of the respective speech frames is calculated. The main feature of an entropy profile is its lower sensitivity to changes in amplitude of the speech signal. This is argued to retain more detail as compared to the more conventional energy profiles. To extract the entropy profile, a pre-emphasis filter is applied, after which a band pass filter is applied. Subsequently, to extract the entropy profile, the processed signal is divided into frames of 25ms with a 25% – 50% overlap. A histogram with N bins (usually between 50 – 100) is used to represent each frame. The histogram is normalised to satisfy the statistical property of the cumulative distribution function (CDF) after which the entropy for each frame is extracted. This is applied to isolated digit utterances artificially joined together. This is claimed to improve recognition percentage by up to 12.9% as compared to normal energy profiles over a range of signal to noise ratio.

The need to extract features for VAD based on the complexity measure of a signal is investigated in [59]. The complexity measure of a signal is defined as its degree of order. This paper introduces a feature extraction technique based on complexity and puts the emphasis on the linear predictive modelling approach. As stated in [59], an LPC analysis is based on the vocal tract of humans. This is based on the argument that the vocal tract can be best modelled using a linear predictive model of order N . As such an LPC analysis is argued to enhance some frequency components of the speech signal. The proposed feature is based on extracting the LPC features and then subsequently binarizing these features with a set threshold using the mean of noise. The second feature proposed is based on the pitch of the speech signal. A pitch based filter is proposed, after the extraction of the LPC features. These are then binarized with the LPC features and a complexity measure is computed. However, this method is meant to be used in conjunction with other methods and, as such, does not

stand as an individual feature on its own. However, it does highlight the effectiveness of using LPC features for VAD.

The application of the prediction error (PE), due to its quasi-random nature for white noise, makes it potentially useful for VAD purposes [60]. A decomposition of the speech signal into a periodic (the harmonic components of the speech signal) and aperiodic (environmental noise and speech aperiodic components) is proposed. Using the coherence function, the similarity between the PE signal and the DFT signal is measured. An averaged measure in the frequency sub band for both the periodic and aperiodic components is calculated. This is evaluated using a discriminant analysis technique (DA) and a Bayesian approach with Gaussian Mixture Models (GMM). The TIMIT database is used with speech being framed at 16ms. The claimed performance for noiseless signal using DA is 99.19% and GMM 98.72%.

An autocorrelation function is used to extract features for use in [61]. A new feature subset is created as a fusion of the maximum peak of the autocorrelation and cross correlation values. The maximum peak is based on the idea that voiced speech has a higher maximum peak than unvoiced and silence/noisy signals. The cross correlation part utilizes the zero crossing rate (ZCR) to selectively analyse signals with frequencies between 50 – 500 Hz. The speech data is segmented into frames of 50ms and a period within these frames is selected as being between two ZCR points. These are then correlated and the correlation value is used in conjunction with the maximum peak value. The TIMIT database is used together with the QUT NOISE database. Evaluation gives a half total error rates of 11.1% (10 or 15db), 16.3 % (0 or 5db) and 28.7% (-10 or 5db). However, the application of multiple techniques in fusion is unlikely to be suitable for implementation on a mobile device due to the extra computation demand and time this incurs.

The application of the prediction error for co-channel usable speech detection in speaker identification systems is introduced in [62]. This idea is influenced from the spectral flatness of the speech for the prediction error signal. The structural changes in the prediction error are used to identify the usable speech frames. This assumes that the structure of speech frames are definite whilst those of non-speech frames are indefinite. The structure is used to identify the peak pulse distances of the speech frames and normalization is introduced for both male and female speakers to identify the usable speech measure. Evaluation using a speaker identification system yielded an 82% correct identification of usable speech. This work highlights the importance of using the prediction error but does not accommodate for timely usage.

The high order statistics (HOS) of the prediction error for VAD algorithms is proposed in [63]. The adaptation for online use was given in [54], however, a much more detailed rationale for extraction is proposed in [63]. This further uses the second order (variance) of the prediction error in conjunction with the 3rd and 4th order to decompose the signal into voiced, unvoiced and noisy signals. However, the feasibility of actually using this method is not given. An inclination is given as to a possibility of further use of these variants without giving a detailed feasibility with regards to speech recognition.

In [64] a rather distinct approach is adopted based on the short time energy of a frame. A rectangular window is used with frames being 10ms. A general assumption is made that the noise statistics are known prior to evaluation. This is argued to be supported by the Gaussian statistical model which argues that speech, noise and noisy speech are asymptotically independent Gaussian random variables. As such a fusion between the short time energy spectrum and the noise spectrum is adopted to detect voiced speech. However, the application of a rectangular window to the task of speech recognition, which is the ultimate aim in this research, cannot be achieved due to the Gaussian like assumption of the speech signal which

is better captured by a Hamming window or any Gaussian like distribute window. A pictorial comparison is given but not the application over a wide range of VAD adopting applications.

The Gaussian statistical assumption made in [64] was adopted from [65] where a statistical variation of the signals is used. This is applicable for bandwidth reduction and provides estimation and statistics in the frequency domain. Using a decision directed method on the likelihood ratio test estimation to produce a signal to noise ratio estimate for an unknown sample. A HMM based approach is used by assuming Markovian process estimation that represent transitions from speech absent to speech present. In [66] an improvement is applied to [65] with the adoption of the Laplacian and Gamma estimations in addition to the Gaussian probability distribution functions. This approach is claimed to provide a better performance in comparison to the Gaussian in other conditions. However, in this paper, the VAD algorithm was used for bandwidth reduction as opposed to speech recognition. As such, its application is limited for the purposes of this work.

In [67] an entropy based system is proposed similar to [18]. However the summation is not based on the histogram but on using the Teager based energy measure used in [55]. There is a 32 sub band decomposition of the speech signal. A low band energy and high band energy ratio is used to discriminate between voiced and unvoiced sound segments. The proposed band splitting entropy measure attempts to overcome the difficulties in estimating formant frequencies. This increases the accuracy of the measurement. The ratio of the low band energy to full band energy is used in conjunction with the Teager energy measure to make a decision. However, the adoption of a method based on the sub-band splitting of speech signals for a mobile device could be tricky especially with the need for time in synthesizing the signal afterwards and the variability of environmental usage can affect the different energy measures.

The sparse representation of an input signal over a defined dictionary containing atoms is given in [68]. This approach is claimed to have a slightly better performance over the Laplacian method of [66]. This approach uses a downsized TIMIT database with 80 different utterances. The sparse representation of a signal is obtained by solving the minimization problem of the dictionary given above. However, sparse representation can be considered as a method of detecting or characterisation of speech as opposed to possible use for recognition. This limits the suitability of such a method for mobile except in a hybrid use, which is not conducive for the computational power of mobiles.

The application of VAD in speech codecs is identified in [69]. This uses the same concept of wavelet transformation as [55] and uses a support vector machine for the classification/decision part of the VAD algorithm. The application of statistical classification methods can be seen as an advantage here. However, the transformation of the speech signal into a wavelet decomposition is not advantageous because this cannot then be further used to distinctively detect speech phonemes or characteristics that are good for speech recognition.

Phase information is proposed to be used as a feature subset for recognition and VAD. In [70] a Delta phase spectrum is proposed to be extracted and subsequently a Mel frequency based extraction is proposed from the delta spectrum for use in speech recognition. This paper presents ways of extracting such features and highlights observations to be made to compensate for the time step in feature extractions. This time step occurs due to the windowing effect on signals which moves the signal as opposed to the window. Unfortunately, a good deal of statistical analysis is required to compensate for this effect which again does not bode well for the limited computational power of mobile phones. Moreover, the need to compensate for the time step is adequately captured in a more simplistic manner by simply applying overlap of the frames to be analysed.

The consideration of the long term speech information as part of a VAD algorithm is argued by [71]. This paper considers the entire speech signal for VAD calculation. The long term speech divergence between the speech and non- speech frames of the entire signal are used to produce thresholds by comparing the long term spectral envelope to the average noise spectrum. A 98.15% detection rate is claimed using the TIMIT database with connected words. This paper highlights the advantage of the long term spectral information. However, in [72] it is argued that the adoption of the average noise spectrum magnitude information is not attainable because in practice there is no stability in the noise signal.

In [72] the long term signal variability is considered. This is a measure computed using the last frames of the observed signal with respect to the current frame of interest. The TIMIT database with 20ms frame length and a 10ms overlap is used to evaluate the algorithm. The only problem with a consideration of the last few frames as opposed to the first few ones, as described by [71], is that the availability of the desired noise spectrum cannot be guaranteed.

In [73] a different approach is applied still using the long term speech spectrum. The Kullback–Leibler measure is used to detect the speech and pause frames. The logarithmic filter bank energy was assumed as a Gaussian distribution in the Mel scale energy domain. This is then used to produce a probability distribution of 25ms frames with 10ms overlaps. A fixed or adaptable threshold to the observed noise energy is used. This paper claims an application to speech recognition and a slight affiliation to mobile communications. However, it does not identify the effects of applying this technique to mobile platforms with regards to computation times as well as not identifying the choice of features for the recognition process.

A hybrid application is introduced in [74] to the speech signal. An autoregressive modelling is applied to the linear predictive model while a general autoregressive conditional

heteroskedasticity (GARCH) is applied to the prediction error. However, an independent prediction error signal is considered and a structure based time series is used to abridge the two. MATLAB is used to test this method. This thesis does provide the possibility of a hybrid system for mobile platforms. However, this is suggested within the context of a hybrid system consisting of VAD using the prediction error and speech recognition using the linear prediction coefficients.

Clustering techniques are an important and emerging area of consideration for voice activity detection. This is because a pattern recognition approach can be applied to frames in order to detect when there is a pause in speech. In [75] a C means based clustering technique is proposed. A priori set of prototype noise/pause samples are used to come up with the data for the pause cluster. This is updated every time a pause section is found in the input signal. The long term speech information is also adopted in the analysis. However, an on line basic sequential algorithm is used to identify the speech frames. The selection is done by calculating the Euclidean distance between the two clusters. This method highlights importance of clustering techniques but for real time use an online application is adopted for speech detection which stands contrary to the rationale presented in this thesis for an on-device implementation.

In [27] an end point based algorithm is implemented in order to detect speech endpoints to discriminate speech from pause for isolated words. This algorithm employs the short time energy (STE) and zero crossing count (ZCC) for VAD purposes. This is due to the high amplitude of vowels in the STE spectrum and high number of ZCC for fricatives and other silence sounds. A combination of these two techniques is used due to their quick estimation from the speech signal. The only disadvantage to this method is the need to extract other parameters of speech for use in the speech recognition stages.

In [76] a linear cross correlation technique is applied to the discrete Fourier transform spectrum. This is also applied over the long speech spectrum to detect the presence of speech for use in a speaker verification environment. It is argued that the cross correlation of the DFT spectrum of the chosen digits produces a high correlation value for frames with speech and a low correlation value for frames without speech. However, the claimed high recognition percentage is based on the application of this technique to digits that are distinct in sound. This is suitable for speaker verification purposes but not for speech recognition where the order cannot be determined. The second drawback of this application is the computational time involved in calculating the cross correlation matrix which makes it unsuitable for implementation on a mobile device.

3.2 Speech Recognizer

Several different classification approaches have been employed within speech recognition systems. The predominant methods or techniques employed for speech recognition include Hidden Markov models, Gaussian Mixture Models and different forms of artificial Neural Networks. The work proposed in this thesis compares the use of Hidden Markov Models, Feed forward Multi – layer Perceptron Neural Network and Echo state networks as speech recognizers. The majority of the work presented in the literature is based on Hidden Markov models (with well over 40 years of research) as compared to the neural network approaches. This thesis attempts to redress that balance and is based on a neural network approach to on-device speech recognition for mobile devices.

3.2.1 Feed Forward Multi – Layer Perceptron (MLP) for Speech recognition

The feed forward multi–layer perceptron network is the most popular choice of neural networks for pattern recognition tasks. The application of MLP to speech recognition related tasks has moved from being the conventional standalone speech recognizer to being a

recognizer used either in fusion with some clustering techniques or as a parameter estimator for use in hybrid based speech recognition tasks.

As argued in [77] the recognition performance of ANNs are comparable to those of standard hidden Markov models. This paper presents a feed forward MLP network with a constructive training architecture. The number of hidden layers used is one. However, the number of neurons used within the layers is adaptive. This adaptation is implemented subject to the mean squared error of the training data not changing over time or the algorithm getting stuck in a local minimum. A 96.60% recognition rate is claimed over different ranges of recognition tasks. This paper shows the plausibility of applying MLPs for speech recognition but no actual speech recognition task is performed. The recognition rate is used to project possible recognition for other systems.

The application of artificial neural networks to mobile devices is highlighted in [13], where a speaker verification application is developed on a HTC Nexus One android device. A speech corpus of 110 samples is built using the same device. Though this paper does not apply to speech recognition, the plausibility of using ANN for speech processing on mobile devices can be seen in this paper. However, the major problem is that this paper does not mention the parameters used for this network as it is a work in progress.

The application of LPC coefficients to MLP speech recognition is explored in [78]. An MLP architecture with LPC coefficients as feature vectors are used for speech control of a mobile robot. The frame size selected is 240 samples with an 80 samples overlap. The LPC order used in this paper is 12. With 25 audio samples per word. This paper demonstrates the plausibility of using LPC features for speech recognition in a mobile based environment. However, this work does not handle the variability of the speech signal because it only passes a fixed length input to the MLP network.

A sequential MLP is introduced in [79]. The concept of a sequential network is borne out of the HMM approach to speech recognition, where the HMM network deals with observations as a transition between states. In [79], one MLP is used to mimic each state thereby making the transition between one MLP to the next MLP a transition between states. A dynamic time warping technique is used to achieve the sequential arrangement of the MLPs. A dynamic programming procedure with a generalized probabilistic descent algorithm is presented. The results are said to be comparable to Gaussian mixture density HMM. It may be interesting to conduct more research to parameterise its application but with regards to mobile application it is likely to be computationally expensive.

A comparative study is proposed in [80] employing different NN architectures for speech recognition. The MLP used has 3 hidden layers with a 20 x 20 x 15 structure and is evaluated on Turkish isolated digits. Unfortunately, a proper evaluation of the network used is not given as a result of the limitations of building neural networks using MATLAB tools.

In [81] an MFCC feature subset is used with an MLP as the classifier. With 120 coefficients representing a word, the MLP uses input feature vectors of size 120. This paper uses the sigmoid function with a learning rate of 0.4. However, a proper evaluation and characteristics of the network is not given. The network implementation is static.

In [82], MFCC and wavelet transform are used as the feature vectors for the recognition task. MATLAB was used to construct the MLP. The paper claims the wavelet transform presents a better recognition but does not provide a proper evaluation of the network structure.

In [83] MFCC is used with an MLP to achieve Hausa language digit recognition. An MLP was created using MATLAB for the recognition task. A 320 sample database is created and split into 70% training, 15% testing and 15% validation. The network uses 8 inputs with

10 outputs. Validation figures are given for the network performance but not the recognition performance.

In [84] the choices of features used are the DFT and LPC. However, the DFT spectrum used is below 3500Hz because there are no formants above that. The LPC order used was 10. The recognition rate claimed is 62% when presenting the best vector format to the neural network. As stated in the paper, more work needs to be done and as such the details of the network structure is not given.

In [85] an LPC based MLP is used to try and identify the robustness of neural networks to background variability with regards to speech recognition. One of the major issues with speech recognition is the variability of the length of utterances. In many cases, the solution to the variability of the speech length is to attempt to downsize the training dataset. As such, a SOM is used to cluster the data into 6 clusters. A fixed input and output MLP is then used for classification with hidden layers ranging from 1 – 3. As stated, a SOM is applied to deal with the variability issues of the speech signal. Unfortunately, the introduction of this extra stage is not advantageous for mobile implementation.

The use of a SOM in addition to MLP is used in a speaker dependent mode in [86], to classify the speech feature vectors for the Malay language. After clustering the frames, the SOM neurons are assigned phoneme labels. A sequence of observation vectors is thus converted into a sequence of phoneme labels which are then fed to an MLP. The MLP recognizes the sequence of phoneme labels to produce a list of different words using the recognized phoneme sequences. This is claimed to achieve a 4% better recognition accuracy than a standard MLP. Unfortunately, the amount of pre-processing is exhaustive and the subsequent processing time incurred in using 2 classifiers would not merit the 4% increase in performance on a mobile platform.

In [87] an advanced method is said to be created for speech recognition. The experiments are based on Persian digits with 600 recorded samples using half for test and the other half for training. An algorithm is used to downsize the data size by averaging the feature vectors belonging to an input and coming up with a mean feature vector which is then used to train the MLP. A claimed 98% minimum is said to be achieved for this experiment. However, an in-depth explanation of the MLP structure is lacking. Another MLP based phoneme classification is presented in [88] using the likelihood features extracted from a reconstructed phase space. The details of the MLP used here for classification is also not given. However, this paper is based on Persian vowels that are classified using Gaussian mixture models to produce a combinational vector of features that are fed to the MLP classifier.

A hybrid system comprising of ANN and HMM is proposed in [89]. This method uses the ANN as the clustering algorithm for the HMM and subsequently the recognition is based on the HMM. The system is said to have a 1% advantage over a normal MLP based system, with the application of a hybrid RNN having a better performance over the MLP. The problem with building a hybrid system for speech recognition without a significant advantage over a single system precludes its use on processing power limited devices due to the increased time consumption of the target system.

Another combination of MLP and HMM is presented in [90]. This is based on using the MLP as a feature vector in conjunction with the perceptual linear prediction features. A feature subset with 65 elements is created and used as an input to the HMM. In [91] a hybrid system is developed that applies an MLP to predict the next observation vector given the current observation vector and the HMM state the vector belongs to. Arabic digits are used for the experiments. Another hybrid system employing an MLP structure for German phoneme recognition is presented in [92]. The MLP structure used has 1188 input neurons corresponding to 9 frames with 132 feature vectors and 50 output units corresponding to the

50 unique phonemes. The phoneme labels are fed to the HMM for recognition. Best results are obtained using an MLP without a hidden layer.

It is clear from the literature the hybrid and combinational approaches can provide a superior performance when applied properly. In a computationally powerful environment, the increased performance their use produces holds plausibility. However, for on-device speech recognition on mobile platforms, the computational demand is too expensive. The adaptation of one technique, that is itself capable of performing recognition, as an input to another technique that effectively does the same task is simply too exhaustive.

3.2.2 Echo State Network (ESN) Applications to Speech recognition

Echo state networks have recently garnered a lot of interests as a good choice of recurrent neural network (RNN) architecture for use with time series prediction. However, the application of the ESN to speech recognition is still very new and research is still on going in this field. As such, there is no final best architecture for using such networks. In [93 – 95] a series of echo state network structures are developed that are comparable to the concept of left to right state transitions of the HMM. In [93] a competitive set of network filters are applied in such a way that they are arranged together to form a state. These filters are designed to transition from one state to the next while employing a winner take all strategy within the states. The switching of these filters for each time step is achieved by employing a dynamic programming technique that is employed to achieve the optimum path of transition. A segmental K-means designed for speech recognition [96] is employed to batch train the readout filters. 39 dimension vectors are used as input to the ESN.

Experiments were conducted to discern the effect of the spectral radius on the performance of the network. It is claimed that ESN have the possibility of being an alternative to the popular HMM in terms of human speech classification.

In [94] the same work is extended to using this classifier in a noise robust system. The developed system in [93] is applied using a different feature subset of 12 human factor cepstral coefficients and the log frame energy. This is said to show a superior performance over HMM models for 5 and 10 states of transition. A more discriminative approach is applied to this same method in [95] employing an offset asymptote using the hyperbolic tangent transfer function. This produced an improvement to the previous method.

A teacher forced training is implemented in [97] for an ESN. This is used to replace the actual output of the network. These teacher outputs are produced by using a maximum likelihood criterion stage on the network outputs prior to them being fed back into the network. An offline pseudo- inverse method is used to train the network output weights. The network uses a spectral radius within the range of 0.993 and 0.996. MLP vs ESN results are presented showing that the ESN outperforms the MLP implementation for Arabic phoneme recognition tasks using two separate datasets.

In [98] the choice of teacher forced training and feedback is abolished for state update and a spectral radius of 0.8 is used. An experiment into the reservoir size and comparison is given between a reservoir of 175 and 800 neurons with no significant difference in performance. A classification rate of 84.2% is achieved for vowel classification. The feature subset used is based on the Lyon cochlear features. This cochlear filtered data is fed into the network by aligning the different channels and using random connections to select certain channels as input to the network. The patterns are recognized based on the states of each neuron using a readout function.

In [99] the same Lyon cochlear features are used as the feature subset. The aim of the network designed here is to investigate the influence of the network topology and interconnection parameters on the computational power of the reservoir. This network is designed such that the connection fraction of weights, spectral radius and input scaling are

modified. Due to the aim in designing this network, the performance figures are not given. However, the effect of the connection fraction of weights (which is the number of weights initialised as zero and otherwise) is inconsequential. The input scaling is said to have some effect on the network. However, the biggest effect is seen to be due to the spectral radius of the network. In hindsight, this should have been obvious as the entire design of the ESN is based on the spectral radius property of the reservoir for it to exhibit echo states.

In [100] an ESN is used on feature subsets that include the Mel frequency cepstral coefficients (MFCC), Perceptual Linear prediction (PLP), RASTA- PLP and the Gabor technique features. The idea behind this network is to replace the linear readout function of the network with an Extreme kernel machine version of the extreme learning machine. The ESN using the Gabor features is found to perform best in clean environments and the RASTA – PLP in noisy environments. Full details of the network parameters are not given.

3.2.3 Hidden Markov Model Speech recognition

Hidden Markov models (HMM) have been the predominant technique used for speech recognition over the last 40 years. As such a full documentation of the HMM models literature is not possible within the short context of this thesis. This technique is only used for comparison purposes only in this thesis, therefore the literature review presented in respect to this technique is limited to the particular implementation of HMM used. The baseline HMM used in this work is the one presented in [101] where it was applied to isolated digit recognition task. The applied HMM is applied using a left to right transition implementation. This is meant to reflect the structure of the speech signal and the transition of the phonemes that make up a word. The HMM presented in [101] is trained to have a HMM representation of each word in the training dataset i.e. the digits 1 – 9. Each digit has its own HMM representation made up from all the representations of each digit in the training set. The

optimum parameters of each word are estimated using the Viterbi algorithm and a segmental K-means presented in [96]. To test an unknown word characterised by a sequence of observations, the probability of the word belonging to each trained HMM is calculated. The HMM with the highest probability is selected to be the winner and the unknown observation sequence is then classified as the word the winning HMM represents. A 98.2% performance figure is presented for a dynamic time warping based template representation. The identified challenges for HMM is based on the choice of spectral representation and segmentation or classification algorithm that best estimates the different spectral distribution of speech signals.

3.2.4 Pocket-Sphinx on-device speech recognition systems

The amount of information in the literature for on-device mobile speech recognition systems are very limited and non-existent in some cases. In general, for mobile applications the choice of implementation is an on-line based approach [56, 57, 58]. However, the Pocket-Sphinx system presented in [102] is based on the Sphinx II speech recognition system presented in [103] and the pocket sphinx SDK does make use of adopting strategies to cope with the limitations of mobile devices. As such, the system is a tuned down version of the Sphinx system that trades performance against use in an embedded system employment. The Sphinx II system in [103] II is based on semi-continuous HMM that employs a parameter sharing approach for use with HMMs. Further details of these approaches are given in the respective papers. The key omission in the paper are the real-time performance figures for the system. Given that the Pocket-Sphinx system represents a state of the art HMM based on-device speech recognition system, the novel on-device speech recognition system developed in this thesis is based on a neural network perspective of speech recognition. As such it represents a viable alternative to the HMM based Pocket-Sphinx approach because of the processing speed.

3.3 Summary

This chapter reviews the different VAD and Speech recognition techniques available in the literature. The techniques explored are critically analysed with the sole purpose of evaluating a mobile friendly approach using them. The development of a speech recognition system on a mobile has to consider the VAD approach used because there is need to terminate the listening process for the audio signal automatically to both conserve power consumption and reduce computational overheads. The identified problem of existing VAD algorithms is their over decomposition of the feature signal; making their feature sets inconsequential for subsequent use in an on-device speech recognition process. There is a gap in the literature with regards to algorithms that can facilitate real time processing of speech signals on mobile devices. It is suspected that this framework is lacking because the only approaches considered to date use an online approach where the audio signal is transmitted to a powerful remote server for processing. The resultant text then simply passed back to the mobile client. This invariably means any problem of communication between the mobile device and the server results in having no speech recognition.

The speech recognition part of this literature review highlights a key issue in the use of neural network methods for speech recognition. MLP classifiers have been neglected particularly when it comes to time series tasks due to their static input nature. This is shown in the literature where they are normally used in combination or as hybrid with other techniques. In the case of being lone classifiers; they are employed for simple tasks that do not take into consideration the speech length variability issue. This singular problem associated with MLP has shifted the attention of time series approaches towards recurrent neural networks. The echo state network is a much more recent approach to recurrent neural networks and it can be seen from the literature that this technique is still being applied with

different variations for speech recognition. No optimal ESN strategy of implementation has yet been determined.

Finally the baseline HMM approaches to off-line speech recognition and mobile speech recognition is reviewed as a comparison to the novel neural network architectures to be developed.

The experimentally determined effect of the choices made with regards to the novel Voice activity detection and Neural network based Speech recognition techniques developed in this work are presented in the following chapters.

Chapter Four

Mobile Voice Activity Detection (VAD)

4. Introduction

This chapter presents the respective VAD algorithm experiments performed in this thesis. These algorithms are based on the different features of the speech signal that can be extracted. As such the work presented in this chapter constitutes construction of the early part of the final framework representing the work done on feature extraction and Voice activity detection.

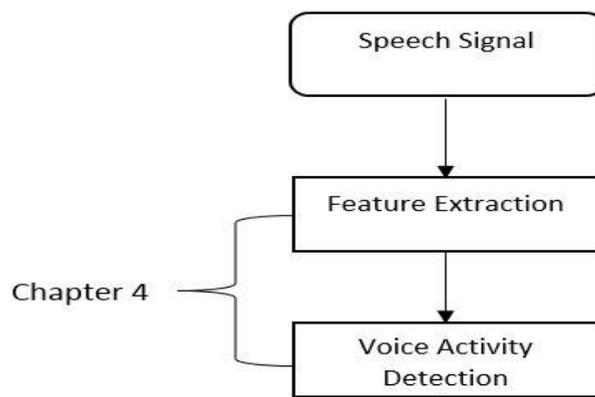


Figure 4.1 Former Part of Framework

The rest of the chapter is divided into sections. Section 4.1 starts by explaining why VAD is adopted for the speech recognition process on mobile devices. Section 4.2 then presents a series of VAD experiments conducted on a mobile device. These experiments are further divided within this section into subsections, with subsection 4.2.1 explaining the windowing pre-processing done prior to the feature extraction. Subsection 4.2.2 presents the results for the short time energy (STE) VAD algorithm, whilst subsection 4.2.3 presents the results for the raw powered prediction error (PE) VAD algorithm. Subsection 4.2.4 then compares the results of the STE and PE VAD algorithms. Subsection 4.2.5 details a related

set of the linear cross correlation (LCC) VAD experiments and subsection 4.2.6 presents the results from these LCC experiments. Section 4.3 then presents two novel algorithms based on K-means clustering on the log PE and standard deviation of the LPC features. Finally a full summary of the results presented in this chapter is given in section 4.4.

4.1 Why Voice Activity Detection?

Speech Recognition processing is a computationally exhaustive task that imposes heavy demands on the processing power and storage data capacity of any target device [18]. Efficient automatic speech segmentation pre-processing can reduce this strain on the mobile resources by removing silence and noise frames prior to the speech recognition processing stage. In addition, there is also a need for the development of a speech pause detection within an automatic mobile speech recognition system to aid in the segmentation of the spoken words and also to stop the microphone recording process in order for the recognition part of the process to start.

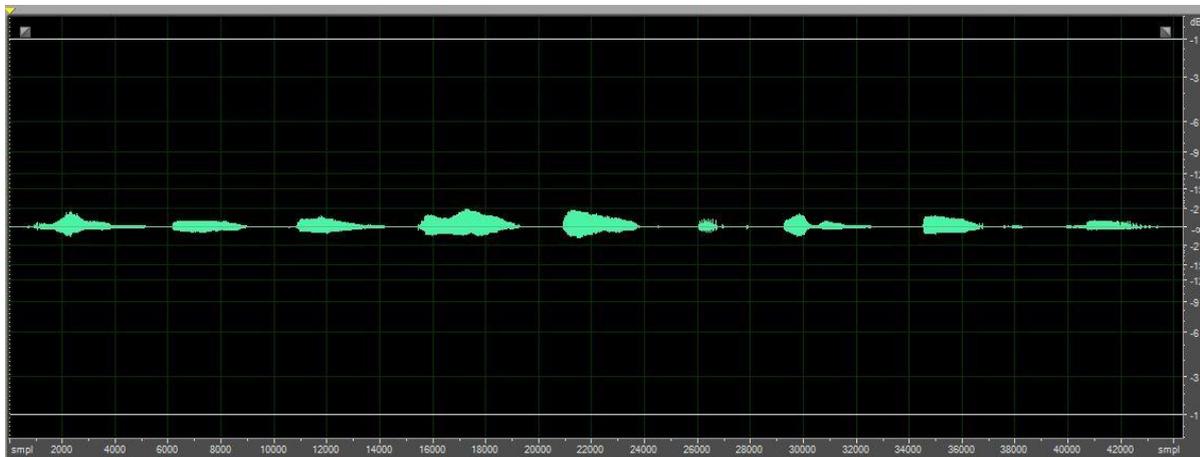


Figure 4.2 - Time domain speech signal containing the digits 1 - 9

Observing the speech signal image shown in figure 4.2, it is easy to spot the regions with speech and the regions without speech. Segmenting this speech signal into sections of speech and silence would allow the system to only process the speech frames whilst ignoring the silence frames. In table 4.1 the regions with speech in figure 4.2 are segmented manually to

identify and illustrate the amount of silence samples that can be ignored for processing with an efficient VAD algorithm; thereby saving processing time.

SAMPLE: 00381VK1 Total Number of samples = 44330			
DIGIT	FIRST SAMPLE	LAST SAMPLE	LENGTH
ONE	709	5213	4504
TWO	6097	8942	2845
THREE	10813	14200	3387
FOUR	15412	19293	3881
FIVE	20858	23878	3020
SIX	25986	28499	2513
SEVEN	29233	32616	3383
EIGHT	34488	38379	3891
NINE	39842	43497	3655
Total Speech Frames		31079	
Total Silence Frames		13251	

Table 4.1 - Manual borders of time domain signal in figure 4.2.

To achieve an effective mobile voice activity detection algorithm a number of established and novel algorithms were developed. These algorithms are based on the different speech features that can be extracted. A detailed description of the respective algorithms is given in sections 4.2 and 4.3. The audio signals used for the respective VAD algorithms are the audio signals from the CSLU2002 database described in section 2.7. As explained in that section this database uses an 8 kHz sampling rate with a 16bit encoding.

4.2 Other Voice Activity Detection Experiments

In this section of this chapter, the baseline experiments conducted prior to achieving the final proposed set of algorithms are presented. These were experiments undertaken to determine a baseline measure of performance with existing VAD algorithms that can be compared to the two novel algorithms. As argued by [59, 60, 61] every VAD algorithm goes through the feature selection stage initially before it passes on to the decision module stage. As a consequence the different VAD algorithms implemented are based on the different features of the speech signal. These features are;

- Short Term Energy (STE)
- Linear Predictive Coding Residual Signal (Prediction error)
- Linear Predictive Cepstral Coefficients (LPC)
- Discrete Fourier Transform Coefficients (DFT)

The experiments conducted applied different methods to extract the information of speech and silence from the time domain signal.

4.2.1 Speech signal Pre-processing

Prior to extracting any of the relevant speech features from the speech signal, a Hamming window is applied to segment the audio signal into frames. The Hamming and Hanning windows are the predominant windows used for speech recognition analysis [24,104]. The Hamming window is based on equation 4.1.

$$W_i = 0.54 + 0.46 \cos\left(\frac{2\pi n}{N-1}\right) \quad (4.1)$$

where W_i is the i^{th} window and N is the total number of samples within the window.

The speech signal has to be processed in short time frames of between 10 – 30 msec. This is to enable the speech processing to capture the periodicity in the speech signal as the vocal tract is said to maintain a fixed set of characteristics for between 10 – 30msec. The Hamming window used in this thesis is used to segment the speech signal into frames of 16ms which translates to a total sample number of 128 per frame for the CSLU2002 database and a total of 320 samples per frame for the TIMIT database audio signals.

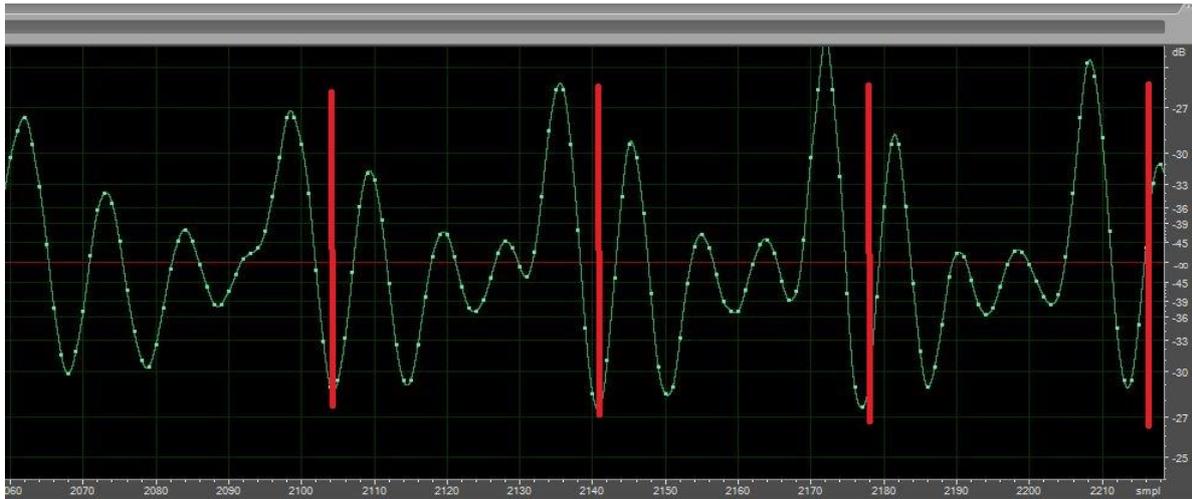


Figure 4.3 - Periodicity plot

The periodicity of the speech signal can be seen in figure 4.3 between the red lines. Observing the number of samples within the figure, it can be seen that there are a total of 110 samples across the four periods. These four periods occupy 13.75msecs, which falls within the 10 – 30 msec range. As such a total of 128 samples per frame is enough to capture this periodicity.

4.2.2 Short Term Energy Voice Activity Detection Experiments

The first set of experiments conducted involved the use of the short time energy feature (STE) of the speech signal. The short term energy is said to give a good indication of the presence of speech within a frame. The short term energy of voiced frames is said to be higher than those of unvoiced frames [27]. Better yet the short time energy of vowel sounds is very high when compared to other phonemes [25]. To achieve an effective short term energy algorithm an additional pre-processing step is applied.

a) Short Time Energy Analysis

The framed speech is first normalised by subtracting the DC offset of the entire speech signal from the respective samples within that frame. The DC of the entire signal is computed as the mean of the entire waveform. After subtracting the DC offset from the respective speech samples within the windows, a STE analysis is performed. This is done by squaring

each sample within the window and subsequently summing all the squared samples within the window. The total sum of the squared samples within window is then the short time energy of that window. This is computed for the entire speech signal, using the windows. The equation for this computation is given in equation 2.1 (page 12, chapter 2).

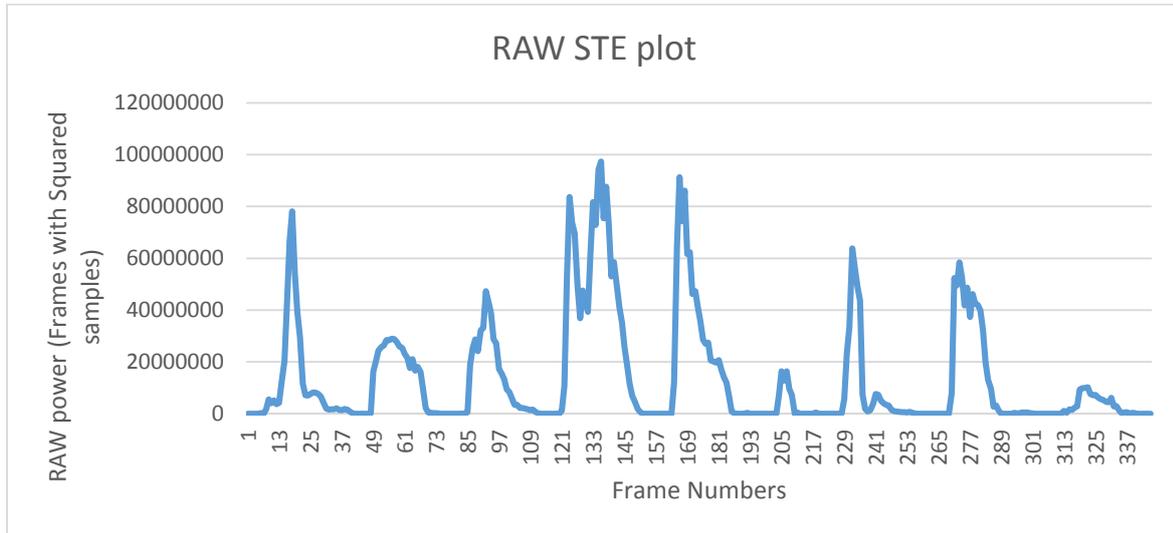


Figure 4.4 - Raw Powered STE plot of Speech signal with 346 frames

In figure 4.4 the raw powered short time energy of the frames created from a speech sample of count 1 – 9 is displayed. This is the raw short time energy without taking the logarithm of it. The energy profile of the digits 1 to 9 can be easily seen in the plot. The spectral envelope of each digit and the regions of silence can also be seen. The preferred choice of representation of the STE is achieved by taking the log of the raw energy profile. However, the raw powered STE was also used in order to achieve a better threshold resolution.

b) Short Time Energy VAD Algorithm

The STE VAD algorithm is based on a threshold decision module. This thresholding algorithm was developed to automatically select a threshold value, which is then used to segment the signal into the regions of speech and region of silence. The steps of the algorithm are given in the Pseudo code of table 4.2.

<i>STE Algorithm Pseudo code</i>
<i>For STE of every frame F_n</i> <i>If $F_n > Max$</i> <i>Max = F_n</i> <i>End</i> <i>T = Max x ST</i> <i>For every Frame F_n</i> <i>If $F_n > T$</i> <i>Insert F_n into VC</i> <i>Else if $F_n < T$</i> <i>Insert F_n into UC</i>
<i>End</i>
<i>Frames with speech VC</i> <i>Frames without speech UC</i>

Table 4.2 - STE Algorithm Pseudo code

F_n = Raw Short time energy of the nth frame
Max = Value of frame with the highest STE value
T = Threshold
ST = Selected Threshold (chosen percentage of Max)
VC = Voice Cluster
UC = Unvoiced Cluster

The algorithm initially determines the threshold T, which is a percentage of the maximum value of the frame with the maximum STE measure. This is determined by first looping through the entire frames and selecting the frame with the maximum STE value. Then a chosen percentage measure ST is used to determine the threshold T for that audio signal. The chosen percentage measure could be determined by the user. Experiments were conducted to determine the optimum generalizable threshold to be used for any input signal. After the threshold T is selected, another variable called frame distance is used to separate the frames in the voice cluster (VC) in order to group them into their respective digits. The second determinant variable called the frame distance is used only after determining the threshold value and effectively clustering the different frames into voiced frames and unvoiced frames.

After segmenting the different frames in the voice cluster into the respective digits. The mobile device was used to play the respective digits to determine if they were correctly recognised or not. Only fully recognized digits are considered as recognized. Partially recognized digits were identified but not counted in the recognition figures. Different experiments were conducted alongside the threshold experiments to determine the most effective frame distance measure to effectively classify the digits. The results are presented in section 4.2.4

4.2.3 Linear Predictive Coding Residual Signal (Prediction Error) VAD Experiments

The prediction error (PE) signal was also used for voice activity detection. This is said to be a good approximation of the excitation source of speech [25]. The idea behind the use of the prediction error signal is due to its spectral similarities to the raw STE plot. The idea behind this algorithm is similar to the STE algorithm, however, the methods of extraction and processing are different.

a) Linear Predictive Coding Residual signal (Prediction error) Analysis

The pre-processing applied to the prediction error (PE) is similar to every other feature extraction, i.e. the use of a Hamming window. The residual signal is a by-product signal from the linear prediction analysis. To extract the prediction error, the linear predictive cepstral coefficients are first extracted using the Levinson–Durbin approach [32]. The prediction error is computed as the difference between the predicted sample and the actual sample. The prediction error is based on equation 2.5 (page 14, chapter 2). The order used for the LPC analysis is 12 and thus produces 12 prediction error features per frame are produced.

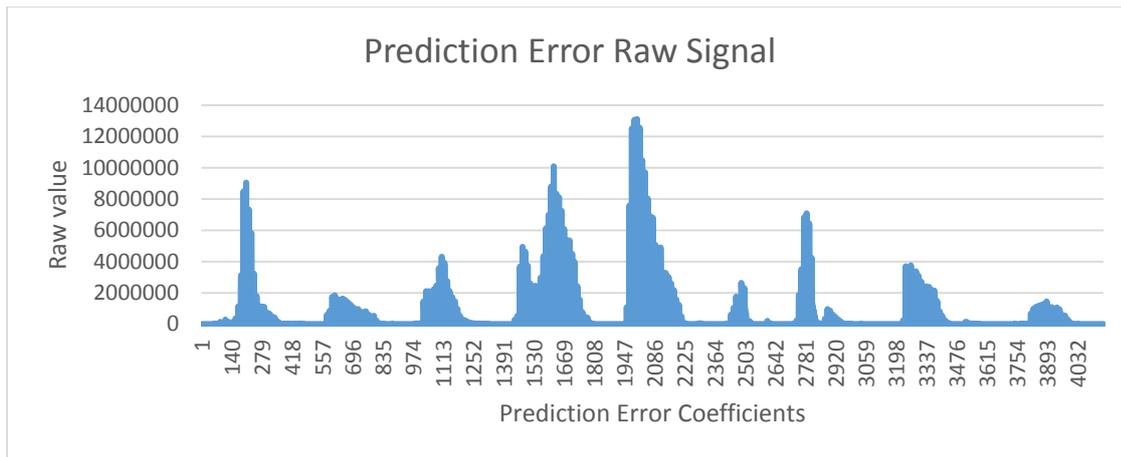


Figure 4.5 - Raw Power Prediction Error signal of 5.541 seconds speech signal

From figure 4.5 it can be seen that the raw prediction error profile is very similar to the short time energy profile of the same signal. This similarity prompted the decision to create a second algorithm for VAD using this feature in order to see if the PE signal is better for VAD purposes than the STE feature. As argued by [25] it is a good approximation of the excitation source of the human speech. This renders it susceptible to a better capturing of the components of the speech signal to distinguish between silence and speech frames.

b) Raw Prediction Error VAD Algorithm

Due to the explained similarities of the STE spectral envelope and that of the PE signal. The same set of procedures for the experiments were adopted. The optimal parameters used for the STE plot were maintained and applied to this signal. These optimal values were used in order to see which feature subset gives a better performance. As stated the only difference in the two algorithms is the choice of features. The concept of the algorithm was the same, the maximum value or a percentage of it is used as the threshold to distinguish between the speech and non-speech regions of the speech signal.

4.2.4 STE VAD Versus PE VAD Results

A total of 10 speakers were used for the evaluation of these two VAD algorithms. These 10 speakers were randomly chosen from the CSLU database. However, prior to this

evaluation, a different set of speakers were used to determine the optimum values for threshold. The frame distance value was also determined at this point. This frame distance is meant to provide a separation between the frames that were adjudged to be voiced frames.

Table 4.4 shows the comparison between the results of the two stated algorithms.

Threshold: 0.81					Frame Length: >=7				
Sample	1	2	3	4	5	6	7	8	9
Speaker 1	0	0	0	0	0	OX	0	OX	0
Speaker 2	0	0	0	X	0	OX	OX	0	0
Speaker 3	0	0	0	X	0	OX	0	0	0

Table 4.3 – Output evaluation table for raw powered experiments.

O = Fully Recognised.
 OX = Partially Recognised
 X = Not Recognised

After segmentation of the respective regions of the speech using the algorithms stated under section 4.2.2 and 4.2.3, a verification of the regions recognised was performed manually. In table 4.3, the “O” means fully recognised. These are the only digits that are computed as recognised within the total recognition figures shown in table 4.4. The partially recognised (OX) digits were those digits either split or which only some phonemes within them were recognised. As for the digits marked X, they were not recognised.

Feature	Threshold	Frame Distance	Recognized
Prediction error	0.81	>7	80%
STE	0.81	>7	73.30%

Table 4.4 - Final results of STE Vs PE of 10 CSLU database speakers

The results provided in table 4.4 shows that the Prediction error VAD algorithm performs better than the STE VAD algorithm. As stated before, the STE features are enhanced by the amplitude of the vowels in the spoken digits, which is why it has been combined with other feature vectors in different works [27, 64]. However, the prediction

error signal is an approximation of the excitation source which makes it less dependent on the vowels in the different digits such that it provides a good resolution of the other phonemes.

These results rendered the prediction error as the more suitable choice for further experimentation. However, with an error rate of **20%**, it is necessary to try and improve the performance figure of the VAD stage, in order to only pass on relevant and accurate information to the speech recognizer. Further experimentation on the Prediction error led to one of the novel algorithms proposed in this thesis; as described in section 4.3.1. However, prior to achieving that level of success, a series of other VAD algorithms were developed and evaluated.

4.2.5 Linear Cross Correlation (LCC) VAD Experiments

The linear cross correlation between two variables X and Y is given in equation 4.2. The correlation value is within the range of 0 and 1. If there is close correlation between the two variables, the correlation value is high (closer to 1) and low (closer to 0) for variables without any correlation.

$$LCC(x, y) = \frac{N(\sum_{i=1}^N x_i y_i) - (\sum_{i=1}^N x_i)(\sum_{i=1}^N y_i)}{\sqrt{N(\sum_{i=1}^N x_i^2) - (\sum_{i=1}^N x_i)^2} \sqrt{N(\sum_{i=1}^N y_i^2) - (\sum_{i=1}^N y_i)^2}} \quad (4.2)$$

where x_i and y_i are two vectors of N samples.

The linear cross correlation is a combination of different statistical entities as contained in equation 4.2. To sum up the meaning of equation 4.2, the LCC value between the two variables X and Y is the covariance between X and Y divided by the product of the standard deviation of X and Y. In [76], this technique was applied to discrete Fourier transform (DFT) feature set of the speech frames where it correlates the DFT frame feature vectors against each other, resulting in a higher correlation value for frames containing speech and low value for frames without speech.

a) Discrete Fourier Transform (DFT) Spectrum LCC VAD experiments

The first step in extracting the DFT spectrum is to window the speech signal and the window size chosen for this experiment is the same as that used in section 4.2.1. After extracting the windowed speech signal, the DFT is extracted, as explained in chapter 2. However, the full DFT of the windowed speech signal results in a DFT spectrum of 128 samples. As explained in chapter 2, the DFT is extracted using the Fast Fourier Transform method of [30]. This is simply done by multiplying the even numbered samples using equation 2.2 and the odd numbered samples using equation 2.3 (page 13, chapter 2). The samples are then later synthesized into a single spectrum. The DFT spectrum is a half symmetric, which means the first half of the spectrum is identical to the second half of the spectrum reversed.

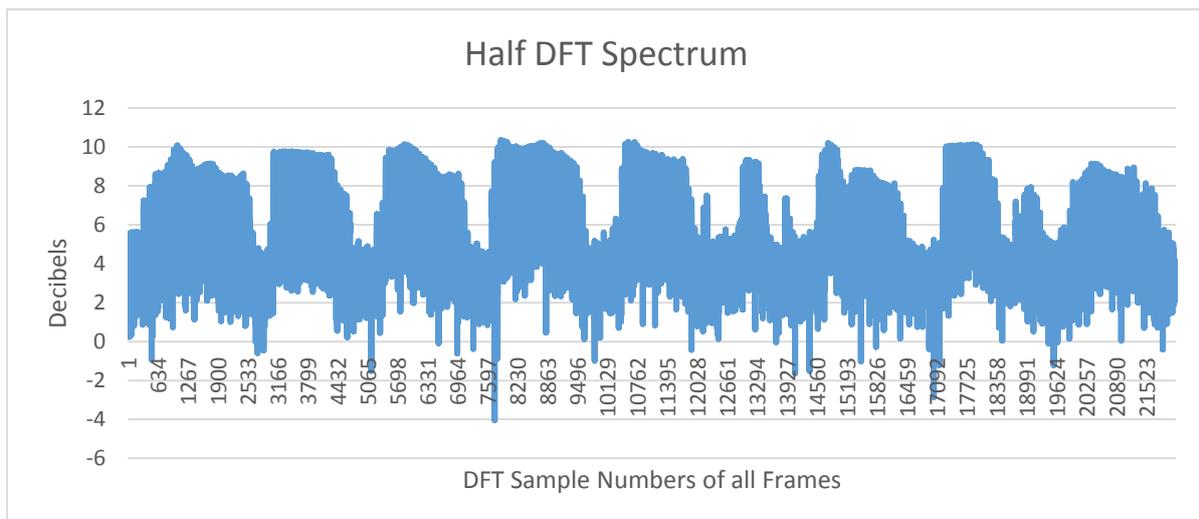


Figure 4.6 - Half side DFT Spectrum of 5.541 seconds speech signal

For this DFT LCC VAD algorithm the first half is chosen, resulting in a DFT spectrum of 64 samples per frame. Figure 4.6 shows the 64 DFT spectrum containing the digits 1 – 9. The spectral envelope of the respective digits can be picked out compared to the silence regions. The concept of the LCC involves correlating the frames in the speech signal against each other including itself. The pseudo representation of this process is given in table 4.5.

<i>LCC Algorithm Pseudo code for N number of frames</i>
<i>For every frame F_c</i> <i>For every frame F_n</i> <i>compute LCC between F_c and F_n using equation 4.2</i> <i>End</i>
<i>End</i>

Table 4.5 - LCC algorithm computation Pseudo code

F_c = Candidate Frame
 F_n = Frame to be correlated with

The first frame is chosen as the candidate frame and it is correlated against itself and the rest of the frames in the speech signal. The second frame is then chosen as the candidate frame and it is correlated starting from the first frame in the signal all the way to the last frame. This is done for all the frames. An epoch for this algorithm is after it has iterated through all the frames in the speech signal. This gives a two dimensional matrix of the correlation values of the respective frames in the speech signal against each other. The results of this are presented in section 4.2.6 where it is compared to the LPC LCC and PE VAD algorithm results.

b) Linear Predictive Coding Coefficients (LPC) LCC VAD experiments

The LPC feature subsets were also used for computation with the LCC algorithm. To extract the set of LPC features from the speech signal it is also windowed and uses the Levinson – Durbin recursion algorithm to extract the LPC features. The order of the LPC features extracted is 64 because there is no symmetry in the LPC feature spectrum. However, other orders of the LPC spectrum were also extracted in order to determine the optimum number of LPC features for the LCC algorithm. The process and procedure adopted is exactly the same as given in table 4.5 except that the frames contain LPC features rather than DFT features. The results are also provided in section 4.2.6 in comparison with the other feature subsets.

c) Log Prediction Error (PE) LCC VAD Experiments

The extraction process of the prediction error was explained in section 4.2.3. The order of the prediction error extracted is the same as for the LPC process and experiments (64). Other values were also extracted alongside the LPC as well for experimentation. However, the key difference in the extraction process of the prediction error for this experiments as opposed to the previous experiments of section 4.2.3 is that the log of the spectrum is used.

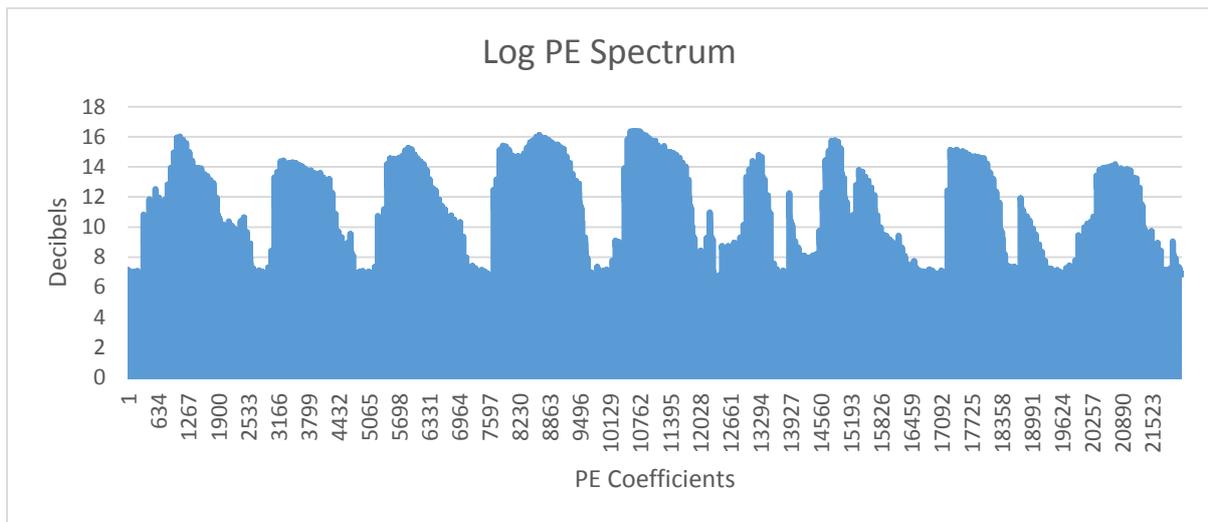


Figure 4.7 – 64 order Prediction Error of 5.541 seconds speech signal

In figure 4.7 the log prediction error is displayed. The unique advantage of the log prediction error is the spectral flatness of the background noise and regions of silence. From the plot it can also be seen that the log prediction error has a very similar spectral envelope to the original DFT spectrum as illustrated in figure 4.6. The log prediction error is then used as the feature vector for the LCC process and the procedure for the calculation of the LCC algorithm is the same as that of the other two features. The final computed correlation values for all the frames are also presented as a two dimensional matrix. The next section compares the respective results of the three LCC VAD algorithms.

4.2.6 Linear Cross Correlation Results of DFT vs LPC vs PE

This section of the results is split into two. The first part provides the respective LCC plots showing for the different speech features and the second part provides a comparative evaluation of these results.

a) LCC Plots of DFT, LPC and PE

The LCC results are displayed in the original matrix form. According to [76] the extraction of the regions with speech lies in the diagonal middle of the plot and this paper also provides an equation to select the regions of speech.

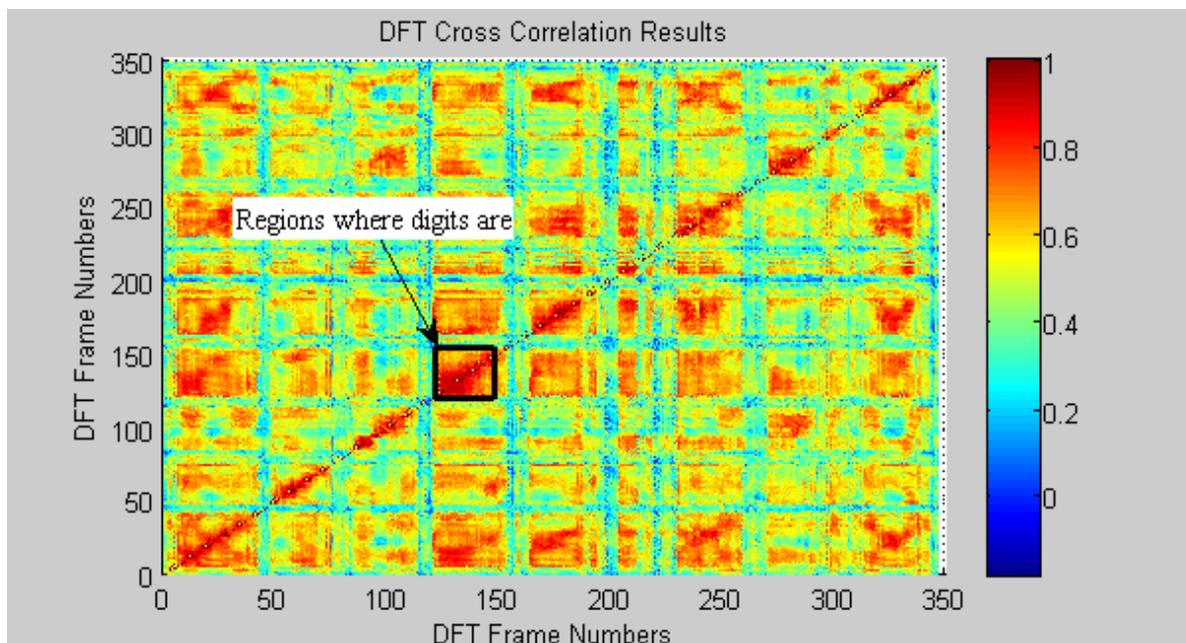


Figure 4.8 - DFT Cross Correlation Plot of speech signal of 5.541 seconds

The DFT plot in figure 4.8 shows the results of the LCC for the DFT spectrum. The horizontal and vertical axis of the plot are the frame numbers. The dark regions are the regions of higher correlation and this is found in the diagonal middle of the plot. The black box in the plot shows how wide the detection algorithm searches through for the digits in the speech spectrum. A moving box is applied to detect the rest of the digits in the DFT spectrum. This moving box is based on the work of [76]. However, prior to the application of the respective digit extraction within the correlated results, there has to be a distinctive

separation of the digits within the correlation figures to do the extraction of any particular digit.

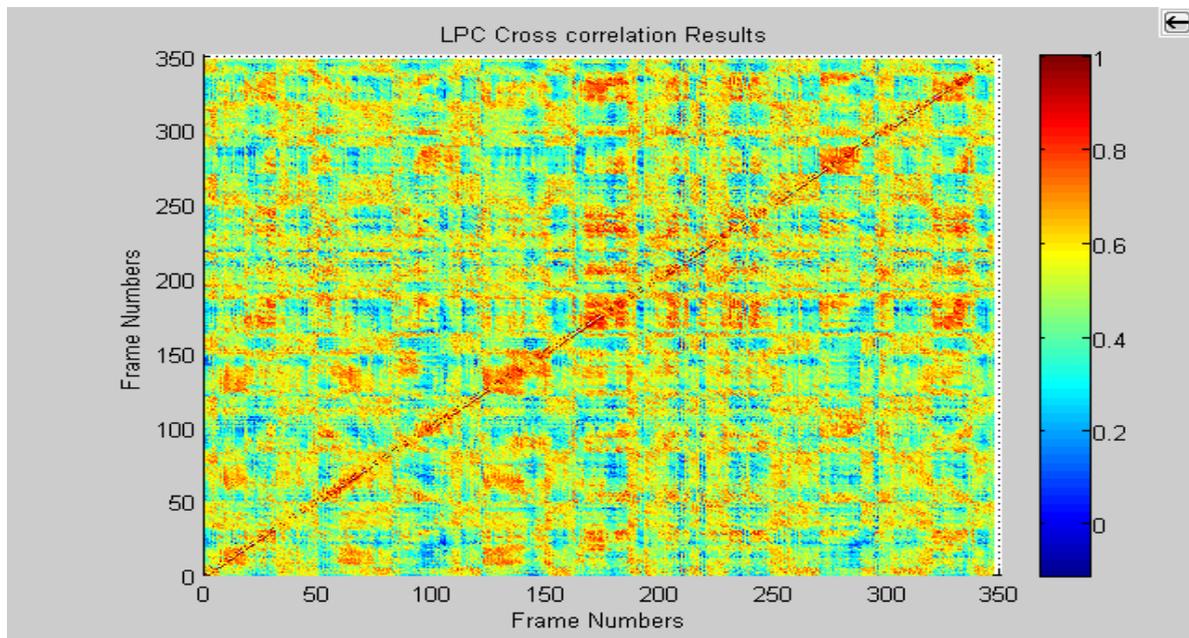


Figure 4.9 - LPC Cross Correlation Plot of speech signal of 5.541 seconds

In figure 4.9 the plot of the LPC spectrum cross correlation results are displayed. As explained earlier the number of coefficients per frame is 64 for the LPC. The digits are meant to be found in the diagonal middle. As can be seen, LPC spectrum does not provide a clear distinction to detect the respective digits.

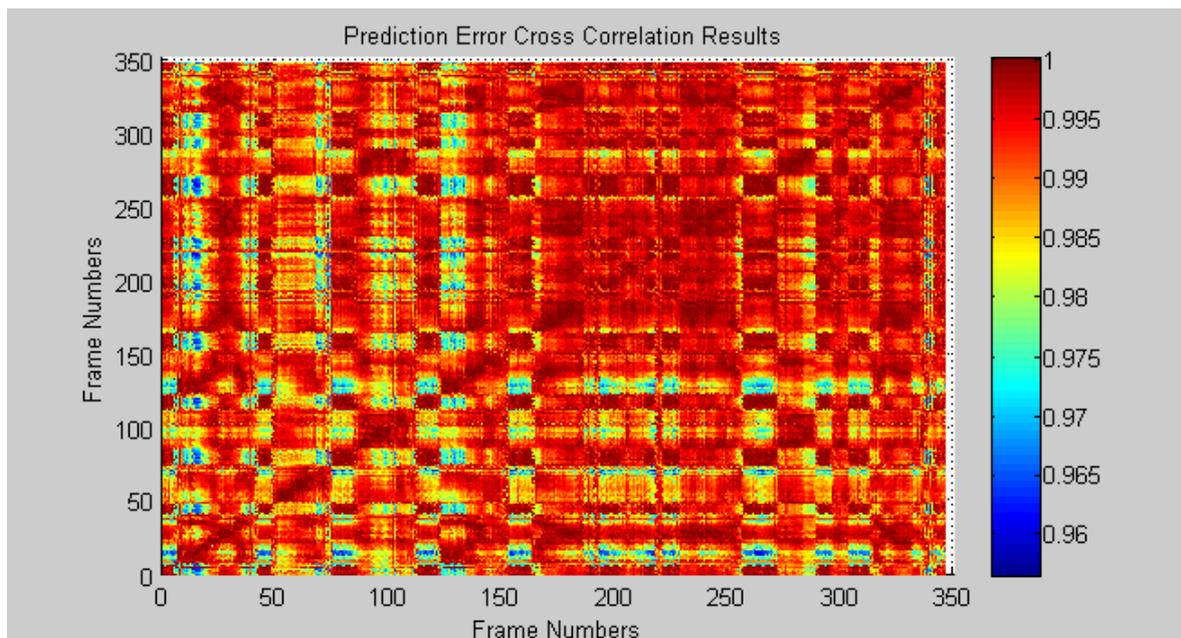


Figure 4.10 - PE Cross Correlation Plot of speech signal of 5.541 seconds

The prediction error cross correlation can be seen in figure 4.10. The matrix is seen to be very highly correlated for the prediction error. It is easier to spot uncorrelated frames than it is to distinctively recognise the digits. The evaluation of the results of the PE plot as well as the evaluation of the other speech features are explained in the next section.

b) Results Evaluation

The plots of the different speech features shows the results of the respective LCC algorithm for these features. It is clear from the DFT LCC plot in figure 4.8 that the DFT matrix shows more distinction in the plot for the respective digits as opposed to the other features. This is due to the nature of the DFT as speech features. The DFT spectrum of any signal is the frequency domain representation of that signal and as such this enriches the Gaussian properties of the speech signal. Statistically speaking, the speech signal is constituted by a different combination of speech, non-speech and noise, which are all different independent Gaussian random variables [64, 65, 66]. Due to this statistical assumption, the part of the signal containing speech will correlate with each other. On the other hand, the part of the speech which is background noise and silence will not correlate with each other because noise does not have a fixed distribution over the signal. This is the reason why the DFT spectrum outperforms the other features. Unfortunately, it is still not effective enough over the entire signal because the non-voiced speech phonemes tend to disappear into the background noise, as is the case for digits such as six.

As explained above, the Gaussian distribution is applicable to the frequency domain and not to the spectral domain. Consequently, there is little correlation in the plot of the LPC feature vectors because it is a set of Cepstral coefficients which are extracted based on the assumption of a linear speech producing mechanism. In contrast, the PE feature vectors have a very high level of correlation between all the frames because this is a statistical error calculated signal. This means that it has no specific inter frame distribution that is meaningful

enough for a direct correlation except in the cases of further higher order statistics as extracted by [63].

Table 4.6 gives the computation time for the LCC algorithm of the 3 features. In all cases, the application of this technique on a mobile device is not appropriate due to the computation time involved in LCC processing being far longer for processing 5.54125 seconds of speech data, and this is excluding the feature extraction time as well.

LCC Computation Times for Different Features for a 5.54125 seconds speech sample		
DFT	LPC	PE
35.23 secs	31.742 secs	34.512 secs

Table 4.6 - LCC computation times in seconds on HTC mobile device

4.3 Novel Voice Activity Detection Algorithms

Two novel Voice Activity Detection algorithms are proposed in this thesis. The first algorithm proposed is based on the application of K means clustering to the Log Prediction error feature. The second algorithm, which is an improvement to the first, is based on the linear cross correlation of the linear predictive coding feature and uses the same K-means to cluster the outcome of the correlation process. The clustering significantly improves the discrimination between the voiced frames and the unvoiced frames.

4.3.1 First novel VAD algorithm using K-means clustering on the Log PE

The first proposed algorithm is based on the Log PE feature. As explained and illustrated with figure 4.7, the Log PE of the speech signal has a relatively flat spectral envelope over the background noise and regions of silence. This resulted in the formulation of an algorithm that could distinctively cluster the speech signal into two clusters of speech and silence/background noise respectively. The concept of the K-means algorithm is explained in section 2.2.2. This experiment is achieved using the following steps.

a) Prediction Error Pre – processing and Extraction

The prediction error is extracted after windowing the speech signal. The windowing technique and window sized used are the same as used in section 4.2.1. After extracting the

windowed speech signal, the Levinson-Durbin algorithm is used to extract the LPC from which the residual (Prediction Error) is extracted. The order of the prediction error used for this experiment is 12, thus every windowed frame of 128 samples after the feature extraction process becomes a 12 Prediction error coefficient frame. These frames are then applied to the K-means algorithm.

<i>Prediction error extraction process</i>
<i>Take audio signal A</i>
<i>Compute W_A Using Equation 4.1</i>
<i>For Every W_A</i>
<i> Extract 12 PE coefficients</i>
<i>End</i>

Table 4.7 - Prediction Error extraction

Table 4.7 shows the pseudo representation of the extraction process for the prediction error. Since the Prediction error feature is extracted in the same process as the LPC coefficients there is no extra computational overheads. The procedure for the K-means algorithm and evaluation is given in the next step.

b) K-means clustering of Prediction Error

The concept of the K-means technique has been explained in chapter 2. However the application of the K-means technique to the prediction error is based on the spectral envelope of the PE profile of the entire speech signal which is the same one used for the PE LCC in section 4.2.5. In this novel algorithm, after extracting the Prediction error feature of the respective frames, the K-means algorithm then clusters the frames into two clusters of speech and silence. The original K-means algorithm is based on a simple minimum distance assignment principle using equation 2.8. The original K-means technique initially assigns random number as cluster centers. The distance between the feature vector in question and the respective cluster centers then determines where the feature vector is assigned. To avoid extra computation with regards to the assignment procedure, the choice of initial cluster center

was changed for this novel algorithm. The adoption of data specific initial center values were to adapt the K-means technique for mobile device compatible computation.

The choice of initial cluster centers was changed to the frame with the minimum mean and the frame with maximum mean within the speech sample. This helps reduce the computational time substantially as the number of iterations is reduced. A bias was also introduced to help lower the threshold in order to produce a more effective separation of the prediction error signal. This bias pushes the threshold for the algorithm below the median of the data. This allows enough of the speech data in the lower region to be captured. The equations for the bias calculation are given in equations 4.3 – 4.6

$$C_A = \left(\sum_{i=1}^N X_i \right) \div N \quad (4.3)$$

$$C_B = \left(\sum_{i=1}^N X_i \right) \div N \quad (4.4)$$

$$\mathbf{bias} = (C_A - C_B) * 0.33 \quad (4.5)$$

$$\mathbf{C_A} = \mathbf{C_B} + \mathbf{bias} \quad (4.6)$$

where X_i is the mean of the i^{th} frame within the cluster, C_A is the Centroid of the first cluster while C_B is the centroid of the second cluster and N the total number of frames within the cluster. The pseudo code implementation of the modified K-means is given in table 4.8.

<i>A Pseudo code Implementation Of the K-means Algorithm</i>
<i>Iteration = 1</i> <i>Compute Frame with Min and Max Mean of A</i> <i>Assign Min as centroid of C_A & Max as centroid of C_B</i> <i>for Iteration 2 -100</i> <i>for every CF applying bias in equation 4.6.</i> <i>if ED of $(C_A - CF) < (C_B - CF)$</i> <i> Assign to C_A</i> <i>else if ED of $(C_B - CF) < (C_A - CF)$</i> <i> Assign to C_B</i> <i>End</i> <i>Calculate C_A using equation. 4.3 and C_B using equation 4.4.</i> <i>If(new C_A and $C_B =$ Previous C_A and C_B) or (Iteration = 100) terminate</i>
<i>End</i>

Table 4.8 - K-means computation pseudo code

A = Audio Signal

C_A = Cluster A

C_B = Cluster B

CF = Correlated Frame

ED = Euclidean Distance

The K-means algorithm pseudo code shows the procedure adopted for the K means algorithm. The results of these experiments are given in section 4.3.2

c) Post K-means Evaluation

After the effective clustering of the respective frames into silence and speech clusters. The frame distance variable was introduced. As the speech cluster contains all the frames containing speech in an ascending order. The frame distance is used to separate the respective digits. Using the frame distance, a pause is searched for within the clusters. When a set minimum number of frames are missing between two adjacent frames, this is adjudged to be the pause/break in between the successive digits.

After segmentation of the signal using K-means and the frame distance variable, a manual verification method was used to determine the correct/incorrect detection of the respective digits. The detected regions of speech belonging to the respective digits were

played using the mobile device. This method of verification is the same as for the previous algorithms. Only detected digits that are fully recognised are counted in the results. Partially recognised digits are classified as incorrectly segmented/recognised as well as non-recognised digits.

4.3.2 PE K-means VAD algorithm results

a) Data Used

The speech data used for this experiment is also from the CSLU 2002 database. A total of 160 speech samples were used in this work. These are divided into 10 speakers, with each sample within the audio samples containing the digits 1 – 9 spoken isolated. All the speech samples have a sampling rate of 8 kHz and 16 bit encoding format.

b) Device Used

The HTC Desire smartphone used in these experiments had the following specifications:

- Processor: Single Core, 1000MHz
- System Memory: 576 MB RAM/ 512 MB ROM

This mobile device is classified a little below average in specifications as compared to other smartphones.

c) Results and Conclusions

Table 4.9 shows the results for the K-means PE VAD algorithm

Feature	Frame Distance	Recognized
Log Prediction Error(PE)	≥ 4	86.60%

Table 4.9 - PE K-means Recognition Results for CSLU 160 digits (10 speakers)

Table 4.10 then compares the K-means PE VAD results with the PE and STE Raw VAD algorithm results previously presented in section 4.2.4 table 4.4.

Algorithms		
K-means PE	Prediction Error	STE
86.60%	80%	73.30%

Table 4.10 - K-means PE vs. Raw PE vs. STE

It is clear from table 4.10 that the conventional voice activity detection methods such as the short time energy are not as effective at determining voice activity within speech signals as the K-means PE VAD algorithm in terms of classification accuracy. However, computation time is another key factor in deciding which VAD algorithm is best for on – device speech recognition. Table 4.11 contains the computation time for the different feature extraction algorithms. Table 4.12 then shows the computation time for the entire algorithm including feature extraction as well as the VAD algorithms.

Feature Extraction Computation Times in seconds		
DFT	LPC/PE	ZCC/STE
14.9648	3.4332	1.3128

Table 4.11: Computation Time of 44330 samples (5.54125 secs) speech signal

VAD Algorithm Computation Time in seconds		
DFT Cross Correlation (LCC)	Log PE K-means	STE
49.7546	6.684	2.6

Table 4.12 Feature Extraction + Algorithm computation times 44330 Samples (5.54125 secs)

Despite the apparently minimum computation time involved in computing the STE VAD algorithm, it should be noted that to perform any aspect of speech recognition post the VAD stage, either the LPC features or the DFT features have to be extracted in addition to the STE. Taking the LPC as features, this would incur the extraction time of the LPC in addition to the STE VAD algorithm (**2.6 + 3.4332**) resulting in a total VAD + feature extraction time of **6.0332** seconds. This is very close to the K-means PE computation which already includes the LPC extraction time. Meanwhile, the K-means PE has a VAD detection performance of **86.60%** compared to the STE feature algorithm which has a **73.30%** performance.

With regard to the computational time involved in extracting the DFT spectrum and then computing the cross correlation from it, it is clear that it takes considerably more time than the computational time involved in extracting the LPC features and computing the PE based K-means.

Overall the prediction error signal (LPC residual) with K-means clustering is seen to be the optimal choice for performing VAD on the mobile platform. The fact that the prediction error is a by - product of the LPC process means that it incurs no additional impact on the computational time. This provides a unique advantage. The prediction error can be used for voice activity detection while the linear predictive coding can then be used for speech recognition. The only thing needed is to identify the frames with Speech using the prediction error and then these frames can be chosen from the linear predictive coding signal and used for recognition. The second advantage of the LPC/PE is that a lower order version of these signals can be extracted without causing distortion to the signal as opposed to the DFT Spectrum.

The PE with K means clustering results have been shown to be faster and more effective method of identifying usable speech frames for recognition. However, how the choice of LPC features (as opposed to DFT or MFCC) affects the speech recognition process has not been ascertained at this point, this is done in the next chapter. The recognized frames in the K-means algorithm will be referred to as usable speech at the moment pending a proper evaluation of the frames in the recognition stage. Another unique advantage of the process is that it requires no training and each particular clustering or extraction process performed is always going to be speech signal specific. The dynamism of another signal or a different previously used signal will not affect the process as it treats each data as an individual data

set and calculates the K-means centroids and does the entire voice/silence clustering and processing on that particular data set.

4.3.3 Second novel VAD algorithm using LCC on the Standard Deviation of the LPC

The second novel VAD algorithm is based on the desire to improve mobile device speech recognition using an on-device implementation. In section 4.3.1 (table 4.6) the computation times for the standard LCC algorithms on the mobile device were presented. In this algorithm we present a two stage/process based VAD algorithm. This algorithm is broken into two stages so they can be independently implemented. This can provide a multipurpose outlook algorithm, where the user or researcher can use the first stage for further processing with their own algorithm whilst ignoring the second stage which is strictly our choice of implementation in this paper. The two stages involved are;

- Linear cross correlation using minimum standard deviation frames
- K-means clustering (Modified)

a) Linear Cross Correlation (LCC) Stage

The LCC concept was earlier applied to the DFT, LPC and PE speech features and the results have been presented in section 4.2.6. The direct application of the LCC to the LPC features were not as distinctive as the DFT results. The reasons for the failure of the direct application of the LPC was explained in the same section.

For this novel algorithm the LCC was applied to the LPC frames differently. After computation of the LPC features using the Levinson – Durbin method, the frames containing the LPC features of the entire signal are taken and equation 4.7 is used to compute the standard deviation (SD) of the respective frames. SD is a measure of noise and interference within a signal [105].

$$\sqrt{\sigma^2} = \frac{1}{N-1} \sum_{i=0}^{N-1} (x_i - \mu)^2 \quad (4.7)$$

where σ^2 is the variance, x_i is the signal and μ is the mean of the signal and N total number of signals.

The frame with the minimum standard deviation (MSD) is then chosen as the primary frame and this is used as the only candidate Frame for the LCC algorithm to correlate with every other frame including itself. This gives a LCC minimum standard deviation correlation value for every frame within the audio signal. The frames with the lowest correlation values are the frames with speech and vice versa. A pseudo code implementation is given in table 4.13.

<i>A Pseudo Implementation Of the LCC Algorithm</i>
<i>Break A into Frames of 128 samples</i>
<i>for every F</i>
<i>Compute LPC with Order 12 using equation 2.4.</i>
<i>end</i>
<i>for every LF</i>
<i>Compute the SD using equation 4.7.</i>
<i>end</i>
<i>Using MinSD as X in equation 4.2.</i>
<i>for every LF as Y in equation 4.2.</i>
<i>Compute LCC using equation 4.2.</i>
<i>end</i>

Table 4.13. Pseudo Implementation of LCC code

A = Audio Signal

F = Frame

LF = Frame containing LPC

SD = Standard Deviation

MSD = Minimum SD

Note. Only the MSD frame is used to compute the LCC values. Consequently, this is much faster than the original LCC methods which produced a matrix of LCC values for all frames against each other.

After computing the algorithm each frame has its corresponding correlation value with regards to the frame with the minimum SD. Figure 4.11 gives the initial output of the correlation algorithm, where the frames with speech are at the lower regions of the plot.

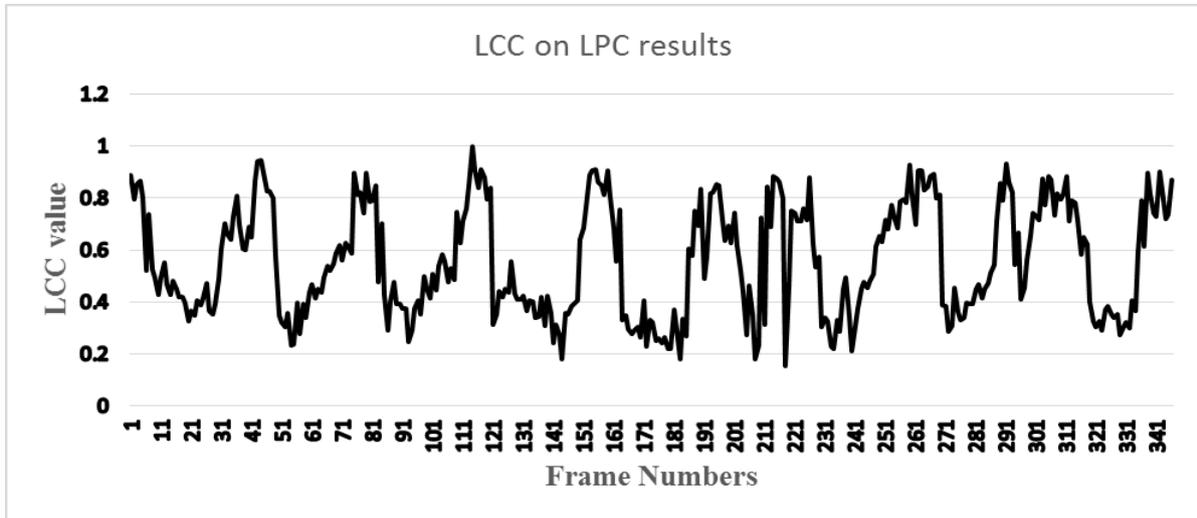


Figure 4.11 - LCC LPC using MSD plot

Reversing the vertical axis of the plot will give figure 4.12.

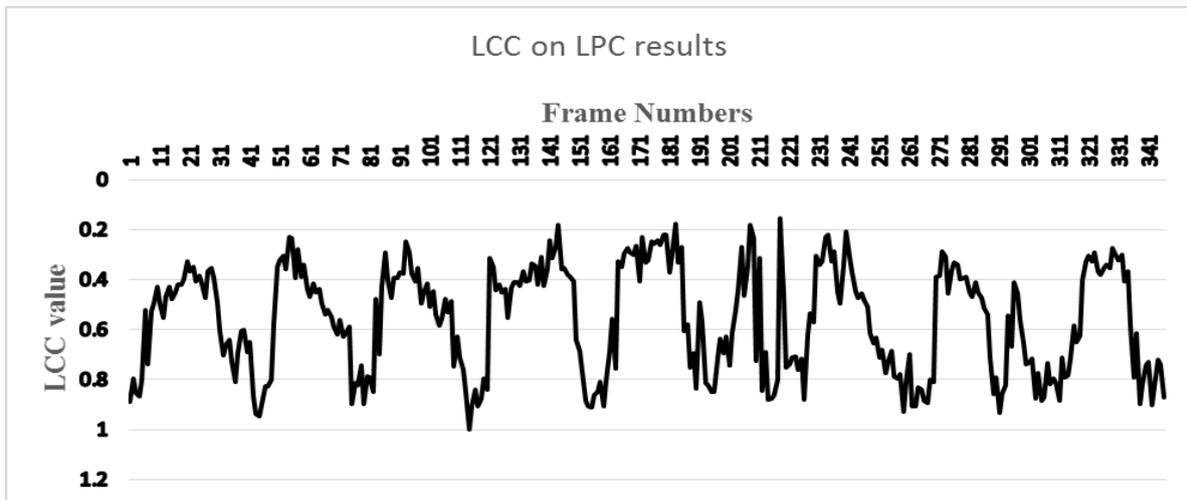


Figure 4.12 – Flipped figure 4.11

The reversed plot shows the position of the respective digits in the plot. This result can then be used for the stage 2 of our proposed system. However, this result could also be used for a different VAD algorithm to pick out the respective digits.

b) K-means Clustering Stage

The K-means algorithm was applied to the LCC MSD, correlation results are presented in figure 4.12, due to its fast computation time with respect to the other VAD algorithms. The K-means algorithm was used to cluster the LCC MSD correlation values into two clusters of speech and silence.

c) Modifications to K-Means

The standard K-means algorithm was modified with data specific choices for each of the correlated results in a similar way to the first novel algorithm presented in section 4.3.1. However, the difference here is that the frame with the minimum and maximum correlation values are taken as the initial centroids as opposed to the frames with the minimum and maximum mean. The pseudo code implementation of the Modified K-means is given in table 4.14.

<i>A Pseudo Implementation Of the K Means Algorithm</i>
<pre> for Iteration = 1 Compute Frame with Min and Max Correlation Value of A Assign Min as centroid of C_A & Max as centroid of C_B for every CF if ED of $(C_A - CF) < (C_B - CF)$ Assign to C_A else if ED of $(C_B - CF) < (C_A - CF)$ Assign to C_B end end Calculate C_A and C_B using equations 4.3 and 4.4. If(new C_A and $C_B =$ Previous C_A and C_B) or (Iteration = 100) terminate end </pre>

Table 4.14 - K-means clustering implementation pseudo

A = Audio Signal
C_A = Cluster A
C_B = Cluster B
CF = Correlated Frame
ED = Euclidean Distance

The output from the K-means clustering algorithm, is two clusters, one with speech frames and the second with silence. The cluster with speech is used for further processing in order to extract the respective digits. The cluster with the silence region is ignored.

The maximum standard deviation or rather the frame with the highest standard deviation was also used as a candidate for the correlation process. This also yielded interesting results, but this was not as distinctive as that of the minimum standard deviation. In our experiments and the results we present, both the minimum standard deviation frame and the next minimum standard deviation frame were used. The next minimum proved even better in some cases as compared to the minimum.

d) Experimental Data

A total of 20 different speakers with the digits 1 – 9 (total of 180 digits) were used in total for these algorithms. As before, these samples are from the same Centre for Spoken Language Understanding (CSLU2002) database. The first 10 were used as a guide set to modify and apply the different variable adjustments to optimise the system parameters. The second set of 10 audio samples was used to test the algorithm against the first proposed algorithm in section 4.3.1. The results are given in table 4.15.

A final stage was added to the algorithm to identify and segment the respective digits. These digits were grouped using a decision algorithm that uses 2 variables to make its decision. Only the cluster with speech is passed on to this algorithm to try and find out where the respective digits are. These two variables are;

- **Frame Distance:** This variable is used to check for the pauses between the respective digits. There are 9 digits with pauses in between them. This variable is the distance between the frames containing the respective digits. This can be adjusted and there is no definite size to this particular variable as the speed of speech cannot be controlled because it depends completely on the individual speaking. However, attempts were made to try and generalize this as will be shown in the results.
- **Inter Cluster Distance:** This particular variable is used to complement the effort of the frame distance variable in the grouping of the digits. There are cases where there is a small silence within the utterance of a digit. An example of this is the digit 6. Looking at figure 4.11 it can be seen that for the digit 6 there is a slight break between the “SI” part of it and the “X” part. Using the first variable, this break could cause the two speech sections to be grouped into two different groups as separate digits. The inter frame distance checks every respective group to see if it meets a set minimum. If it does not attain the set minimum, this then tries to assign it to the group closest to it using the Euclidean Distance measure.

Different adjustments to these variables were used to come up with an acceptable optimal number. A number of experiments were conducted in this regard and then the same 90 speech samples, used with the previous work, were tested to compare and see if there was any improvement.

After the adjustments of the respective variables, the algorithm developed is in the form of a mobile application. This was deployed on the same mobile phone (HTC) and was used to automatically test and then manually verify the correct segmentation of the respective digits. Only the set of digits that were fully recognised were recorded as recognised. The digits that were broken or split by the algorithm were ignored in the computation of the final result numbers.

4.3.4 Linear Predictive Coding K-means on Standard Deviation Results

Standard Deviation	Frame Distance	Inter Frame Distance	Recognized Male	Recognized Female
Minimum	≥ 2	≤ 6	77.70%	91.10%
Next Minimum	≥ 2	≤ 6	71.10%	95.50%
Minimum	≥ 2	≤ 7	77.70%	93.30%
Next Minimum	≥ 2	≤ 7	66.60%	93.30%
Minimum	≥ 2	≤ 8	75.50%	91.10%
Next Minimum	≥ 2	≤ 8	71.10%	93.30%
Minimum	≥ 2	≤ 9	73.30%	93.30%
Next Minimum	≥ 2	≤ 9	68.80%	95.50%
Minimum	≥ 2	≤ 10	75.50%	93.30%
Next Minimum	≥ 2	≤ 10	66.60%	97.70%
Minimum	≥ 3	≤ 6	75.50%	86.60%
Next Minimum	≥ 3	≤ 6	71.10%	95.50%
Minimum	≥ 3	≤ 7	75.50%	86.60%
Next Minimum	≥ 3	≤ 7	68.80%	95.50%
Minimum	≥ 3	≤ 8	80%	88.80%
Next Minimum	≥ 3	≤ 8	71.10%	93.30%
Minimum	≥ 3	≤ 9	82.20%	88.80%
Next Minimum	≥ 3	≤ 9	71.10%	95.50%
Minimum	≥ 3	≤ 10	77.70%	88.80%
Next Minimum	≥ 3	≤ 10	68.80%	97.70%
Minimum	≥ 4	≤ 6	82.20%	84.40%
Next Minimum	≥ 4	≤ 6	62.20%	97.70%
Minimum	≥ 4	≤ 7	80%	82.20%
Next Minimum	≥ 4	≤ 7	57.70%	97.70%
Minimum	≥ 4	≤ 8	84.40%	84.40%
Next Minimum	≥ 4	≤ 8	62.20%	97.70%
Minimum	≥ 4	≤ 9	80.00%	82.20%
Next Minimum	≥ 4	≤ 9	62.20%	93.30%
Minimum	≥ 4	≤ 10	75.50%	82.20%
Next Minimum	≥ 4	≤ 10	53.30%	93.30%

Table 4.15 - 10 Initial Samples for parameter optimisation

Standard Deviation	Frame Distance	Inter Frame Distance	Recognized
Minimum	≥ 2	≤ 6	76.60%
Next Minimum	≥ 2	≤ 6	80%
Minimum	≥ 2	≤ 10	90%
Next Minimum	≥ 2	≤ 10	86.60%
Minimum	≥ 3	≤ 10	85.50%
Next Minimum	≥ 3	≤ 10	85.50%
Minimum	≥ 4	≤ 6	87.70%
Next Minimum	≥ 4	≤ 6	83.30%
Minimum	≥ 4	≤ 8	87.70%
Next Minimum	≥ 4	≤ 8	83.30%

Table 4.16 - 10 test samples from the algorithm in section 4.3.1

It can be seen from the results that this algorithm and detection outperforms the voice activity detection algorithm of section 4.3.1. The optimal performance of the K-means PE VAD algorithm in that section was **86.6%** compared to the **90%** result for the LCC MSD VAD algorithm on the same data. The results presented in tables 4.15 and 4.16 can be further broken down on the basis of gender. The best performance of this algorithm when comparing only female speakers is **97.7%** whilst the best male speaker performance is **84.4%**. The disparity in these performance figures is due to the average sound of male speakers, it can be noted that there is a lot of vibration in the male speech as opposed to the female speech which has a smoother pitch. A t-test was conducted and this is presented in Appendix 1.

Algorithm Computation Time Of DFT LCC Vs LPC	
DFT LCC	LPC MSD LCC
35.23 secs	2.76 secs

Table 4.17 - Linear Cross Correlation Computation Times

In table 4.17 the computation time for the respective LCC VAD algorithms is given. The LCC stage of this experiment only takes **2.76 secs** to process as compared to the

original DFT LCC process in the previous section which took **35.231 secs** on the same speech data of 5.54125 secs. The primary reason for this difference is that only 1 frame was used to create the LCC MSD values that were then clustered by the K-means algorithm. The novel method of extracting the LCC MSD value of the LPC frames shows a significantly faster processing time on the same mobile device and as such makes this technique applicable to mobile devices. The time constraint can be regarded as one of the primary foci in designing on-device mobile based speech recognition systems.

The task of assigning the frame distance and inter frame distance is very tricky. However, for on-device mobile speech recognition these variables can prove very important as they allow a degree of customization to the particular user of that mobile phone. This will improve the problem of generalization with regards to speech recognition and as such it helps to concentrate a system down to the user.

It is worthy of note that both the minimum and the next minimum standard deviation cannot be ignored for implementation purposes. The results in table 4.15 shows that in the female cases, most especially, the next minimum MSD values outperform the minimum MSD values. It could be that this is an advantage to the implementation of any given user, hence the choice of variables can be implemented in a dynamic approach to try and adjust them to a specific user.

4.4 Summary

This chapter presents the work undertaken in this thesis with regards to the different VAD algorithms implemented to aid the speech recognition process on the mobile device. In this section of the summary, the results of the different results are compiled together. See table 4.18 and 4.19

Computation Time of The Different VAD Algorithms for 5.54125 seconds of speech in secs						
STE	Raw PE	DFT LCC	LPC LCC	PE LCC	K-means PE	K-means LPC
2.6	4.753	49.754	35.1752	37.945	6.684	6.1932

Table 4.18 - Computation time for all VAD algorithms including feature extraction times in seconds

Algorithm Performance %			
STE	Raw PE	K-means PE	K-means LPC MSD
73.3	80	86.6	90

Table 4.19 – Final performance of all VAD algorithms

The time for the respective algorithms is given in table 4.18. As explained much earlier, despite having the minimum computation time, the STE algorithm, its performance of **73.30%** compared to the other techniques shows it to be the least viable for VAD purposes. This is in addition to it not being able to be adopted as a single feature vector for VAD and speech recognition. The raw PE features did perform better than the STE which is why the feature was selected for further processing. This feature has the advantage that it is computed from the same process as the LPC. Consequently, it seemed favourable to adopt it as the same extracted feature set could be used for both VAD and speech recognition. Further work which applied the standard LCC on these features, including the LPC and DFT, showed that the performance of the LCC with these features was not distinctive enough to be adopted for VAD. In addition, the processing time involved for all 3 features do not make it viable on a mobile device.

The two novel algorithms, which were submitted to AI2014 and AI2015 shows a significant improvement to the previous algorithms. Though the second one was a further improvement to the first proposed algorithm, the features used are different. The last novel algorithm shows the best VAD detection performance over the previous set of algorithms as

shown in table 4.19. In addition, the processing time for this algorithm is also close to real-time.

As can be seen from the time plot in table 4.18, for a **5.541** seconds audio signal with **44330** samples, the final proposed algorithm takes only **6.19** secs to process. This is equivalent to **1.11** seconds to process each second of speech data on the mobile device used. The mobile device used for the VAD algorithms is the HTC mobile device and all the computation time presented are for this mobile device. The reason for using this particular mobile device as opposed to the more powerful Samsung Note 3 is because a system that is adapted to a less powerful mobile device will perform much quicker on the powerful device.

As a consequence of the work done in this chapter, the best performing techniques are inserted into the frame work to complete the part of the framework relating to this chapter.

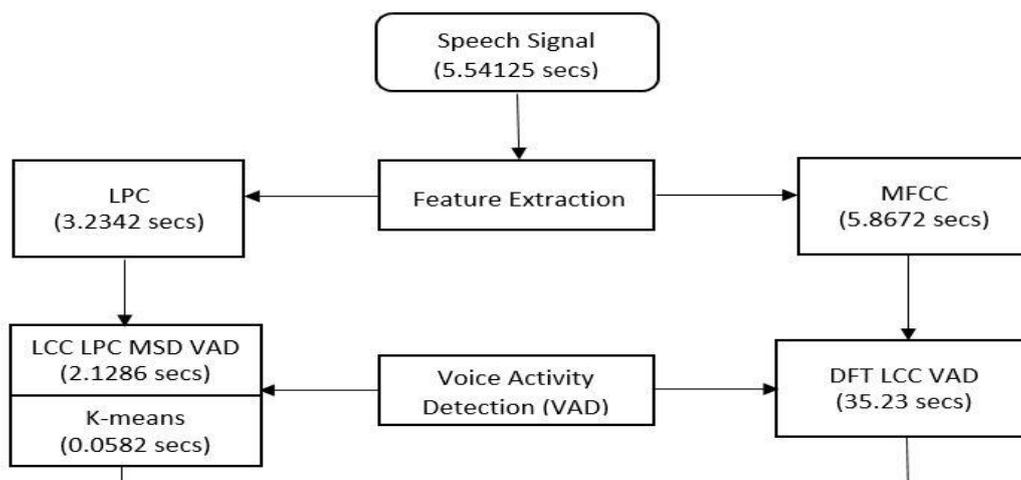


Figure 4.13 Framework Outcome of Chapter 4

The outcome of the respective VAD algorithms is that they produce a concatenated set of audio signals consisting of the digits with each having variable lengths (time). This variability in the speech signals is a fundamental characteristic of speech, where the same word being spoken by different people inherently has a different number of samples. More so, even the same speaker uttering the same word 10 times will not have all the 10 utterances being the

same length. To cope with this variable length issue, the speech recognition systems adopted for this work, needs to be able to accommodate this variability in order to correctly identify the respective words or phonemes within the words.

In the next chapter a speech recognition technique is identified that can adopt a variable length audio signal with minimum further classification or segmentation in order to meet the limited processing power of the mobile device. In addition, to minimise computation time on the mobile device, the different speech recognition techniques adopted as the recognizers in the next chapter are strictly based on the adoption of a single technique that can use the output of the VAD algorithms without further modification to the feature subset.

Chapter Five

Mobile Speech Recognition (SR)

5. Introduction

This chapter presents a set of experiments conducted in order to use the outputs of the previous chapter experiments as input into a mobile speech recognition system. The different techniques adopted here are guided by the fundamental principle of minimising processing in order to accommodate mobile devices lack of computational power and memory. The work done in this chapter constitutes construction of the latter part of the framework developed in the course of this research.

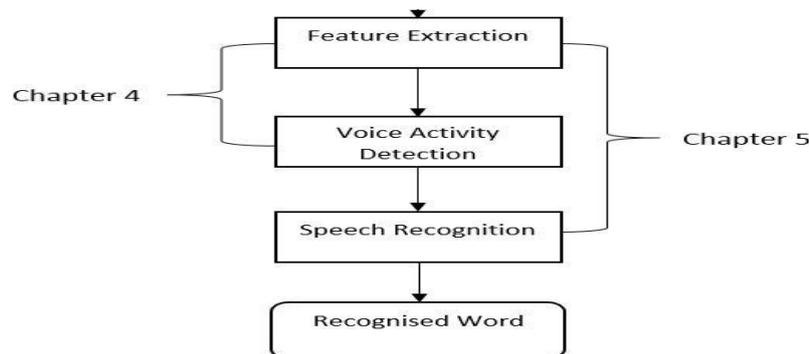


Figure 5.1 Latter Part of Framework

This chapter is also divided into subsections with each describing the application of the different techniques adopted. Section 5.1 presents a brief reintroduction into speech recognition and the choices to be made for mobile device speech recognition. Section 5.2 then presents the databases used for these experiments and how they are extracted. Section 5.3 presents the speech signal and how it is evaluated for the extraction process and reasons why. Sections 5.4 and 5.5 then present an Echo State speech recognition system with experimental results. Section 5.6 presents a proposed Novel Dynamic MLP speech recogniser

with the architecture, algorithm and results. Whilst section 5.7 is the comparison with a hidden Markov model equivalent. Sections 5.8 and 5.9 then provide a conclusion and summary of the chapter.

5.1. Speech Recognition

Speech recognition is the phenomenon by which a device understands and translates human utterances into machine understandable format for further processing [106]. For mobile platforms, limited as they are with relatively low speed processing units and small data storage capacity, it is necessary to implement an effective voice activity detection algorithm and choose an efficient speech recognizer in order to achieve viable on-device continuous speech recognition. The work with regards to Voice Activity Detection has been presented in the previous chapter. The outcome from this work was that a VAD algorithm using the linear predictive cepstral coefficients as feature vectors gave the optimum performance. The output of that final algorithm is therefore the determinant for the choice of speech recognizer to be used for the mobile device. Hence LPC feature vectors are selected as the preferred input format for the speech recognition systems investigated in this work.

However, different speech recognition systems also adopt the Mel frequency cepstral coefficients (MFCC) as the dominant feature [107]. Some also adopt its second order derivative in time to better reflect the information contained in the speech signals [24]. As stated by [107, 24] and quite a lot of work in the literature, MFCC are the preferred choice of features for speech recognition as opposed to the linear predictive cepstral coefficients (LPC). It is argued that this is because the LPC is calculated based on the assumption that the speech production mechanism is a linear model [108], whilst the MFCC features are based on the human hearing perception of speech. As a result the Mel scale is linearly distributed within the frequency range of 0 – 1000Hz and logarithmically distributed above 1000Hz [33], to

better reflect the changes within the speech signal. Consequently, this work also compares the use of MFCC feature vectors as input to the same speech recognition systems.

There are three different implementations of speech recognition used in this thesis, and by extension this chapter. These are based on three different classification techniques for the recognition process. These techniques are:

- i) Echo State Networks
- ii) Feed Forward Multi-layer Perceptron
- iii) Hidden Markov Models

The aforementioned techniques all require different pre-processing and data preparations; as explained in their respective sections.

5.2. Databases

For all the speech recognition experiments the two databases presented in chapter 2 were used. The first database, which is from the Centre for Spoken Language and Understanding (CSLU), has a total of 20 speakers. Each speaker has a total of 16 samples containing the digits 1 – 9. These 1 – 9 digits were all split into their respective individual samples using the manual border process described in chapter 4. This resulted in a total of 2880 digits. These digits were further split into a training dataset of 2520 and a testing dataset of 360. This split was done randomly and as such there is a cross-over of some speakers in the training and testing datasets. Due to this cross over this dataset was classified as a speaker dependent approach.

The second database is the TIDIGITS database which was already divided into a training and test set. With the training dataset having 163 speakers (55 males, 57 females, 25 boys and 26 girls) and the testing data also having 163 speakers (56 males, 57 females, 25 boys and 25

girls). For the training dataset of 163 speakers, a total of 2 samples per digit per speaker were selected for the training and this also applied to the testing dataset. This resulted in a total of 2934 digits for training and 2934 digits for testing. As stated in chapter 2 the two datasets are completely different. Further details about encoding and sampling rate of these two databases can be found in chapter 2 section 2.4.

5.3. Application of Speech Signal to Speech Recognizers

The application of the speech signal to the speech recognizers is an important part of the speech recognition process. This stage defines how the data is presented to the classifier network and as such decides the classifier architecture to be used for the respective recognition process. Based on the results from the previous chapter and experiments, it was decided that the recognisers chosen should be able to handle the output of the VAD algorithm with minimum further processing. However, this choice of input format does not contradict the general notion of signal input presentation choices for which there are two ways of presenting a speech signal to a recognizer. These two approaches are either a phonetic approach or a whole word approach. Both approaches suffer from the need to address one of the fundamental characteristics of speech signals; the issue of speech length variability.

The speech length variability issue is the difference in the length of speech with regards to spoken words [109]. As per the example given in the previous chapter, a speaker uttering the same digit 10 times is not guaranteed to produce samples of the same length for all the 10 utterances.

The first approach is the Phonetic approach which involves breaking words down into their respective phonemes. Features from these phonemes are then presented to the recognizer and the recognised phonemes subsequently synthesised to form a word. For example, taking the digit 1, this can be broken down into the three phonemes “W”, “A” and “N”. The task of the speech recogniser is then to classify the respective phonemes as “W”,

“A” and “N” and sum them together to form the word at the output of the recogniser. This approach assumes that the speech signal can be modelled as a Left – Right signal [110]. The respective frames of the audio signal represent one of these phonemes during a short time period and then they transition into the next phoneme and so on until the end of the word/digit. These frames are fixed in length but their number varies according to the length of the word, this way the length variability is addressed by recognising the frames and synthesising the word after. Identifying the phonemes is an exhaustive task.

The second approach is a word level approach. In this approach the underlying phonetic structure is not considered. The entire digit/word is presented to the recognizer without trying to identify the respective regions containing the phonemes. The recognizer is then left to make the decision over the entire speech waveform. One of the main problems associated with this approach is dealing with the problem of length variability. One attempt to solve this issue is by using a pre-processing segmentation technique to deprecate the speech signal into the right length and then present it to the recognizer. Another approach is to develop a fixed length system by padding the excess input space of a given default length expected by the recognizer. The experiments conducted in this chapter investigate both phoneme and word approaches and present a novel approach to getting around the speech length variability issue in whole word recognition systems.

5.4. Echo State Network Recognizer

An echo state network was investigated to see if it can be used as a phoneme based speech recognizer using the output of the Voice activity detection algorithms. As stated in section 5.2, the dataset was pre-processed into frames of data prior to training the network. However, to train the ESN a further pre-processing is necessary in order to provide the phonetic labelling required by the supervised learning system. The steps involved in the application of this techniques are given below.

a) Data Preparation

The data preparation for the echo state network is an important part of the recognition process. This is where the phonetic labels are provided for use during training. An echo state network is a form of supervised recurrent neural network. One of the underlying principle and reason behind the development of recurrent neural networks is to deal with time series related tasks. This is where the output of a process is dependent on different previous steps to give the desired output. This is particularly relevant where the phonetic approach comes into place.

To train a supervised recurrent neural network, the different samples representing the different digits need to be pre-processed to label the different frames within them with their respective phoneme labels. In order to automate this process a Kohonen self-organising map (SOM) was used to achieve the phonetic labelling. The entire data representing a particular digit, for example all the data representing the digit 1 were presented to the SOM network in order to achieve a labelling good enough to be used as the teacher for the ESN network. Prior to applying the SOM to achieve the phoneme labelling, the respective digits were manually defined as containing a given number of phonemes.

Digit	Phonemes	No. of Phonemes
1	W, A & N	3
2	T & UW	2
3	TH, R & E	3
4	F, O & R	3
5	F, AY & V	3
6	S, IY & X	3
7	S, EH, V, I & N	5
8	AY & T	2
9	N, AY & N	2

Table 5.1 - Digit Phoneme labels

The different phonetic units within the digits were classified using table 5.1. These phoneme number values were then used by the SOM to automatically create a phonetic label for the respective frames within the speech databases used.

b) Self-Organising Map Preparations

A self-organising map was used in order to label the phonetic units within the respective digits. The labelled output of this experiment was then used as the target output for the frames being fed to the echo state network. For this SOM experiment, MATLAB was used for fast processing of the speech signal. The SOM is not meant to be implemented on the mobile phone because this training is assumed to be a once-only process that can be performed off-line. Once trained, the trained ESN system can then be deployed on the mobile device.

For the digit 1, a SOM is designed with input size 12, this being the total number of features within a frame for both the LPC and MFCC features. The SOM is a 1 dimensional map with 3 output neurons. These neurons are the digit 1 classifier output for the respective frames. The SOM uses a Batch weight learning procedure, where the weight changes are calculated in such a way that each neuron's new weight vector is the weighted average of the input vectors that the neuron and neurons in its neighbourhood responded to with an output of 1. A maximum iteration of 200 is applied for the procedure. The output of this process is that every frame is assigned a label indicating which neuron it belongs to. In this research this reflects to the phonemes. The 3 outputs are adjudged to be the 3 phonemes that are contained in the digit 1.

This procedure was adopted for all the digits, applying the same steps for both the training and testing data of that digit to attain a synergy between the two datasets. For example, the digit 7 has more phonemes than the rest, so it uses a five neuron output SOM to classify the respective phonemes. This training pre-processing principle was applied to both databases. However, an additional 1 extra phoneme unit was attached to the TIDIGITS database because the audio signals in this database contained a lot of background silence. To

accommodate these regions of background silence, an extra silence neuron was added to all the digits in the TIDIGIT experiments.

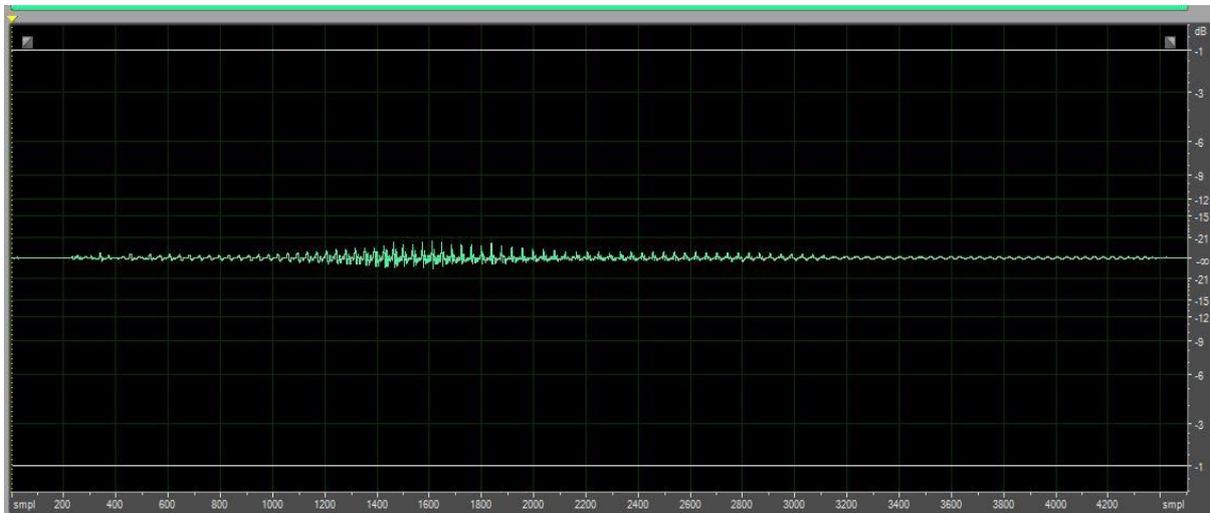


Figure 5.2 - A plot of digit 1 from CSLU

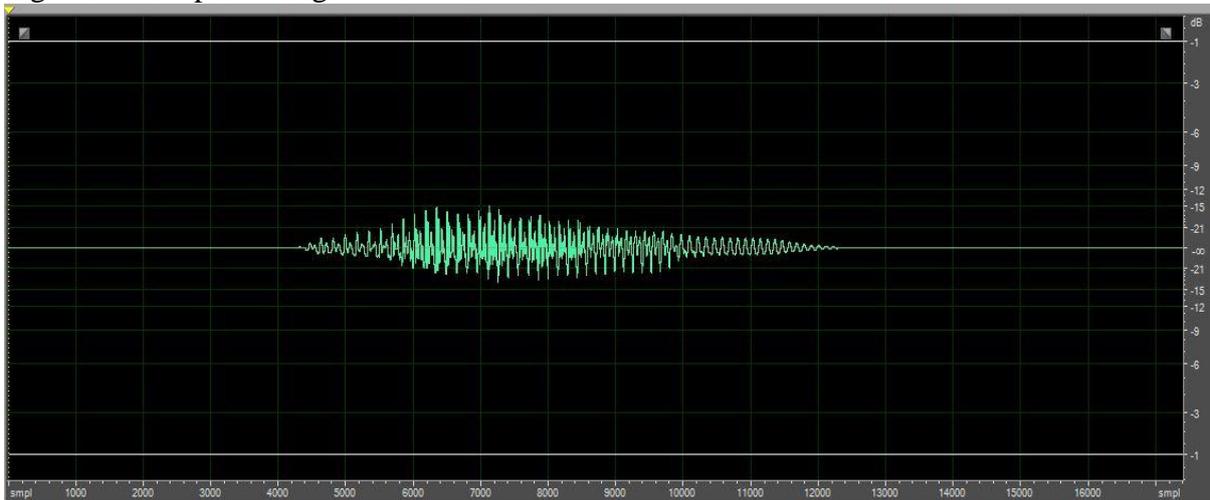


Figure 5.3 - A plot of digit 1 from TIDIGITS

The amount of silence region within the audio signal of a CSLU database sample can be seen in figure 5.2 while figure 5.3 shows a plot of a digit 1 sample from the TIDIGITS. This shows that the TIDIGITS samples contain a substantial amount of silence either side of the spoken digit and this is applicable across the entire database. The TIDIGITS is used as supplied without trying to reduce the amount of background silence in order to be able to effectively allow comparison and application of the techniques presented.

The SOM outputs were used to manually identify the phoneme tags and synthesize all the phonemes together whilst counting overlaps as the same phoneme. This led to the final

phoneme number of 17 for the respective digits. An example of the overlap can be found in the two digits of four and five. Both these digits have the “F” phoneme and as such labelling them as different phonemes for the network can be counter-productive. Avoiding this disparity led to the 17 phonemes as stated earlier.

After the respective phonetic transcription of the frames has been carried out, the CSLU database has a total of 63501 frames for the entire training data. These frames are transcribed to represent one of the 17 phonemes. For the TIDIGITS dataset, there is a total of 183893 frames representing 18 phonemes.

c) **Echo State Network (ESN) Architecture**

As originally stated, the work in this thesis is based on a mobile device implementation of speech recognition and every decision with regards to the experiments conducted is tailored towards achieving an effective on-device implementation of speech recognition. Consequently, the architecture adopted for the ESN is based on this decision. As a matter of fact, the choice of the ESN is also based on this reason. As explained earlier ESN’s present a different approach to recurrent neural networks because they accord the researcher the opportunity to only train one set of weights (the output weights) and at the same time use a linear regression method of training these weights. For this reason they can be easily trained in comparison to other recurrent neural networks. This does not, however, excuse them from the memory size and computational demand problem of using recurrences.

For the same reason, the architecture adopted here does not necessarily present the optimum network architecture for achieving the maximum classification performance but does present an effective network that conforms to the computational power and memory of the mobile device. The design decisions taken produced a feasible echo state network architecture that can take the output of the VAD algorithms and at the same time be

compatible with mobile device used. The audio data presented to the network was the respective digits one at a time. With each digit comprising of frames containing the 12 features within them. Each digit in both datasets contains a different number of frames.

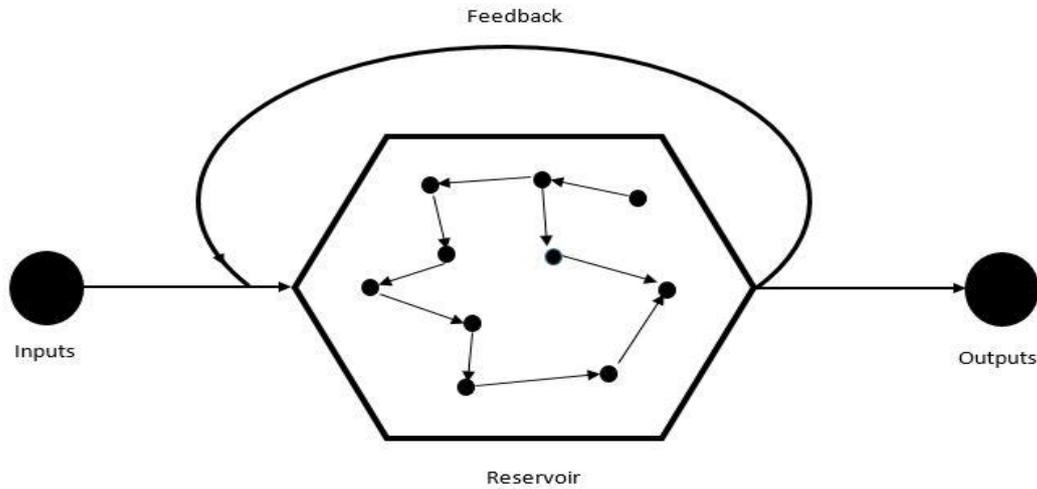


Figure 5.4 - ESN Architecture

The transfer functions used in each layer are given in equations 5.1 (reservoir) and 5.2 (output).

$$h(n + 1) = f(W^{in}i(n + 1) + W^{h(n)}) \quad (5.1)$$

where h is a unit in the hidden layer, f is the activation function, W^{in} is the weight connecting the input and reservoir units, i is the input to the network and $W^{h(n)}$ is the weight of the previous hidden unit.

$$o = f(W^{out}(i(n + 1), h(n + 1))) \quad (5.2)$$

where o is the output of the network, f is the same activation function and W^{out} is the weight connecting the reservoir unit and output unit, $h(n+1)$ is the reservoir output given by equation 5.1.

The ESN architecture uses 12 input nodes with adjustable connection between the inputs and reservoir. The variable numbers of reservoir units also have adjustable connectivity to

themselves and the output units based on the particular choice of output decided. The feedback connections provide the reservoir with memory of all previous patterns. This network is based on the original concept suggested by [111] and the adoption of a single feedback structure for the network creates a transitional behaviour between the frames presented to the network representing the left to right ordering of the speech signals [110]. This memory allows a connection to exist between the preceding frames and the current frame that allows the neural network to automatically adapt to the speech signal in such a way that it does not become device memory intensive. Paper [111] also recommends avoiding feedback from the output to the reservoir for time series problems.

There are several different parameters that can be modified for this type of network. These include the spectral radius, weight range, input connectivity and the interconnectivity of the reservoirs. The spectral radius is the most important parameter because it defines the condition that guarantees the stability of the network. It is a property of the reservoir and according to [111] it has to be < 1 to ensure stability. The reservoir weights are generated randomly within a particular weight range. These parameters all play a crucial role in the performance of the network for different tasks. There were different formulations of the number of output in order to capture different phoneme calculations using table 5.1 as guide.

The training procedure adopted for the output weights of the network is the pseudo-inverse training algorithm. The pseudo inverse of the reservoir activated units are used to train the output weights [46]. However, use of the pseudo inverse algorithm places heavy demands on the device memory, as it is required to store reservoir history during training. Despite the memory challenge, the pseudo inverse weight update algorithm is used because it offers a one-shot training of the output weights and can be performed off-device in a once only process. This would be particularly useful for larger vocabulary datasets [46]. The concept of the pseudo inverse training algorithm is to concatenate the input and hidden nodes

into a single column matrix. This is done for all the respective time steps that constitute the output, although for the case of the architecture adopted here, there is only one time step. One of the reasons for adopting the single time step and this particular architecture was to limit the memory intensive calculations and storage of previous time steps within the network. To calculate the weight updates during training, equation 5.3 is used.

$$W^{out} = O^{desired} X^T (XX^T)^{-1} \quad (5.3)$$

where $O^{desired} X^T$ is a matrix containing the targets of the network with regards to all the stored concatenated values of X and (XX^T) is a second matrix containing all the concatenated values with regards to previous states. W^{out} is the same as for equation 5.2.

Due to the nature of echo state networks, the only set of weights to be trained are the output weights. As the name of the pseudo inverse technique implies, the second matrix is inverted and calculations are done. What makes this computation so memory intensive is that, the respective matrices are collected over time and over the entire range of the training data. This is very memory intensive process, especially for very large datasets, and, as such this requires training to be done offline.

d) Echo State Network Algorithm

As shown in table 5.1, the ESN speech recognition algorithm is based on 17 output units for the CSLU and 18 output units for the TIDIGITS databases. The network takes as input one frame of a speech sample at a time and produces, as output, the classification of that frame while storing the reservoir units to be fed to the next frame. It continues until it gets to the end of that digit sequence. For the testing part of the network, a digit sequence is only accepted to be recognised when the network recognises at least **70%** of the total number of frames within a digit fed into the network for recognition. This is to allow it cover at least a wide range of phonemes.

<i>Echo State Network Forward Pass</i>
<i>For every $D \in N$</i> <i>For every $F \in D$</i> <i>Calculate H using equation 5.1</i> <i>$Net1 = f(H)$</i> <i>save $Net1$ as PH</i> <i>Propagate to the Output</i> <i>Calculate O using equation 5.2</i> <i>$Net2 = f(O)$</i> <i>End</i>
<i>End</i>

Table 5.2 - Echo State Network Forward Pass

N = Total Number of Digits

D = Digit

F = Frame

H = Hidden unit

PH = Previous Hidden

NF = New Frame

Win = Input Weight

The forward pass through the network is summarised in table 5.2. During training this pass is applied to every digit sequence until the entire training data is exhausted. After the forward pass is calculated for all the digits within the training data. The weights of the network are updated. The respective reservoir states are stored during the forward pass procedure. The stored reservoir state are then passed into the training part of the algorithm where the weights are updated using the pseudo inverse procedure.

This same algorithm was adopted when testing the network, although in that case, the output of the network was used to label each previously unseen frame. After all frames in the test sequence has been processed, the label sequence is examined to see if it corresponds to a given digit sequence. The next section describes the different experimentations performed to optimise the spectral radius, input-connectivity, interconnectivity and weight range using the dataset presented.

5.5. Echo State Network Experiments and Results

The different experiments conducted with regards to the echo state network are first classified based on the output size chosen. Then for each output size, the different parameters of spectral radius, input connectivity and inter connectivity are all modified.

a) Output Experiments

The number of outputs in the first set of experiments is based on table 5.1 where the respective number of unique phonemes is adjudged to be 17 over the entire digit collection of 1 – 9. The adjustments of the parameters for the network as stated is based on manual adjustments. The respective training experiments were conducted on a desktop computer with the sole intention of passing the trained ESN weights unto the mobile device for testing purposes. Computation time is an important factor for mobile device use and this time factor is affected by the mobile processor and memory resources.

The different experiments conducted are presented in terms of the global parameters to be modified. The following sections define these parameters and explain the experiments conducted for this purpose. There is no set globally accepted values for this parameters. As argued by [46] the only way to go about modifying these parameters is to do it one at a time. This is because the network configuration also affects how these parameters work in unison with each other.

b) Parameter Modifications

- Spectral radius: This is used to normalize or scale the reservoir weights of the network. It is defined as the maximum eigenvalues of the reservoir weights. As originally proposed this has to be less than 1 (usually between 0 and 1) to maintain the echo state property of the network as well as the stability of the network. As recommended by [112] this should be modified in successive multiples of 0.2

between 0.8 and 0.2. However, a higher number was adopted to cover the range better.

- **Input Connectivity:** This is the connectivity of the input to the reservoir i.e. the weights connecting the input layer to the reservoir layer. This is 1 for full connectivity and 0 for no connectivity. Different ranges of it determine the amount of connection that exists between the input and reservoir layers of the network.
- **Interconnectivity:** This variable is used to control the amount of connection that exists between the current reservoir node and the previous reservoir node of the network. This is also between the range of 0 and 1. With 0 being no connection and 1 having a full connection.
- **Weight range:** This is the range of the random weights to be generated for the input and output weights. The value used determines the generation of random weights between the negative of that number and itself. The generated weights are random but are a good distribution within the range provided.

The respective parameters of the network defined above are then modified one at a time.

The spectral radius of the reservoir weights connectivity was the first choice of the parameters to be modified. The other parameters of the network were maintained as suggested by [46] to modify one parameter at a time. Thus, the other parameters were modified after the spectral radius and subsequently this followed all the way till the weight range was modified. The concept applied is to modify one of the parameters and after obtaining the best performance within the range maintain it and move to the next parameter and so on until the last parameter (Weight range).

c) **Results**

As explained earlier the different parameters were modified in different combinations. The results are presented in table 5.3 and 5.4. For a better understanding of the results, the

experiment conducted to determine these parameters used the Mel frequency cepstral coefficients (MFCC) as the initial frame features, as these were adjudged to be the best features for speech recognition [24], see table 5.3 and 5.4. However, LPC features were also used to compare to the MFCC features results. These are presented in table 5.4.

Spectral Radius	Input Connectivity	Inter connectivity	Weight Range	Reservoir Size	Performance
1	0.9	0.97	0.5	50	28.14%
0.8	0.9	0.97	0.5	50	28.24%
0.6	0.9	0.97	0.5	50	42.49%
0.4	0.9	0.97	0.5	50	41.93%
0.2	0.9	0.97	0.5	50	41.43%
0.4	0.8	0.97	0.5	50	41.71%
0.4	0.6	0.97	0.5	50	33.99%
0.4	0.4	0.97	0.5	50	41.73%
0.4	0.2	0.97	0.5	50	31.21%
0.4	1	0.97	0.5	50	32.31%
0.4	0.8	1	0.5	50	41.56%
0.4	0.8	0.8	0.5	50	42.69%
0.4	0.8	0.6	0.5	50	33.95%
0.4	0.8	0.4	0.5	50	39.83%
0.4	0.8	0.2	0.5	50	41.71%
0.4	0.8	0.8	0.4	50	34.41%
0.4	0.8	0.8	0.3	50	41.61%
0.4	0.8	0.8	0.2	50	35.45%
0.4	0.8	0.8	0.1	50	43.09%
0.4	0.8	0.8	0	50	42.59%

Table 5.3 - Manually concatenated MFCC ESN phoneme (frame) recognition results for training data.

The results in table 5.3 are for manually put together phonetic labels for the training data. The manual phonetic labels are put together by individually labelling the frames of the digit sequences (1 – 9). The label for all the data representing the digit sequence 1 are done separate from the digit sequence 2 and so on. Whilst, the results in table 5.4 are based on automatic phonetic labelling of the frames. All the data representing the entire digit sequences of 1 – 9 are presented, and they are all labelled in one process.

Spectral Radius	Input Connectivity	Inter connectivity	Weight Range	Reservoir Size	Performance	
					LPC	MFCC
1	0.9	0.97	0.5	50	75.41%	67.74%
0.8	0.9	0.97	0.5	50	75.92%	64.79%
0.6	0.9	0.97	0.5	50	69.97%	46.02%
0.4	0.9	0.97	0.5	50	54.14%	47.90%
0.2	0.9	0.97	0.5	50	76.82%	69.32%
0.4	0.8	0.97	0.5	50	74.73%	53.61%
0.4	0.6	0.97	0.5	50	75.15%	56.95%
0.4	0.4	0.97	0.5	50	70.33%	60.71%
0.4	0.2	0.97	0.5	50	67.37%	63.21%
0.4	1	0.97	0.5	50	76.35%	50.67%
0.4	0.8	1	0.5	50	71.70%	49.19%
0.4	0.8	0.8	0.5	50	74.58%	66.33%
0.4	0.8	0.6	0.5	50	54.16%	62.35%
0.4	0.8	0.4	0.5	50	74.93%	54.14%
0.4	0.8	0.2	0.5	50	73.61%	67.35%
0.4	0.8	0.8	0.4	50	54.30%	65.77%
0.4	0.8	0.8	0.3	50	54.30%	68.51%
0.4	0.8	0.8	0.2	50	71.22%	45.32%
0.4	0.8	0.8	0.1	50	69.52%	65.93%
0.4	0.8	0.8	0	50	69.07%	65.22%

Table 5.4 – Automatic 17 output SOM used for ESN with MFCC and LPC on training data

Different applications of the parameters for the network have been suggested in different papers. However, due to the difference in architecture these parameters are not suitable for direct adoption. The output of these experiments is the respective recognised frames.

However, for evaluation purposes, a post processing stage was applied to get a final word recognition performance for a particular digit. The chosen method of evaluation was that, the network was deemed to have recognised a digit if **70%** of the frames in a particular digit were correct.

Using the **70%** evaluation of frames, the highest digit recognition achieved was **86.46%** on the training data using 0.5 weight range, 0.4 spectral radius, 0.8 input connectivity and 0.8 interconnectivity with 400 reservoir units. Given this rather low test performance on the

training data and in addition to the computation time of **29.65** secs for each minute of audio data on the mobile phone (see table 5.9), it was decided not to go ahead with this technique.

d) Analysis

The echo state network experiments and results are in no way an indictment of the performance of ESN with regards to computer based speech recognition, but on-device mobile use. On the contrary it highlights how good the ESN is at modelling transitions between two inputs with the limited feedback structure used in these experiments. Echo state networks have been previously used in preference to hidden Markov models because of this advantage [95]. However, the comparison done in that work used performance figures obtained from an online implementation without paying any attention to the effects on the computational and memory resources.

It is possible that, recurrent neural networks can be used for speech recognition and perform almost as well as HMMs if the input to the network was the output of a segmentation process just like the HMM approach. The state transitions of the hidden Markov models are based on transition probabilities, which in recurrent neural networks are modelled by the memory of previous states provided by the feedback connection. Though the architecture adopted in this work is shown to be non-optimal for the speech recognition task, the choice of ESN architecture was chosen due to the mobile device resources. However, for the sake of practicality, the adopted echo state network presented in tables 5.3 and 5.4 used the training data for testing. The outcome was not convincing enough on the training data performance and computation time to adopt its use for further work because this would involve adopting the following solutions for improving the ESN performance:

- i) The adoption of a larger memory thereby accommodating more of the input frames in a feedback manner; i.e. saving $t-1$, $t-2$, $t-3$ $t-N$ reservoir states.

- ii) The adoption of a classification algorithm that segments the respective phonemes within a digit prior to being fed to the neural network. The classification algorithm would then put together all the frames corresponding to a particular phoneme and the output is an averaged version of this phoneme or a unique representation of this phoneme.

Unfortunately, the problem of the adoption of the first solution for the mobile device is the issue of the extended memory which makes the network size monumental for the mobile device. In addition, the feedback involves extra calculations for every digit and the more the collection in memory the more the computational time. The second solution is not adoptable for this research because the speech recognition process implemented for the mobile device assumes the voice activity detection (VAD) stage as part of the recognition process. The adoption of an additional phoneme segmentation stage between the VAD stage and speech recognition stage would create additional computation for the mobile device.

Given the results from the ESN experiments, it was decided that a different approach had to be adopted, but one that maintains the advantage of being able to deal with the speech signal variabilities without the adoption of a new technique or segmentation algorithm between the VAD process and speech recognition process.

5.6. Proposed Novel Dynamic Feed Forward Multi-Layer Perceptron Recognizer

The experiments conducted for the MLP uses the same databases previously detailed for the ESN. However, as an MLP is a static input neural network the work done using the MLP neural network is based on a static approach i.e. presenting static whole word patterns to the network for recognition. To cope with the dynamism of the speech signal, most work in the literature adopt a segmentation pre-processing of the audio signal [85, 86] after which the output of that procedure is passed on to the Multi-Layer perceptron as a static input for the recognition task.

The novelty introduced into this technique is the concept of dynamism. The “**input-connectivity**” and “**interconnectivity**” of the ESN network are adopted in the structure of the MLP. These two parameters allow the network to exhibit dynamism by taking a signal of a varying length and feeding it to the network. By monitoring the length of this signal using the input-connectivity, only units that are needed by that particular signal are used. Subsequently, in the hidden layer the interconnectivity allows only hidden units used by the input layer to be used to calculate the output. Consequently, the speech signal is no longer restricted or concatenated to fit the neural network recogniser. On the contrary the network is the one being modified to deal with the speech signal length variabilities. The original MLP structure and training procedures are maintained and as such this algorithm is also explained in steps.

a) Speech signal pre-processing

The speech signal features used for this set of experiments are the linear predictive cepstral coefficients (LPC) and the Mel frequency cepstral coefficients (MFCC). As before, these two features are extracted in 10 – 30 msec frames. However, after the features have been extracted in frames, each input pattern to the network consists of the frames for a complete word. During training, the target reflects the digit spoken i.e. 1 for the digit 1 etc. For every digit, there are different numbers of frames, therefore the input pattern size is different for each word/digit.

b) Dynamic MLP Architecture

The architecture adopted for the dynamic MLP is based on the structure of the speech signal. An external input is adopted where the different frames are fed simultaneously into the network in parallel. This input is weighed and passed through to the hidden layer of the network and subsequently the output layer of the network. Figure 5.5 presents the network architecture.

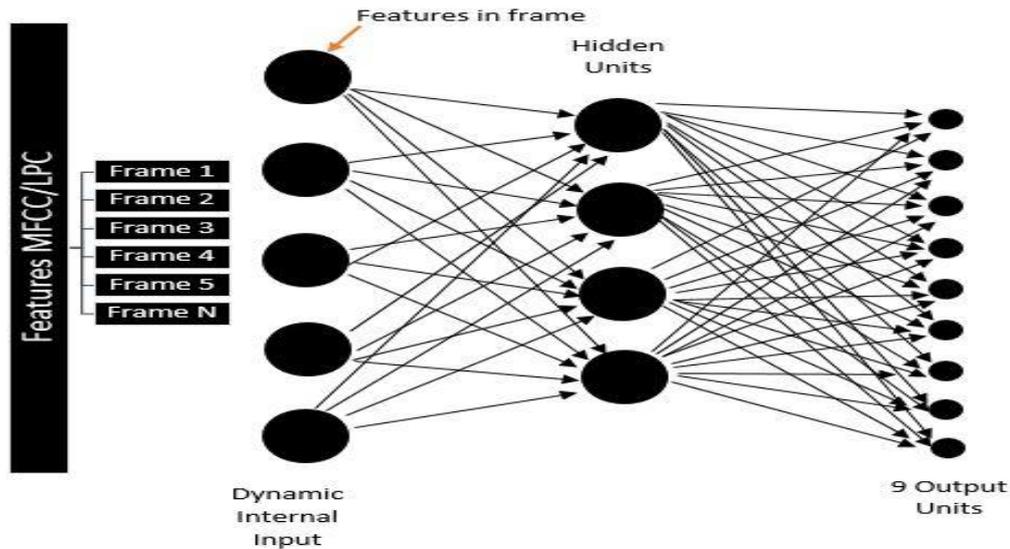


Figure 5.5 - Dynamic MLP structure

An arbitrary input size large enough to accommodate the digit with the largest frame numbers was chosen. The highest number of frames for the TIDIGITS database is 172 whilst for the CSLU database, the highest number of frames is 62. Consequently, the network was designed with a default external input size of 172 frames resulting in an internal input size of 2064 (172 frames * 12 features per frame). The standard MLP feedforward equations are used where the input layer is designated **I** (having i_1, i_2, \dots, i_k number of neurons), the hidden layer is referred to as **H** (also having h_1, h_2, \dots, h_n number of neurons) and finally the output layer of the network is regarded as **O** (having o_1, o_2, \dots, o_j number of neurons).

For an input signal to pass through the network equations 5.4 through 5.7 are used.

$$h_n = \sum_{k=1}^K W_{nk} i(k) \quad (5.4)$$

where h_n is the n^{th} neuron in the hidden layer and W_{nk} is the weight connecting the k^{th} neuron in the input layer to the n^{th} neuron in the hidden layer being calculated and $i(k)$ is the k^{th} neuron of the input layer.

$$f(h_n) = \frac{\sinh h_n}{\cosh h_n} \quad (5.5)$$

where f is the activation function and h_n is the same from equation 5.4. \sinh is the hyperbolic sine and \cosh is the hyperbolic cosine.

$$o_j = \sum_{n=1}^N W_{jn} f(h_n) \quad (5.6)$$

where o_j is the n^{th} neuron in the hidden layer and f is the linear function and W_{jn} is the weight connecting the n^{th} neuron in the hidden layer to the j^{th} neuron in the output layer, $f h_n$ is same as equation 5.5.

$$f(o_j) = o_j \quad (5.7)$$

where f is the activation function and o_j is the output neuron of the network same as equation 5.6.

The network output is computed for all the neurons in the output layer using the equations given. The network is trained, by updating the weights connecting the hidden to the output and subsequently propagating the error of that calculation backwards to the weight connecting the input to the hidden or in the case of two hidden layers to the weights connecting the hidden layer 2 to the hidden layer 1 before this is then passed on to the weights connecting the input to the hidden. The back propagation algorithm used to train the network is explained in detail in chapter 2 section 2.3.

c) Dynamic MLP Algorithm

The procedure adopted for a digit to pass through the network from the external input, all the way to the output unit of the network is given in the forward pass pseudo code in table 5.5. This table gives a better description of what happens inside the network as the digit passes from the input to the output. It is worth noting that the particular architecture, and

described pseudo code given is for a network with 1 hidden layer. For experimentation, and in the quest to find an effective network architecture, an extra hidden unit was also added to the network because in earlier formulations of the network there were issues of stability. Adding an extra unit to the network gave it a more stable number of calculating units as opposed to single hidden layer Dynamic MLP. The results of this are presented in Appendix 2.

<i>Dynamic MLP algorithm Forward Pass</i>
<p><i>For every D with F_N frames</i></p> <p style="padding-left: 2em;"><i>For the N^{th} F of D</i></p> <p style="padding-left: 2em;"><i>Arrange elements in Ascending numbers of IU</i></p> <p style="padding-left: 2em;"><i>End</i></p> <p style="padding-left: 2em;"><i>Pass IU to X</i></p> <p style="padding-left: 2em;"><i>Propagate X to H using only connected W^{in}</i></p> <p style="padding-left: 2em;"><i>For every $h_n \in H$</i></p> <p style="padding-left: 4em;"><i>Calculate h_n using equation 5.4</i></p> <p style="padding-left: 4em;"><i>Net1ⁱ = f(h_n) as calculated using equation 5.5</i></p> <p style="padding-left: 2em;"><i>End</i></p> <p style="padding-left: 2em;"><i>Propagate Net1ⁱ to O using W^{out}</i></p> <p style="padding-left: 2em;"><i>For every $o^n \in O$</i></p> <p style="padding-left: 4em;"><i>Calculate o using equation 5.6</i></p> <p style="padding-left: 4em;"><i>Calculate Net2ⁱ using equation 5.7</i></p> <p style="padding-left: 2em;"><i>End</i></p>
<i>End</i>

Table 5.5 - pseudo code of a forward pass of the Dynamic MLP

The procedure in table 5.5 is used to pass a digit through the network. Word length variability is included in the network by allowing the entire digit to pass to the hidden units of the network using only the input to hidden weights corresponding to the size of the input pattern i.e. the digit is only made to use of the number of nodes that reflect its length in the input layer. This adaptation significantly reduces the network calculation processing when

compared to padding the input with zeros and still having to calculate for the nodes which are of no relevance to the required calculation.

The Dynamic MLP thus allow a digit propagate from the input of the network all the way to the output of the network using only the number connections it needs to get to the output. This concept makes the input layer dynamic and at the same time, the weights connecting the input nodes to the hidden nodes of the network dynamic. However, the output nodes of the network remain the same because there is only a total of 9 digits in the training data which the network is being trained to recognise. The output of the network thus represents the nine digits. Both the hidden and output layers have activation functions. The activation functions adopted for these layers are the hyperbolic tangent and linear function respectively and their computations are given in equations 5.5 and 5.7.

The network was trained using the backpropagation algorithm but without any momentum factor due to the dynamic nature of the input and hidden units, a fixed learning rate was adopted. This is to ensure uniformity in the calculation and to detect any particular problems that might arise from the network becoming unstable. The input, hidden and output layers were completely flushed out when a new digit was passed into the network. This ensures that a previous larger sample does not affect the calculation of a new sample shorter.

The backpropagation algorithm works exactly similar to the forward pass of this algorithm but in a backwards manner i.e. only the weights used by the network when a digit passed through are modified using the back propagation algorithm. It can thus be said that dynamism is also implemented in the training procedure. This is also one of the reasons for adopting a fixed learning rate for the network. The parameters adopted for the network also involved the stopping conditions for the network. The maximum number of iterations used for the network is 3000 iterations and the minimum error used was 0.00001. This ensured a good balance between training and error calculations.

d) Results

The Dynamic MLP with the respective database test sets results are presented in table 5.6. The results presented here are for both databases and both sets of speech features. The performance of the network on the training data is 100% in all instances. The results presented in table 5.6 have the label of 1 or 2 attached to the feature name. These labels are used to identify how the training data was presented to the network.

Database	Hidden Units	Feature			
		LPC 1	MFCC 1	LPC 2	MFCC 2
CSLU	50	82.50%	19.60%	92.50%	95.27%
CSLU	100	80.80%	18.00%	94.40%	96.94%
CSLU	200	76.11%	14.70%	95.55%	96.38%
CSLU	300	60.27%	18.33%	95.27%	96.66%
TIDIGITS	50	66.18%	84.83%	67.28%	86.05%
TIDIGITS	100	66.87%	88.24%	69.73%	90.59%
TIDIGITS	200	69.97%	81.79%	69.87%	93.01%
TIDIGITS	300	69.32%	81.45%	70.89%	93.25%
TIDIGITS	400	69.52%	82.41%	69.35%	91.61%
TIDIGITS	500	70.07%	78.90%	70.51%	91.71%

Table 5.6 - Results of the final set of experiments conducted using the Dynamic MLP

For the label 1 the training data is arranged in ascending numbers of digits only, where all the feature vectors of the digit 1 are presented before that of 2, 3 and so on until 9. This ascension is limited to the digits not the number of frames contained in the digit. The number of frames therefore varies quite randomly during training because the arrangement is done based on the speaker position in the database.

In the second format which is labelled 2 next to the feature name, the features for the training data are extracted and arranged based on the ascending number of frames. For example, the digit with the lowest number of frames comes first, the digit with the second

lowest number of frames second, all the way up to the digit with the largest number of frames. This arrangement does not take into consideration the position of the speaker in the database but just the number of frames per digit within the sample. Consequently, the arrangement of the speakers (male, female, boy and girl) are quite random because the first digit could have come from anyone of these speakers. This is not just limited to the speaker, it is also the same for the digit position. For example, in the training data of the TIDIGITS database, one sample representing the digit 9 was the one having the lowest number of frames and as such it was the first one in the arrangement of the training data.

e) Analysis

The results presented in table 5.6 show that the second implementation favours the training of the network. This implementation appears to allow the network to learn the training data slowly over time as opposed to the much more random learning curve of the first implementation, where the first digit could be a digit with a higher or a lower number of frames. The second implementation guarantees that the training data is applied smoothly in such a way that no preceding digit in the training data set is larger in the number of frames than the succeeding digit. This can be seen in the results for the various formulations of the hidden units.

The two hidden unit approach that was adopted to initially stabilise the network also did improve the results. This is presented in Appendix 2. Though there were improvements in some cases it was quite minimal. The performance of a second hidden layer therefore does not warrant the extra amount of computation that this entails on the mobile device.

Database	Hidden Units	Feature	Recognized
CSLU	50	LPC2	92.50%
CSLU	50	MFCC2	95.27%
CSLU	100	LPC2	94.40%
CSLU	100	MFCC2	96.94%
CSLU	200	LPC2	95.55%
CSLU	200	MFCC2	96.38%
CSLU	300	LPC2	95.27%
CSLU	300	MFCC2	96.66%
TIDIGITS	500	MFCC2	91.71%
TIDIGITS	400	MFCC2	91.61%
TIDIGITS	300	MFCC2	93.25%
TIDIGITS	200	MFCC2	93.01%
TIDIGITS	100	MFCC2	90.59%

Table 5.7 Dynamic MLP best performances

From table 5.7 the best performances of the Dynamic MLP are presented. These are extracted from table 5.6. The explained advantage of the second implementation can be seen in this table because the best performance of the network are all based on the second implementation. The best performance for the MFCC features was **96.94 %**. This feature was then selected to test for comparison using the HMM in the next section. The best performance for the LPC features was **95.55%**. This was to be expected because these features are based on the assumption that the vocal tract is a linear speech production mechanism which is not true. However, the LPC feature performance is close enough to the MFCC feature set performance to be considered, as this will be shown in section 5.8, the computation time for the LPC feature vectors does make it favourable enough to neglect the extra **1.44%** performance advantage seen in the MFCC results.

5.7. Comparison with Hidden Markov Model

Hidden Markov Models are one of the most common implementation of speech recognition. As stated earlier the amount of research done in this technique dates as far back as 40 years. In this thesis, the work undertaken with regards to this technique is to use its performance to compare with the developed neural network systems.

The HMM technique is based on the concept of Markov chains which models a sequence of observation by states which have transitions between them. The next state is always dependent on the current state and this is what makes it suitable for Speech recognition purposes. The different states can be used to represent the different phonemes in the digits. The performance of the HMM is dependent on the number of states making up the individual HMM models, and a range of different states were used for experimentation. The guiding principle to the number of states is that given in table 5.1 used for the computation of the ESN training target data. This was used as a guide to the number of states but at the same time there are suggestions [113] for adopting 3 as the number of states per phoneme. This is because phonemes are said to have three states [113]. The initial state of the phoneme, followed by the stable state of the phoneme and finally the end of the phoneme which is in most cases the transitional state for it to transition to the first state of the next phoneme or even to the background. Due to this adoption of the number of states larger number of states were used.

The algorithm used for the hidden Markov models are based on a readily available java library available for use at [114]. This library implements the hidden Markov model based on the suggestion of [101]. The suggested HMM is primarily based on a speech recognition implementation and as such this renders it a very good candidate for our choice of experimentations. A K-means classification algorithm [96] used to generate the parameters of the HMMs to be trained.

Database	States	Performance
TIDIGITS	4	96.76%
TIDIGITS	8	97.30%
TIDIGITS	12	98.02%
CSLU	4	95.83%
CSLU	8	96.94%
CSLU	12	98.60%

Table 5.8 - Hidden Markov Model results for MFCC Word recognition.

Table 5.8 shows the hidden Markov model results for the two databases. By comparing tables 5.4, 5.7 and 5.8, it can be seen that the best hidden Markov models outperform both the ESN and DMLP network. However, there is a crucial advantage to this particular HMM which is not available to the DMLP. With the HMM system, a separate HMM is created for each digit. This gives HMM a unique advantage over the Neural networks methods because while a misclassification or wrong labelling by one HMM model need not necessarily affect the overall performance of a HMM system. This is not the case with the neural network implementations because they are an encompassed model that adopts all the training sequences and trains within the same network. Any misclassification will affect the performance of every other labelled data used for training the network.

5.8. Conclusion

The three approaches adopted for the mobile devices are presented. The two neural network based approaches may not necessarily be the best of implementations but were designed to address the limitations of a mobile device implementation. As stated the aim of this research is to find an effective method of speech recognition on the mobile device and as such the HMM is used only for comparison. The reasons why the ESN failed to perform as optimal as it has been researched to perform, is as a result of the restricted ESN architecture selected. The Dynamic MLP, on the other hand, builds on the concept of the adopted ESN architecture by adopting the parameters that are of importance.

The Dynamic MLP shows a novel approach to speech recognition by using static feed forward neural networks that has been adopted to the speech recognition process as opposed to a vice versa approach. This method can accommodate as input the variable length output of the respective VAD algorithms and classify them without any modifications to the length. The Dynamic MLP shows that this is possible with a feed forward neural network. Mobile device resources could easily be utilised with this network.

The particular mobile device used for this experiments was the Samsung Note 3. This mobile device is more powerful than the previous mobile device used for VAD experiments. That is an advantage because the older device is low in memory and as such the internal memory of the mobile device could not handle the HMM or neural network structures to be implemented on the device.

In this work, recognition performance is not the only parameter of interest. Speed of processing on the mobile device is equally important. Consequently, the computation times for the three algorithms with the testing data over two averaged executions were calculated and given in table 5.9.

Label	ESN (MFCC)	Dynamic MLP (MFCC)	HMM (12 States)
Average Computation time	1458.7844secs	68.875secs	831.6335secs
1 second of Test data	0.4942secs	0.0233secs	0.2818secs
1 minute of Speech data	29.65 secs	1.398 secs	16.908 secs

Table 5.9 - Average computation times of ESN, DMLP and HMM on Samsung Note 3.

It is clear from table 5.9 that the Dynamic MLP using 400 hidden units can be seen to perform **2120.9%** faster than the ESN and **1209.4%** faster than the HMM over 1 minute of speech data. The very high computation time of the echo state networks makes it unviable for use on a mobile platform, even if its recognition performance could be improved. The relatively long computation for the HMM system also makes it less optimal on a mobile

platform. The computational time advantage for the Dynamic MLP does, however, make it suitable for a mobile device approach.

Table 5.10 compares both the recognition performance rates and the computation times for the best performing DMLP and HMM implementations.

Algorithm	Structure	Performance	Total Computation Time (seconds)	Computation Time for 1 sec
Dynamic MLP	100 Hidden Units	96.94%	0.4556 secs	0.0031 secs
HMM	12 States	98.60%	40.275 secs	0.277 secs

Table 5.10 Comparison between best performance Dynamic MLP and HMM for 143.4 seconds of speech data

Although the HMM does outperform the Dynamic MLP in terms of recognition performance, it can also be seen from table 5.10 that the HMM takes significantly longer to compute than the DMLP system. Using the time for the best performing network structure of the Dynamic MLP, it would be safe to say that the marginal recognition performance advantage of **1.66%** for the HMM can be disregarded. This is because the Dynamic MLP computation time is **8935.4%** faster than the HMM for each second of the test data. This is too big a computational time advantage to ignore for the sake of a **1.66%** performance advantage.

The computation time for the HMM would also grow exponentially over more sets of target data because HMMs are designed and implemented for isolated recognition in such a manner that a HMM representation of every word is created. To test a HMM for recognition of any unknown word is to test the word against each HMM implementation of the respective training data. This incurs an additional computation time which would grow exponentially. However, the problem of comparison of the test for each trained word representation individually is not shared by neural networks which makes them faster because they are computed using the same network to process all digits. Although the computation time of the

neural network would also increase as the network size increases, this increase is expected to be significantly less than for the equivalent HMM system. However, future work to qualify this assertion would be necessary.

As a consequence of the different techniques experimented on in this chapter and their respective outcomes, the latter part of the framework adopted is constructed with the outcomes shown in figure 5.6 below.

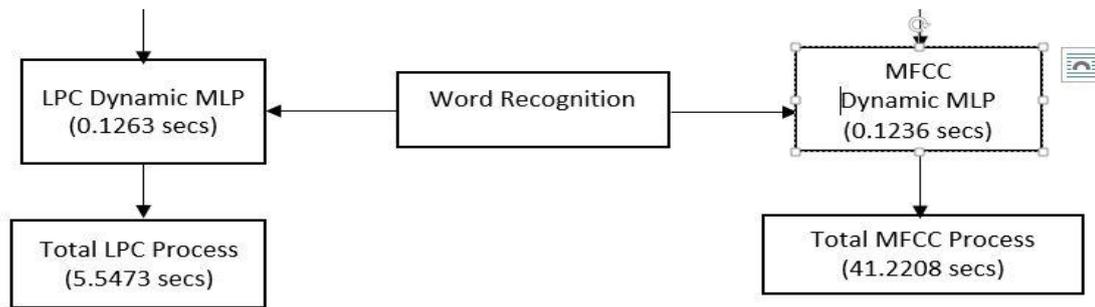


Figure 5.6 Framework Outcome Chapter 5

5.9. Summary

This chapter has presented experimental work on two neural network speech recognition classifier structures. Firstly, a recurrent echo state neural network is presented. The results from this were not sufficient to consider moving forward with the research, so a second approach was tried using a dynamic MLP. The rationale behind such a decision, provided by insights gained with the ESN, lead to the implementation of a much more dynamic MLP structure. Some degree of dynamism were introduced into the architecture of the MLP network and the outcome was a network tailored to fit the speech recognition task on a mobile device. A standard speech recognition adopted HMM was also used to compare the results of the novel Dynamic MLP system. The comparison highlights the importance of future work to further adapt the Dynamic MLP for large vocabulary speech processing and recognition purposes.

Chapter Six

Conclusion and Future Work

6. Introduction

The research presented in this thesis investigated the development of an on-device isolated mobile based speech recognition system. Mobile devices are known to have an inferior performance level, both in terms of computational power and memory as compared to personal or server computers. This has led to current mobile based recognition systems adopting a client- server infrastructure to compensate for the limited capacity of mobile devices. It is therefore in the absence of a practical framework for on-device based mobile recognition that this research is framed. A working framework has been developed in this research that can be used as the future adoption for mobile device based speech recognition system. A bottom-up approach is adopted that starts from the speech signal and goes all the way up to the recognition stage. For each stage of the research, a black box approach was adopted to optimize that particular stage/step for the mobile device before going ahead to the next stage. At the same time the challenges of working with practical mobile device capabilities were considered alongside the respective implementations. By using current commercial mobile devices for the research, as opposed to emulators, the developed framework has been properly tested on the end user devices that are the intended target for this technology. The trend in previous research of this type has focused around achieving high recognition performance figures and ignoring the consequences of the performance requirements on end user computational resources. The framework developed in this work is contrary to this notion because the novel approaches investigated have been developed bearing in mind the consequences of performance on the limited processing capability of the

mobile end user devices that are the target for this technology. The choices made in this respect are detailed in the following sections.

6.1 The Choice of LPC Features

In the literature of speech recognition, there is a high level of preference for MFCC features for recognition as against the LPC [4,17,23]. However, the preference for using the LPC features in this work comes from the fact that LPC features are faster to compute than MFCC features. As such, in terms of the initially set objectives of this research, an efficient choice of speech parameters has been identified. The investigation into the respective speech features yielded LPC features as the best candidate for isolated on-device speech recognition. Though they do not represent the absolute best recognition performance, they do have the shortest computation time. Moreover, a few works in the literature have also adopted these features for recognition, especially for a speaker dependent environment, and they have proved to be good candidates for this [78,85,108]. The results presented in chapter 5, as well as the ones presented in the framework, have proven that LPC features can be adopted for on-device speech recognition and that they are capable of producing very good recognition performance.

6.2 The Choice of VAD

One of the most obvious ways to get away with performing Voice activity detection for speech recognition is by the adoption of filtering techniques. As such, sophisticated filters have been designed in the literature over time to do this task [5,107]. However, the reason for adopting a feature based VAD approach in this research is that recent advancements in the design of mobile devices have resulted in a plethora of configurations and filters for the microphone set up on mobile devices. The most notable and physically verifiable aspect of

these advancements being the adoption of a dual microphone structure for modern mobile devices. Due to the market competition involved in the mobile market, the filtration parameters and specifics of these mobile devices are either hidden from the literature or under patent. However, the working principle of this dual microphone set up is quite straightforward even though the specifics and parameters are not available. The primary microphone is used for recording the phone call or any audio related task while the second microphone is used to cancel out the background noise or any interference from the primary microphone. Unfortunately, without a first-hand knowledge of the parameters and working principle of these microphones, attempts at filtration can be futile. A developed filter can end up working against the already established dual microphone parameters.

Consequently, statistical VAD algorithms were developed that have the flexibility of adjusting to different input signals. This primarily benefits on-device mobile speech recognition because it allows for customisation to the respective speaker speaking characteristics. This equally makes the developed techniques future proof. In addition, since VAD is done after feature extraction it can be considered as a post filtration analysis and as such does not affect the filter set up of the mobile devices. Use of a VAD algorithm also allows a pause mechanism to be adopted in the recording process thereby saving power consumption. The developed techniques have aided the achievement of the initial research objective in terms of identifying an efficient VAD algorithm that is capable of real-time performance whilst taking into account the limitations of the mobile device.

6.3 K-means VAD Choice

By nature, the K-means algorithm does not adopt a training procedure [42]. This allows the K-means technique to be used as a flexible tool to evaluate the outcome of a VAD analysis. The flexibility explained in the VAD analysis can also be extended to the K-means

algorithm. The novel modifications introduced into the K-means algorithm, explained in chapter 4, enable a very fast computation of this algorithm compared to the SOM. This makes it suitable for the evaluation of the respective outputs of the VAD analysis on a mobile device. The K-means LPC MSD VAD algorithm presents the best performance and shortest computation time of all the developed VAD algorithms. As a result this technique is the preferred choice of VAD algorithm for the mobile device. The results presented in terms of time of computation for the K-means algorithm particularly justifies its use for on-device processing. Development of this algorithm thus achieves one of the primary objectives of the research – real-time on-device Voice activity detection with the highest accuracy possible.

6.4 Artificial Neural Networks classifier choice

While much of the literature on speech recognition is filled with years of research on the HMM techniques, neural networks are also gaining popularity. As argued in [17], the choice of adopting neural networks over HMM is that, ANN can provide a complete phonemic, lexical, pragmatic, syntactic and semantic knowledge of speech. The processing times presented in table 5.10 also proves that a DMLP speech recognition network can outperform a HMM equivalent in terms of processing time. This also makes them more appealing for implementation on the sort of processor restricted end-user devices that are the target for this work. The final outcome from this work therefore achieves the main aim of this work – the development of a fast and efficient on-device mobile speech recognition system. Indeed, this technique has not only achieved the final objective of the research, but has been proven to outperform the current state of the art equivalent system in terms of processing speed. To be precise, this technique is 86 times faster than the equivalent state of the art HMM technique.

6.5 Contributions to Knowledge

The work presented in this thesis demonstrates conclusively that the general notion, prevalent within the speech recognition literature, of adopting increasingly complicated and computationally expensive techniques to achieve better speech recognition performances, is not viable for mobile speech recognition. This result is particularly supported by the fact that the developed technique is 86 times faster than the current state of the art speech recognizer adopted in the most popular On-device speech recogniser system currently in circulation; and by extension the generally adopted technique within the speech recognition community.

This work has also shown that it is possible to achieve real-time mobile on-device speech recognition with comparable recognition performance to a client-server approach. These results therefore prove that speech recognition can be achieved on mobile devices without the need to adopt a client-server approach.

Finally, the work presented in this thesis has enabled techniques to be developed and adopted that bear in mind the constraints of mobile devices whilst, at the same time, not compromising the quality and importance of each step in the speech recognition process.

6.6 Mobile Device Speech Recognition framework

Consequently, the outcome of the research presented in this thesis has enabled the development of a framework for future mobile device recognition. This framework is given in figure 6.1.

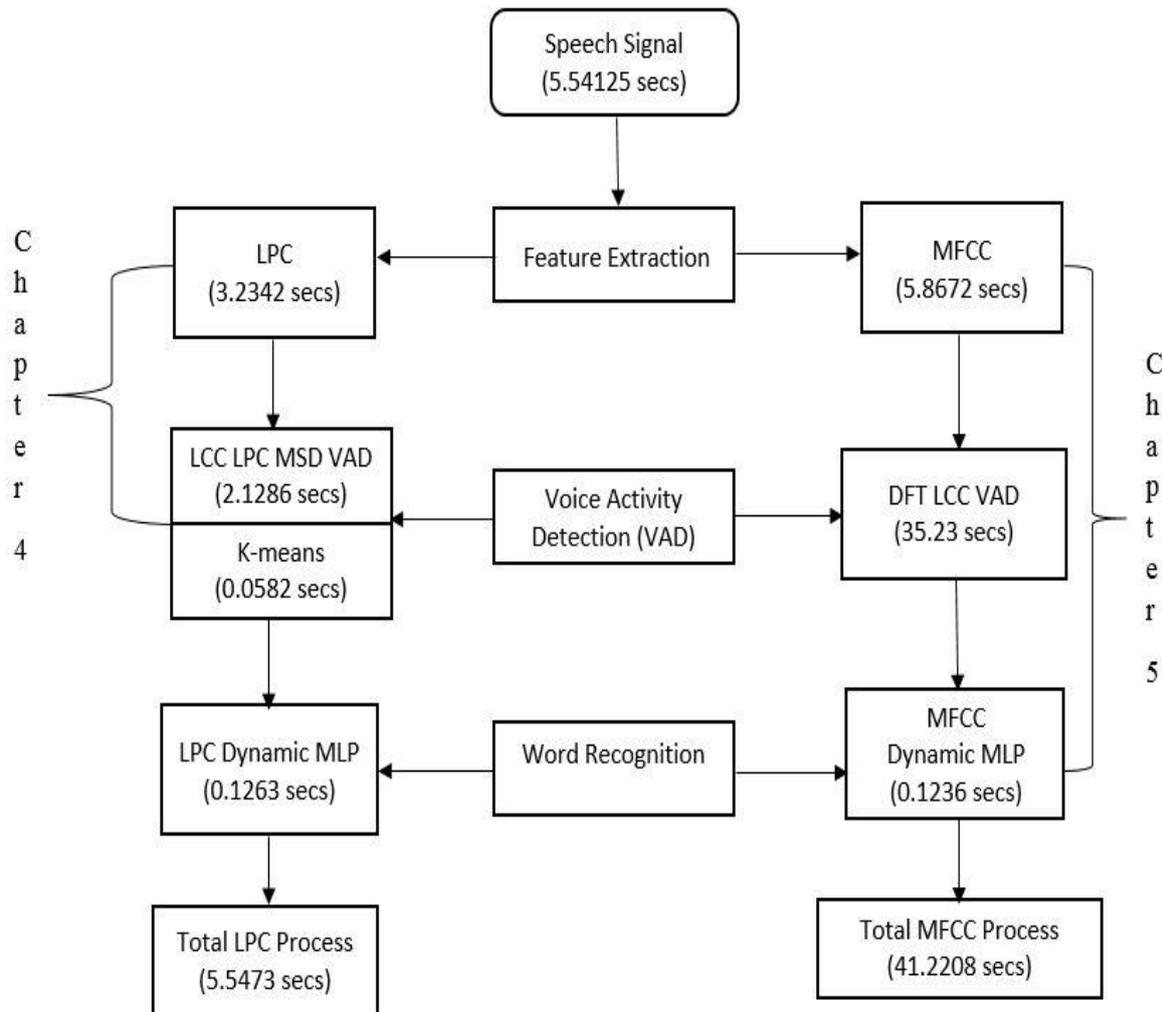


Figure 6.1 Framework showing Computation Times

The computational times taken to compute the respective steps for the LPC based speech recognition algorithm can be seen from the framework flowchart. Despite the different steps involved, it is worthy of note that the entire process of LPC feature extraction, VAD segmentation and word recognition for a sample of **5.54125 secs**, takes less computation time than is incurred by the MFCC feature extraction process alone. For this reason, the use of

MFCC features is not recommended on mobile devices as the performance figures difference between the two feature sets are negligible for this task. See table 6.1

Feature	Hidden Units	Performance %
LPC	200	95.55%
MFCC	100	96.94%

Table 6.1 Performance for CSLU for the LPC and MFCC feature sets over 145.4 seconds

It is also worthy of note that the computation time given for the Dynamic MLP in figure 6.1 is that of a 300 hidden units network tested with an external input layer of 172 frames. When the network was presented with the CSLU test dataset, the computation time per second dropped from an average of **22.6 msec**s for the TIDIGITS data to **18 msec**s for the CSLU data.

As stated before the best performing Dynamic MLP is significantly better in terms of processing time than its HMM equivalent. For a quick recap of these performance, table 5.10 is repeated here as table 6.2.

Algorithm	Structure	Performance	Total Computation Time (secs)	Computation Time for 1 sec
Dynamic MLP	100 Hidden Units	96.94%	0.4556 secs	0.0031 secs
HMM	12 States	98.60%	40.275 secs	0.277 secs

Table 6.2 - Comparison between best performance Dynamic MLP and HMM over 145.4 seconds

It can be seen from table 6.2 that the Dynamic MLP easily outperforms the HMM in terms of computation time, whilst only having a marginally lower recognition performance.

6.7 Recommendations

Given the work presented in this thesis. It is recommended that for an On-device mobile isolated speech recognition the following steps should be adopted:

- The mobile device audio signal encoding format should be considered because this will save computation time in decoding into formats suitable for speech processing. The different formats of audio signals (WAVE, MP3, AAC etc.), each have different encoding formats and as such decoding them requires extra computation time. For this reason the particular format of the end user mobile device for which the system is to be developed should be considered as the priority format ;
- For a speaker dependent approach, the LPC features should be adopted. MFCC can be used for recognition but this requires an extra set of techniques to be adopted to deal with isolated speech signals and by extension continuous speech signals. While the techniques adopted for the LPC in this research are based on isolated speech, these techniques can be easily modified with regards to making them viable for continuous recognition;
- A flexible VAD algorithm should be adopted in order to deal with different audio signals and allow customisation for user specific use. The optimum solution found in this research is the K-means based statistical approach;
- Without a licence or access to hardware design of mobile devices, it is recommended to avoid using filter based VAD algorithms;
- For word level recognition, it is important to adopt a technique or algorithm that can handle the variability of the speech signal without concatenating or discarding too much information. The Dynamic MLP developed in this work is an optimal example of this approach.

6.8 Future Work

The results presented in this thesis provide a good framework from which more sophisticated on-device speech recognition systems can be developed for mobile devices. The results are encouraging enough, in terms of performance and computation times for on-device recognition, to warrant further work in improving this recognition challenge. The following is the recommended future work in this regard;

6.7.1 Further Investigation into Human speech production and feature selection

The original convention in speech recognition with regards to feature selection is based on the assumption that the human vocal tract maintains a fixed set of characteristics within a time frame of 10 – 30 msec [25]. The point of interest with regards to this production is based on looking at the structure of word utterances. An example to illustrate this is the digit 1. While it is accepted that the number of phonemes within this digit are three (W, A and N), the analysis done on the speech signal is performed in terms of the window size of 10 – 30msecs. This ensures having more than one frame representing each of the phonemes. However, assuming the digit 1 lasts a second to utter, and a window or frame size of 25msecs is used, this will result in 40 frames for the word, resulting in multiple representations of the respective phonemes making up the word.

Using the Dynamic MLP concept, it would be interesting to defy the convention of 10 – 30msecs time steps and instead segment the speech signal into only the 3 phonemes and then present the pattern recognition system with only 3 frames representing the 3 phonemes with their variable lengths. This could result in a frame size of 300msecs +/- for each of the three respective phonemes. The plausibility or viability of this method would need to be ascertained with further work.

6.7.2 Advancing into Continuous Speech recognition

The adoption of a continuous speech recognition system is very marketable. This could be used for dictation software and different aspects of speech recognition on a mobile device. How the VAD algorithms developed in this research could be adopted to the more complex task of continuous recognition would be an interesting area of research. The VAD algorithms developed are customisable and this leaves them open ended enough to attempt integration into a continuous speech recognition system with a few degree of tuning. However, how much tuning and reconstruction is required cannot be ascertained without further investigation.

An insight into this solution can be linked to the frame distance parameter of the respective VAD algorithms. This would be a good starting point because the pauses between words in continuous speech recognition are minimal compared to isolated speech recognition.

6.7.3 Adoption of Dynamic MLP into other aspects of the recognition process

In chapter 3, a few of the algorithms presented in the literature adopted the original MLP algorithm as a feature extractor. This was then used to extract a new set of features called MLP features for use by another recogniser. The use of the MLP as a feature selector still suffers from the problem of speech length variabilities. The adoption of the Dynamic MLP as the feature extractor, as opposed to the conventional MLP, would be an interesting set of work to see how this affects the performance figures claimed in such publications.

6.7.4 Further Work on Echo state networks

The recommended solutions for the ESN problem explained in chapter 5 could be further explored. Unfortunately, an on-line approach would probably have to be adopted to accommodate for the computational time. However, an integrated segmentation/classification

algorithm could be adopted, that does away with the need for VAD algorithms so allowing time to accommodate for the extra computation time required by ESN classifiers and their post-processing. The effect of this will be interesting to integrate into the recognition process.

6.7.5 Dynamic MLP in other domains

While the Dynamic MLP developed has been developed with the speech signal variabilities in mind, these variabilities are not limited to speech signals. This approach could be extended to different time series problems. In time series problems it is a common occurrence that data of different length and size represent the same entity. From the classification results obtained and the computation time saved by using a Dynamic MLP, this approach could be a good candidate for extension into different domains of time series problems or even other problems, not related to time series, that involve variable length data.

References

1. Rao, T. V., Ch Sridevi, and M. S. Mohan. "Voice and Gesture Based Electric-Powered Wheelchair Using ARM." *IJRCCT* 1.6 (2012): 278-283.
2. Rosenberg, Aaron E., et al. "Demisyllable-based isolated word recognition system." *Acoustics, Speech and Signal Processing, IEEE Transactions on* 31.3 (1983): 713-726.
3. Echeverry-Correa, J. D., et al. "Topic identification techniques applied to dynamic language model adaptation for automatic speech recognition." *Expert Systems with Applications* 42.1 (2015): 101-112.
4. Hanilçi, Cemal, and Tomi Kinnunen. "Source cell-phone recognition from recorded speech using non-speech segments." *Digital Signal Processing* 35 (2014): 75-85.
5. K. Y. Chan, S. Nordholm, K. F. C. Yiu, and R. Togneri, Speech enhancement strategy for speech recognition microcontroller under noisy environments, *Neurocomputing*, vol. 118, pp. 2013, 279-288.
6. Siniscalchi, Sabato Marco, et al. "Exploiting deep neural networks for detection-based speech recognition." *Neurocomputing* 106 (2013): 148-157.
7. S. So and K. K. Paliwal, Scalable distributed speech recognition using Gaussian mixture model-based block quantisation, *Speech Commun.*, vol. 48, pp. 2006, 746-758.
8. M. N. Stuttle. A Gaussian Mixture Model Spectral Representation for Speech Recognition. Submitted to Cambridge University, July 2003.
9. L. Deng, M. Aksmanovic, X. Sun, and C. J. Wu, Speech recognition using hidden Markov models with polynomial regression functions as nonstationary states, *Speech and Audio Processing, IEEE Transactions on*, vol. 2, pp. 1994, 507-520.

10. I. Patel and Y. S. Rao, Speech Recognition Using Hidden Markov Model with MFCC-Subband Technique: Recent Trends in Information, Telecommunication and Computing (ITC), 2010 International Conference on, pp. 168-172, IEEE, 2010.
11. A. Coy and J. Barker, An automatic speech recognition system based on the scene analysis account of auditory perception, *Speech Commun.*, vol. 49, pp. 2007, 384-401.
12. Biswas, Astik, et al. "Articulation based admissible wavelet packet feature based on human cochlear frequency response for TIMIT speech recognition." *Ain Shams Engineering Journal* 5.4 (2014): 1189-1198.
13. A. Alarifi, I. Alkurtass, and A. Al-Salman, Arabic text-dependent speaker verification for mobile devices using artificial neural networks: Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on, vol. 2, pp. 350-353, IEEE, 2011.
14. A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, Phoneme recognition: neural networks vs. hidden Markov models vs. hidden Markov models: Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on, pp. 107-110, IEEE, 1988.
15. Shahamiri, Seyed Reza, and Siti Salwah Binti Salim. "Real-time frequency-based noise-robust Automatic Speech Recognition using Multi-Nets Artificial Neural Networks: A multi-views multi-learners approach." *Neurocomputing* 129 (2014): 199-207.
16. Li, Xiangang, et al. "A Comparative Study on Selecting Acoustic Modeling Units in Deep Neural Networks Based Large Vocabulary Chinese Speech Recognition." *Neurocomputing* (2015).

17. Mirhassani, Seyed Mostafa, and Hua-Nong Ting. "Fuzzy-based discriminative feature representation for children's speech recognition." *Digital Signal Processing* 31 (2014): 102-114.
18. K. Waheed, K. Weaver, and F. M. Salam, A robust algorithm for detecting speech segments using an entropic contrast: Circuits and Systems, 2002. MWSCAS-2002. The 2002 45th Midwest Symposium on, vol. 3, pp. III-328-III-331 vol. 3, IEEE, 2002.
19. Z. Hachkar, B. Mounir, A. Farchi, and J. El ABBADI, Comparison of MFCC and PLP parameterization in pattern recognition of Arabic alphabet speech, Canadian Journal on Artificial Intelligence, Machine Learning & Pattern Recognition, vol. 2, pp. 2011, 56-60.
20. D. Huggins-Daines, M. Kumar, A. Chan, A. W. Black, M. Ravishankar, and A. I. Rudnicky, Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices: Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on, vol. 1, pp. I-I, IEEE, 2006.
21. R. C. Rose and I. Arizmendi, Efficient client–server based implementations of mobile speech recognition services, *Speech Commun.*, vol. 48, pp. 2006, 1573-1589.
22. Definitive data and analysis for the mobile industry. Available online at <https://gsmaintelligence.com/> Accessed 18th June, 2015.
23. M. J. Alam, P. Kenny, and D. O’Shaughnessy, Regularized minimum variance distortionless response-based cepstral features for robust continuous speech recognition, *Speech Commun.*, vol. 73, pp. 2015, 28-46.
24. Plannerer, Bernd. : An introduction to speech recognition. 28 March (2005). <http://spoken-number-recognition.googlecode.com/svn/trunk/docs/introSR.pdf>. Accessed 17 June 2013.

25. L. R. Rabiner and R. W. Schafer. (1979). Digital processing of speech signals, Institution of Engineering and Technology.
26. Rabiner, Lawrence R., and Biing-Hwang Juang. (1993). Fundamentals of Speech Recognition. Englewood Cliffs: PTR Prentice Hall.
27. L. R. Rabiner and M. R. Sambur, An algorithm for determining the endpoints of isolated utterances, The Bell System Technical Journal, vol. 54, pp. 1975, 297-315.
28. D. Enqing, L. Guizhong, Z. Yatong, and C. Yu, Voice activity detection based on short-time energy and noise spectrum adaptation: Signal Processing, 2002 6th International Conference on, vol. 1, pp. 464-467, IEEE, 2002.
29. T. Tashan. Biologically Inspired Speaker Verification. Submitted to Nottingham Trent University, November 2012.
30. J. W. Cooley and J. W. Tukey, An algorithm for the machine calculation of complex Fourier series, Mathematics of computation, vol. 19, pp. 1965, 297-301.
31. P. Vaidyanathan, The theory of linear prediction, Synthesis Lectures on Signal Processing, vol. 2, pp. 2007, 1-184.
32. D. L. Jones, S. Appadwedula, M. Berry, M. Haun, J. Janovetz, M. Kramer, D. Moussa, D. Sachs, and B. Wade, Speech Processing: Theory of LPC Analysis and Synthesis, , pp. 2009.
33. T. Kamm, H. Hermansky, and A. G. Andreou, Learning the Mel-scale and optimal VTN mapping: Center for Language and Speech Processing, Workshop (WS 1997). Johns Hopkins University, Citeseer, 1997.
34. Hai, Jiang, and Er Meng Joo. "Improved linear predictive coding method for speech recognition." *Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on*. Vol. 3. IEEE, 2003.

35. Zheng, Fang, Guoliang Zhang, and Zhanjiang Song. "Comparison of different implementations of MFCC." *Journal of Computer Science and Technology* 16.6 (2001): 582-589.
36. Doniec, Marek W., Brian Scassellati, and Willard L. Miranker. "Emergence of language-specific phoneme classifiers in self-organized maps." *Neural Networks, 2007. IJCNN 2007. International Joint Conference on. IEEE, 2007.*
37. Gardón, A. Postigo, C. Ruiz Vázquez, and A. Arruti Illarramendi. "Spanish phoneme classification by means of a hierarchy of kohonen self-organizing maps." *Text, Speech and Dialogue*. Springer Berlin Heidelberg, 1999.
38. Szczurowska, Izabela, Wiesława Kuniszyk-Józkowiak, and Elżbieta Smółka. "The application of Kohonen and Multilayer Perceptron Networks in the speech nonfluency analysis." *Archives of Acoustics* 31.4 (S) (2014): 205-210.
39. Rahman, Md Mijanur, and Md Al-Amin Bhuiyan. "DYNAMIC THRESHOLDING ON SPEECH SEGMENTATION." *International Journal of Research in Engineering and Technology*. doi: 10.15623/ijret.2013.0209061
40. J. Wilpon and L. Rabiner, A modified K-means clustering algorithm for use in isolated work recognition, *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 33, pp. 1985, 587-594.
41. H. Yin, *The self-organizing maps: background, theories, extensions and applications*, Computational intelligence: A compendium, Springer, 2008.
42. K. R. Žalik, An efficient k'-means clustering algorithm, *Pattern Recog. Lett.*, vol. 29, pp. 2008, 1385-1391.
43. C. G. Looney, A fuzzy clustering and fuzzy merging algorithm, *CS791q Class notes*, , pp. 1999.

44. Noriega, Leonardo. "Multilayer perceptron tutorial." *School of Computing. Staffordshire University* (2005).
45. R. Rojas, *Neural networks: a systematic introduction*, Springer Science & Business Media 2013.
46. M. Lukoševičius, *A Practical Guide to Applying Echo State Networks*, *Neural Networks: Tricks of the Trade*, Springer, 2012.
47. H. Jaeger, Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach, GMD-Forschungszentrum Informationstechnik 2002.
48. Álvaro, Francisco, Joan-Andreu Sánchez, and José-Miguel Benedí. "Recognition of on-line handwritten mathematical expressions using 2d stochastic context-free grammars and hidden markov models." *Pattern Recognition Letters* 35 (2014): 58-67.
49. Fink, Gernot A. *Markov models for pattern recognition: from theory to applications*. Springer Science & Business Media, 2014.
50. L. R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, *Proc IEEE*, vol. 77, pp. 1989, 257-286.
51. CSLU database Available online at <https://www.cslu.ogi.edu/corpora/spkrec/>.
52. TIDIGITS database details Available online at <https://catalog.ldc.upenn.edu/topten>.
53. S. A. Ali, N. G. Haider, and M. K. Pathan, A LPC-PEV based VAD for word boundary detection, *Int.J.Electr.Comput.Sci*, vol. 12, pp. 2012.
54. D. Cournapeau, T. Kawahara, K. Mase, and T. Toriyama, Voice activity detector based on enhanced cumulant of lpc residual and on-line em algorithm, *Proceedings of Interspeech06*, , pp. 2006.

55. B. Wu and K. Wang, Voice activity detection based on auto-correlation function using wavelet transform and teager energy operator, *Computational Linguistics and Chinese Language Processing*, vol. 11, pp. 2006, 87-100.
56. SIRI Available Online at <http://www.apple.com/uk/ios/siri/>
57. Google Speech Recognition Available Online at <http://www.google.co.uk>
58. Cortana Personal assistant. Available Online at <http://www.windowsphone.com/en-gb/how-to/wp8/cortana/meet-cortana>
59. H. Huang and F. Lin, A speech feature extraction method using complexity measure for voice activity detection in WGN, *Speech Commun.*, vol. 51, pp. 2009, 714-723.
60. S. Ben Jebara, Voice activity detection using periodic/aperiodic coherence features: *Signal Processing Conference, 2008 16th European*, pp. 1-5, IEEE, 2008.
61. H. Ghaemmaghami, B. J. Baker, R. J. Vogt, and S. Sridharan, Noise robust voice activity detection using features extracted from the time-domain autocorrelation function, *Proceedings of Interspeech 2010*, , pp. 2010.
62. A. N. Iyer, M. Gleiter, B. Y. Smolenski, and R. E. Yantorno, Structural usable speech measure using lpc residual: *International Symposium on Intelligent Signal Processing and Communication Systems*, vol. 2, pp. 236-240, 2003.
63. Nemer, Elias, Rafik Goubran, and Samy Mahmoud. "Robust voice activity detection using higher-order statistics in the LPC residual domain." *Speech and Audio Processing, IEEE Transactions on* 9.3 (2001): 217-231.
64. Enqing, Dong, et al. "Voice activity detection based on short-time energy and noise spectrum adaptation." *Signal Processing, 2002 6th International Conference on*. Vol. 1. IEEE, 2002.
65. Sohn, Jongseo, Nam Soo Kim, and Wonyong Sung. "A statistical model-based voice activity detection." *Signal Processing Letters, IEEE* 6.1 (1999): 1-3.

66. Chang, Joon-Hyuk, Nam Soo Kim, and Sanjit K. Mitra. "Voice activity detection based on multiple statistical models." *Signal Processing, IEEE Transactions on* 54.6 (2006): 1965-1976.
67. Wang, Kun-Ching, and Yi-Hsing Tasi. "Voice activity detection algorithm with low signal-to-noise ratios based on spectrum entropy." *Universal Communication, 2008. ISUC'08. Second International Symposium on*. IEEE, 2008.
68. Deng, Shi-Wen, and Ji-Qing Han. "Statistical voice activity detection based on sparse representation over learned dictionary." *Digital Signal Processing* 23.4 (2013): 1228-1232.
69. Chen, Shi-Huang, et al. "Improved voice activity detection algorithm using wavelet and support vector machine." *Computer Speech & Language* 24.3 (2010): 531-543.
70. McCowan, Iain, et al. "The delta-phase spectrum with application to voice activity detection and speaker recognition." *Audio, Speech, and Language Processing, IEEE Transactions on* 19.7 (2011): 2026-2038.
71. Ramirez, Javier, et al. "Efficient voice activity detection algorithms using long-term speech information." *Speech communication* 42.3 (2004): 271-287.
72. Ghosh, Prasanta Kumar, Andreas Tsiartas, and Shrikanth Narayanan. "Robust voice activity detection using long-term signal variability." *Audio, Speech, and Language Processing, IEEE Transactions on* 19.3 (2011): 600-613.
73. Ramírez, Javier, et al. "A new Kullback-Leibler VAD for speech recognition in noise." *Signal Processing Letters, IEEE* 11.2 (2004): 266-269.
74. Solvang, Hiroko Kato, Kentaro Ishizuka, and Masakiyo Fujimoto. "Voice activity detection based on adjustable linear prediction and GARCH models." *Speech Communication* 50.6 (2008): 476-486.

75. Górriz, Juan Manuel, et al. "Hard C-means clustering for voice activity detection." *Speech communication* 48.12 (2006): 1638-1649.
76. Tashan, T., Allen, T. and Nolle, L. (2013), Speaker verification using heterogeneous neural network architecture with linear correlation speech activity detection. *Expert Systems*. doi: 10.1111/exsy.12030
77. S. Masmoudi, M. Chtourou, and A. Ben Hamida, Isolated word recognition system using MLP neural network constructive training algorithm: *Systems, Signals and Devices, 2009. SSD'09. 6th International Multi-Conference on*, pp. 1-6, IEEE, 2009.
78. Wijoyo, Thiang Suryo. "Speech recognition using linear predictive coding and artificial neural network for controlling movement of mobile robot." *Proceedings of International Conference on Information and Electronics Engineering (ICIEE 2011)*. 2011.
79. Chen, Wen-Yuan, Sin-Horng Chen, and Cheng-Jung Lin. "A speech recognition method based on the sequential multi-layer perceptrons." *Neural Networks* 9.4 (1996): 655-669.
80. Dede, Gülin, and Murat Hüsni Sazlı. "Speech recognition with artificial neural networks." *Digital Signal Processing* 20.3 (2010): 763-768.
81. Ahad, Abdul, Ahsan Fayyaz, and Tariq Mehmood. "Speech recognition using multilayer perceptron." *Students Conference, 2002. ISCON'02. Proceedings. IEEE*. Vol. 1. IEEE, 2002.
82. Kumar, P. Phani, K. S. N. Vardhan, and K. Sri Rama Krishna. "Performance evaluation of MLP for speech recognition in noisy environments using MFCC & wavelets." *International Journal of Computer Science & Communication (IJCSC)* 1.2 (2010): 41-45.

83. Luka, Matthew K., Ibikunle A. Frank, and Gregory Onwodi. "Neural network based Hausa language speech recognition." *International Journal of Advanced Research in Artificial Intelligence* 1.2 (2012).
84. Bengtsson, Tobias. "Speech recognition using multilayer perceptron artificial neural network." *Department of Computer Science Lund University*. Available at http://fileadmin.cs.lth.se/cs/Education/EDA171/Reports/2006/tobias_b.pdf
85. M.K.Deka, et al. "An Approach to Noise Robust Speech Recognition using LPC-Cepstral Coefficient and MLP based Artificial Neural Network with respect to Assamese and Bodo Language". *IJCA Proceedings on International Symposium on Devices MEMS, Intelligent Systems & Communication (ISDMISC)* (4):23-26, 2011.
86. Eng, Goh Kia, and Abdul Manan Ahmad. "Malay Speech Recognition using Self-Organizing Map and Multilayer Perceptron." *Proceedings of the Postgraduate Annual Research Seminar*. 2005.
87. Meysam Mohamad Pour and Fardad Farokhi. "An Advanced Method for Speech Recognition". World Academy of Science and Technology, pp. 995-1000, 2009
88. Shekofteh, Yasser, Farshad Almasganj, and Ayoub Daliri. "MLP-based isolated phoneme classification using likelihood features extracted from reconstructed phase space." *Engineering Applications of Artificial Intelligence* 44 (2015): 1-9.
89. Frikha, Mondher, and Ahmed Ben Hamida. "A comparative survey of ANN and hybrid HMM/ANN architectures for robust speech recognition." *American Journal of Intelligent Systems* 2.1 (2012): 1-8.
90. Park, Junho, et al. "Training and adapting MLP features for Arabic speech recognition." *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009.

91. Djemili, Rafik, Mouldi Bedda, and Hocine Bourouba. "Recognition of spoken arabic digits using neural predictive hidden markov models." *Int. Arab J. Inf. Technol.* 1.2 (2004): 226-233.
92. Morgan, Nelson, and Herve Bourlard. "Continuous speech recognition using multilayer perceptrons with hidden Markov models." *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on.* IEEE, 1990.
93. Skowronski, Mark D., and John G. Harris. "Automatic speech recognition using a predictive echo state network classifier." *Neural networks* 20.3 (2007): 414-423.
94. Skowronski, Mark D., and John G. Harris. "Noise-robust automatic speech recognition using a predictive echo state network." *Audio, Speech, and Language Processing, IEEE Transactions on* 15.5 (2007): 1724-1730.
95. Skowronski, Mark D., and John G. Harris. "Noise-robust automatic speech recognition using a discriminative echo state network." *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on.* IEEE, 2007.
96. Juang, Biing-Hwang, and Lawrence R. Rabiner. "The segmental K-means algorithm for estimating parameters of hidden Markov models." *Acoustics, Speech and Signal Processing, IEEE Transactions on* 38.9 (1990): 1639-1641.
97. Hmad, Nadia, and Tony Allen. "Echo State Networks for Arabic Phoneme Recognition." *Proceedings of World Academy of Science, Engineering and Technology.* No. 79. World Academy of Science, Engineering and Technology (WASET), 2013.
98. T. Schmidt, M. Wiering, A. van Rossum, R. van Elburg, T. Andringa, and B. Valkenier, Robust Real-Time Vowel Classification with an Echo State Network: 2010. Available at http://publicationslist.org/data/janny.ramakers/ref-14/Robust%20Real-Time%20Vowel%20Classification%20with%20an%20Echo%20State%20Network%20CONAS_2010.pdf

99. Verstraeten, David, Benjamin Schrauwen, and Dirk Stroobandt. "Reservoir-based techniques for speech recognition." *Neural Networks, 2006. IJCNN'06. International Joint Conference on.* IEEE, 2006.
100. A. Alalshkembarak and L. S. Smith, A noise robust Arabic speech recognition system based on the echo state network, *J. Acoust. Soc. Am.*, vol. 135, pp. 2014, 2195-2195.
101. Rabiner, Lawrence R., and Biing-Hwang Juang. "An introduction to hidden Markov models." *ASSP Magazine, IEEE* 3.1 (1986): 4-16.
102. Huggins-Daines, David, et al. "Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices." *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on.* Vol. 1. IEEE, 2006.
103. Huang, Xuedong, et al. "The SPHINX-II speech recognition system: an overview." *Computer Speech & Language* 7.2 (1993): 137-148.
104. M. Kesarkar, Feature extraction for speech recognition, Electronic Systems, EE.Dept., IIT Bombay, , pp. 2003.
105. S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, FreeTech Books 2003.
106. C. Lee, Speech Recognition and Production by Machines, in J. D. Wright (ed.), *International Encyclopedia of the Social & Behavioral Sciences (Second Edition)*, Elsevier, Oxford, 2015.
107. Y. He, J. Han, T. Zheng, and G. Sun, A new framework for robust speech recognition in complex channel environments, *Digital Signal Processing*, vol. 32, pp. 2014, 109-123.
108. J. Hai and E. M. Joo, Improved linear predictive coding method for speech recognition: *Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim*

- Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on, vol. 3, pp. 1614-1618, IEEE, 2003.
109. M. Benzeghiba et al., Automatic speech recognition and speech variability: A review, *Speech Commun.*, vol. 49, pp. 2007, 763-786.
 110. B. Schuller, G. Rigoll, and M. Lang, Hidden Markov model-based speech emotion recognition: Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on, vol. 2, pp. II-1-4 vol.2, 2003.
 111. Jaeger, Herbert. "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note." *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report 148* (2001): 34.
 112. Venayagamoorthy, Ganesh K., and Bashyal Shishir. "Effects of spectral radius and settling time in the performance of echo state networks." *Neural Networks*22.7 (2009): 861-863.
 113. H. Meutzner, V. Nguyen, T. Holz, and D. Kolossa, Using automatic speech recognition for attacking acoustic CAPTCHAs: the trade-off between usability and security: Proceedings of the 30th Annual Computer Security Applications Conference, pp. 276-285, ACM, 2014.
 114. An implementation of Hidden Markov Models in Java. Available online at <https://code.google.com/p/jahmm/> . Accessed 12th January, 2015.

Appendix 1

Speaker 1									
Threshold: 0.67				Frame Length: >4 Frames					
Sample	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0X	0	0	0
2	0	0	0	0	0	0X	0	0	0
3	Won't load with >4 frames								
Threshold: 0.67				Frame Length: >5					
Sample	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0X	0	0	0
2	Overlap		0	0	0	0X	0	0	0
3	0	0	0	Overlap		0X	Overlap		0
Threshold: 0.81				Frame Length: >6					
Sample	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0X	0	0X	0
2	Overlap		0	0	0	0X	0	0	0
3	0	0	0	Overlap		0X	Overlap		0
Speaker 2									
Threshold: 0.81				Frame Length: >5					
Sample	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0X	0	0	0
2	0	0	0	0	0	0X	0	0	0
3	0	0	0	0	0	0X	0	0	0
Speaker 3									
Threshold: 0.81				Frame Length: >5					
Sample	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0X	0	0	0
2	0	0	0	0	0	0X	0	0	0
3	0	0	0	0	0	0X	0	0	0
Speaker 4									
Threshold: 0.67				Frame Length: >9					
Sample	1	2	3	4	5	6	7	8	9
1	0	0	0	Overlap		0X	0	0X	0
2	Overlap			Overlap					
3	Overlap			Overlap					
Threshold: 1.11				Frame Length: >9					
Sample	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0X	0	0X	0
2	Overlap		0	0	Overlap		0	0	0
3	Overlap			0	Overlap		Overlap		0
Threshold: 0.81				Frame Length: >5					
Sample	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0X	0	0X	0
2	0	0	0	0	0	0X	0	0X	0
3	0	0	0	0	0	0X	0	0	0
Speaker 5									
Threshold: 2.65				Frame Length: >5					
Sample	1	2	3	4	5	6	7	8	9

1	0	0	0	0	0	0X	0	0	0
2	0	0	0	0	0	0X	0	0	0
3	0	0	0	0	0X	0X	0	0	0
Threshold: 0.67					Frame Length: >5				
Sample	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0X	0	0	0
2	0	0	0	0	0	0X	0	0	0
3	0	0	0	0	0	0X	0	0	0

Table A1 Parameter modifications for Raw STE algorithm

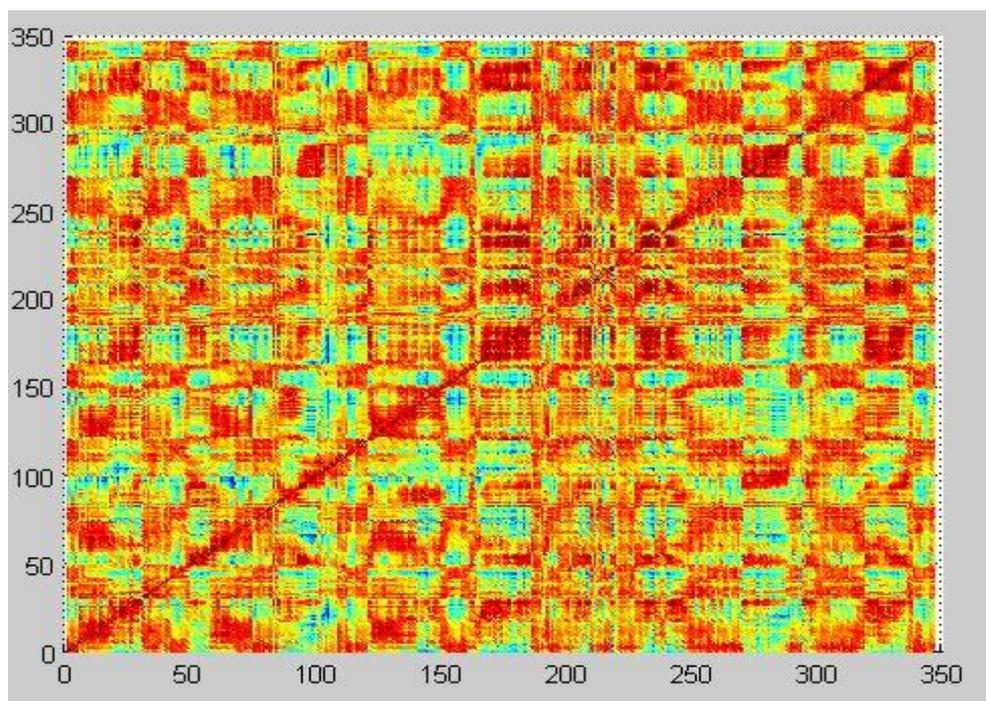


Figure A1 21 Order LPC Cross Correlation for speech signal with 44330 samples (346 frames)

Different parameter modifications for the VAD algorithm in chapter 4 for the raw STE are given in table A1. While figure A1 presents the result of a 21 order LPC LCC.

Independent Samples Test										
		Equality of Variances		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	of the Difference	
									Lower	Upper
R	Equal variances assumed	3.742	.058	11.782	58	.000	19.47333%	1.65279%	16.16492%	22.78175%
	Equal variances not assumed			11.782	50.430	.000	19.47333%	1.65279%	16.15432%	22.79235%
Group Statistics										
	Gender	N	Mean	Std. Deviation	Std. Error Mean					
R	Female	30	91.65%	5.01%	0.91%					
	Male	30	72.18%	7.54%	1.38%					

Table A11 – Independent samples test result for second novel VAD algorithm.

Appendix 2

Database	Hidden Layer 1	Hidden Layer 2	Feature	Learning Rate			Recognized
				Input to Hidden	Hidden to Hidden	Hidden to Output	
TIDIGITS	200	50	MFCC2	0.0002	0.0002	0.0001	92.97%
TIDIGITS	200	100	MFCC2	0.0002	0.0002	0.0001	89.67%
TIDIGITS	200	200	MFCC2	0.0002	0.0002	0.0001	92.50%
TIDIGITS	200	50	LPC2	0.0002	0.0002	0.0001	73.99%
TIDIGITS	200	100	LPC2	0.0002	0.0002	0.0001	73.07%
TIDIGITS	200	200	LPC2	0.0002	0.0002	0.0001	72.25%
TIDIGITS	300	50	MFCC2	0.0002	0.0002	0.0001	92.60%
TIDIGITS	300	100	MFCC2	0.0002	0.0002	0.0001	93.86%
TIDIGITS	300	200	MFCC2	0.0002	0.0002	0.0001	92.60%
TIDIGITS	300	300	MFCC2	0.0002	0.0002	0.0001	89.53%
TIDIGITS	300	50	LPC2	0.0002	0.0002	0.0001	72.18%
TIDIGITS	300	100	LPC2	0.0002	0.0002	0.0001	73.24%
TIDIGITS	300	200	LPC2	0.0002	0.0002	0.0001	73.14%
TIDIGITS	300	300	LPC2	0.0002	0.0002	0.0001	72.93%

Table A2 2 Hidden Layer tests for Dynamic MLP

As stated in chapter 5, 2 hidden layer tests were conducted for the TIDIGITS best performing hidden units to see if there was going to be any improvements. The improvements were not worthy of the extra computation time. **93.86%** was the best performance for the TIDIGITS database. Extra units were not used for the CSLU database as the performances were deemed considerable enough.