DEVELOPMENT OF A TEXT-INDEPENDENT AUTOMATIC SPEAKER RECOGNITION SYSTEM

by

TUMISHO BILLSON MOKGONYANE

DISSERTATION

Submitted in fulfilment of the requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

in the

FACULTY OF SCIENCE AND AGRICULTURE

(School of Mathematical and Computer Sciences)

at the

UNIVERSITY OF LIMPOPO

SUPERVISOR: Mr MJD Manamela

CO-SUPERVISOR: Dr TI Modipa

2021

Dedication

I dedicate this dissertation to my family.

DECLARATION

I declare that the **Development of a Text-Independent Automatic Speaker Recognition System** dissertation hereby submitted to the University of Limpopo, for the degree of **Master of Science (Computer Science**) has not previously been submitted by me for a degree at this or any other university; that it is my work in design and in execution, and that all material contained herein has been duly acknowledged.

Mokgonyane, TB (Mr)

23/04/2021
Date

Acknowledgements

First and foremost, I give thanks to the Almighty God for giving me the strength, wisdom and guidance throughout the course of the research work. I would also like to give appreciation to the following people for their contributions towards the completion and success of this research work:

- My supervisors, Mr MJD Manamela and Dr TI Modipa.
- My mentor, Tshephisho Joseph Sefara, for the moral support and guidance, without him this work would not have been a reality.
- The Deep Learning Indaba, Machine Learning Summer School South Africa, Deep Learning IndabaX South Africa, Black in AI and Centre for High Performance Computing for allowing me to attend their workshops, which helped advanced my skills and knowledge needed for this research study.
- My family and friends for showing me support through my studies.

Abstract

The task of automatic speaker recognition, wherein a system verifies or identifies speakers from a recording of their voices, has been researched for several decades. However, research in this area has been carried out largely on freely accessible speaker datasets built on languages that are well-resourced like English. This study undertakes automatic speaker recognition research focused on a low-resourced language, Sepedi. As one of the 11 official languages in South Africa, Sepedi is spoken by at least 2.8 million people. Pre-recorded voices were acquired from a speech and language national repository, namely, the *National Centre for Human Language Technology* (NCHLT), were we selected the Sepedi NCHLT Speech Corpus. The open-source pyAudioAnalysis python library was used to extract three types of acoustic features of speech namely, time, frequency and cepstral domain features, from the acquired speech data. The effects and compatibility of these acoustic features was investigated. It was observed that combining the three acoustic features of speech had a more significant effect than using individual features as far as speaker recognition accuracy is concerned. The study also investigated the performance of machine learning algorithms on low-resourced languages such as Sepedi. Five machine learning (ML) algorithms implemented on Scikit-learn namely, *K-nearest neighbours (KNN), support vector machines (SVM), random forest (RF), logistic regression (LR),* and *multi-layer perceptrons (MLP)* were used to train different classifier models. The GridSearchCV algorithm, also implemented on Scikit-learn, was used to deduce ideal hyper-parameters for each of the five ML algorithms. The classifier models were evaluated on recognition accuracy and the results show that the MLP classifier, with a recognition accuracy of 98%, outperforms KNN, RF, LR and SVM classifiers. A graphical user interface (GUI) is developed and the best performing classifier model, MLP, is deployed on the developed GUI intended to be used for real-time speaker identification and verification tasks. Participants were recruited to the GUI performance and acceptable results were obtained.

TABLE OF CONTENTS

List of Figures

List of Tables

List of Code Snippets

List of Abbreviations

FAR	False Acceptance Rate
FRR	False Rejection Rate
GUI	Graphical User Interface
KNN	K-Nearest Neighbours
LR	Logistic Regression
MFCC	Mel-Frequency Cepstral Coefficient
ML	Machine Learning
MLP	Multi-Layer Perceptron
MOS	Mean Opinion Score
NCHLT	National Centre for Human Language Technology
RF	Random Forest
RMSE	Root Mean Squared Error
SVM	Support Vector Machine
TAR	True Acceptance Rate
TRR	True Rejection Rate

List of Publications

1. Sefara, T.J., and **Mokgonyane, T.B**., and Marivate, V. "*Practical Approach on implementation of Wordnets for South African languages*," in Proceedings of the 11th Global Wordnet Conference. University of South Africa (UNISA). pp. 20-25. 2021
2. Sefara, T. J., and **Mokgonyane, T.B**. "*Emotional Speaker Recognition based on Machine and Deep Learning,"* In 2020 2nd International Multidisplinary Information Technology and Engineering Conference (IMITEC), pp. 1-8. 2020
3. **Mokgonyane, T.B.**, Sefara, T. J., Manamela, M. J., Modipa, T. I., and Masekwameng, M. S. "*The Effects of Acoustic Features of Speech for Automatic Speaker Recognition,"* In 2020 International Conference on Advances in Big Data, Computing and Data Communication Systems. Durban, Kwa-Zulu Natal, South Africa, pp. 210-214.
4. Masekwameng, M. S., **Mokgonyane, T.B.**, Modipa, T. I., Manamela, M. J., and Mogale, M.M. "*Effects of Language Modelling for Sepedi-English Code-Switched Speech in Automatic Speech Recognition System,"* In 2020 International Conference on Advances in Big Data, Computing and Data Communication Systems. Durban, Kwa-Zulu Natal, South Africa**,** pp. 205-209.
5. **Mokgonyane, T.B.**, Sefara, T. J., Modipa, T. I., and Manamela, M. J. "*Automatic Speaker Recognition System based on Optimised Machine Learning Algorithms,"* In IEEE AFRICON 2019, Accra, Ghana, pp. 1-7.
6. Mogale, M.M., Sefara, T.J., **Mokgonyane, T.B.,** Manamela, M.J., and Modipa, T.I., "*Grammar-driven Text-to-speech Application for Articulation of Mathematical Expressions"*. In Southern African Telecommunication and Networks and Application Conference (SATNAC) 2019. Fairmont Zimbali Resort, Kwa-Zulu Natal, South Africa.
7. Sefara T.J., **Mokgonyane, T.B.,** Modipa, T.I., and Manamela, M.J., "*HMM-based Speech Synthesis System incorporated with Language Identification for Low-resourced Languages"*. In 2019 International Conference on Advances in Big Data, Computing and Data Communication Systems. Drakensberg Sun Resort, KZN, South Africa, pp. 96-101
8. **Mokgonyane, T.B.,** Sefara T.J., Modipa, T.I., and Manamela, M.J., "*The Effects of Data Size on Text-Independent Automatic Speaker Identification System"*. In 2019 International Conference on Advances in Big Data, Computing and Data Communication Systems. Drakensberg Sun Resort, KZN, South Africa, pp. 192-197
9. **Mokgonyane, T.B.,** Sefara, T.J., Modipa, T.I., Mogale, M.M., Manamela, M.J., & Manamela, P.J., "*Automatic Speaker Recognition System based on Machine Learning Algorithms"*. In 2019 SAUPEC/RobMech/PRASA Conference. Bloemfontein, South Africa, pp. 141-146.
10. **Mokgonyane, T.B.,** Sefara, T.J., Manamela, M.J. & Modipa, T.I., "*Development of a Text-Independent Speaker Recognition System for Biometric Access Control"*. In Southern African Telecommunication and Networks and Application Conference. Arabella, Western Cape, South Africa, pp. 128-133.
11. Manamela, P.J., Manamela, M.J., Modipa, T.I, Sefara, T. J. & **Mokgonyane, T. B.,** "*The Automatic Recognition of Sepedi Speech Emotions based on Machine Learning Algorithms"*. In International Conference on Advances in Big Data, Computing and Data Communication Systems. Durban, South Africa, pp. 507-513
12. **Mokgonyane, T. B.,** Sefara, T. J., Manamela, P. J., Manamela, M. J. & Modipa, T. I., "*Development of a Speech-enabled Basic Arithmetic m-Learning Application for Foundation Phase Learners"*. In 2017 IEEE AFRICON. Cape Town, South Africa, pp. 794-799.

CHAPTER 1: INTRODUCTION

1.1 Introduction

Biometric recognition is the task of automatically granting access or permission to services by capturing, analysing and comparing some of a human being's behavioural and physiological attributes (Adamski, 2013). The physiological attributes include fingerprints, a human face, an iris, a palm and a voice. A human voice is a biometric attribute that, opposed to other biometric attributes such as fingerprints and faces, is not yet commonly used for person identification. Automatic voice recognition (also known as speaker recognition) is a method of access control whereby a system uses a recording of a speaker's voice to validate or determine the identity of that speaker. Ongoing research in the speaker recognition field has stretched over 50 years now with significant progress made in improving the performances through the application of more effective algorithms (Hashimoto *et al.*, 2016, Marciniak *et al.*, 2014, Furui, 2005). Due to significant progress that has been made in the artificial intelligence field, the speaker recognition technologies have taken a new path and over recent years, this technology has evolved to become a low-cost and effective solution to automated identification of individuals.

A human voice is a biometric attribute that depends heavily on the speaker who uttered it. Several studies have reported that no two people's voices sound precisely identical (Gbadamosi, 2013, Kinnunen & Li, 2010). The acoustic aspects of the distinctions between human voices are uncertain and not easy to differentiate from the signal aspects representing the segments recognition (Charan *et al.*, 2017). Three sources of variation between speakers exist, according to Ramachandran *et al*. (2002), and these are *(1) differences in speaking styles including the speaker's accent; (2) differences between vocal chords and forms of vocal tracts*; and *(3) differences in speaker expressions when communicating a specific meaning (words or phrases they use)*. The human voice is a very powerful tool due to these sources of variation among speakers and can thus be used in security systems (Singh *et al.*, 2012). It is easy to measure and to compare the physical characteristics of a speech signal as compared with other biometric features such as fingerprints, face, iris and DNA (Casserly & Pisoni, 2013). The characteristics of a speech signal are also very well-known and simple to use (Furui, 2005), with several efficient algorithms available to work with them (Hashimoto *et al.*, 2016, Marciniak *et al.*, 2014). In signal processing, speaker recognition is a very important field and it has a number of applications, especially in security systems (Singh *et al.*, 2012). Some general speaker recognition applications include control on the use of credit cards, protection of confidential information, verification of customers for telephone banking, forensics, surveillance and remote computer access (Ramachandran *et al.*, 2002).

1.2 Problem Statement

In today's modern world, computer-based technologies (i.e., the ways people interact with each other, the vehicles they drive, the equipment they own, the medical facilities they visit, or the places they live and work around) help or affect nearly any area of life. The digital age has brought about a security concern in people’s living spaces, and thus information security and access control are currently the most interesting areas of research. Several access control approaches have been proposed and they include knowledge-based approach (the use of usernames and passwords), token-based approach (the use of smart cards, passports, driver’s licenses and insurance cards), and biometric recognition approach. Biometric recognition is the most effective approach for access control since it is based on a part of an individual – a measureable physiological or behavioural feature, which is often more difficult to fake, steal or imitate than a password or a token. Users do not have to remember it and cannot forget it at home by accident (Siddique *et al.*, 2017). However, of all the biometric characteristics, voice is the only biometric characteristic that allows users to authenticate remotely. With over 50 years of research in the automatic speaker recognition field, no research attempts have been made in developing automatic speaker recognition systems using data collected from speakers of the low-resourced South African languages. It is therefore not known whether speaker recognition is possible with regards to these languages and the effects of these language towards the performance of speaker recognition systems is not known. The South African official languages can also be listed amongst the particularly low-resourced languages, according to de Wet *et al.* (2016). As one of the South African official language, the Sepedi language is reported to be a language with more than 2.8 million speakers and spoken by most residents in the Limpopo province, South Africa

(Census, 2011). This research study proposes to develop an automatic speaker recognition system for recognition of native speakers of the Sepedi language.

1.3 Research Questions

The main research questions of the study are as follows:

a) Can a speaker recognition system give a significant performance if trained with data collected from speakers of low-resourced languages?
b) What is the effect of a particular spoken language towards the performance of a speaker recognition system?

1.4 Aim and Objectives

This study aims to train and develop a speaker recognition system uses the speaker's voices to verify and identify the speaker's identities in order to allow only the speakers who are identified or verified the right to access information systems, devices or services that have to be secured from unauthorized users. In response to achieving the aim of the study, we have set out the following research objectives:

a) To acquire pre-recorded Sepedi speech data from publicly available speaker recognition databases.
b) To extract acoustic features of speech from the acquired speech data.
c) To train speaker classifier models using machine learning algorithms and compare their performances to select the best performing model.
d) To deploy the best performing speaker classifier model that determines speaker identities and to verify the claimed speaker identities.
e) To develop a graphical user interface that performs real-time automatic speaker recognition capabilities.

1.5 Scientific Contribution

This research study intends to develop an automatic speaker recognition system for recognition of native speakers of the Sepedi language to investigate the significance of a spoken language towards the performance of the speaker recognition system.

This research study is also intended to show that automatic speaker recognition is possible with low-resourced languages and the contributions are listed as follows:

- This study identifies the lack of tools for South African indigenous languages for speech and language processing, such as speaker recognition databases.
- This study experiments on one South African low-resource language (Sepedi) which is widely spoken in the Limpopo province by approximately 52.9% of the people (Census, 2011).
- The findings of this study contributes more towards a broad understanding of automatic speaker recognition research in the context of low-resourced languages and also how the automatic speaker recognition technologies can be ported and adapted to other South African low-resource languages.
- This research project develops an automatic speaker recognition system that can serve to enhance authentication methods in many computer-based systems.

1.6 Ethical Considerations

This research study does not include any sensitive or personal data that may endanger humans if disclosed.

1.7 The Dissertation Arrangement

This dissertation is arranged as follows:

- Chapter 2 presents a theoretical background highlighting literature from previous studies on speaker recognition.
- Chapter 3 presents the methodology covering the tools used to conduct the experiments, the dataset, feature extraction techniques, feature normalisation, training and evaluation of the classifier models.
- Chapter 4 presents the system implementation covering the system flow chart, database design and the graphical user interface.
- Chapter 5 reports the experimental results.
- Chapter 6 concludes the dissertation highlighting the limitations, contributions, recommendations and future work.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This chapter discusses the speaker recognition theoretical background covering the fundamentals of speaker recognition, acoustic features of speech and supervised machine learning algorithms.

2.2 Fundamentals of speaker recognition

This section discusses the fundamentals, categories, applications and phases of speaker recognition systems.

2.2.1 Verification and identification systems

Speaker *verification* and *speaker identification*, depicted in Figure 2.1, are the two fundamental tasks of speaker recognition. Speaker verification is the task of deciding whether the test speaker's voice belongs to a certain enrolled speaker. In this case, the test speaker makes an identity claim first and the speaker verification system decides whether the identity claim made is correct or incorrect, in which the identity claim will be accepted if it is correct or rejected if incorrect. Speaker verification system's potential applications of include telephone banking, remote computer log-in and telephone fraud prevention (Reynolds, 1995). Speaker identification is the task of deciding the identity of the test speaker (user) from a collection of enrolled speakers (user does not make a prior identity claim). Speaker identification systems are used in application areas such as forensics, automatic labelling of speakers from recorded meetings and surveillances (Reynolds, 1995).

Depending on the range of operation, a speaker identification system can be classified as either *open-set identification systems* or *closed-set identification systems* (Kekre & Kulkarni, 2013). With closed-set identification systems, every speaker has to be enrolled in a speaker database and the test speaker is selected to be the speaker with the closest match to the test speech signal. However, with open-set identification systems, not all speakers are enrolled in a speaker database. In this case, the system therefore carries out an extra task of rejection in the event that the test speaker is not enrolled in the speaker database (Kekre & Kulkarni, 2013).

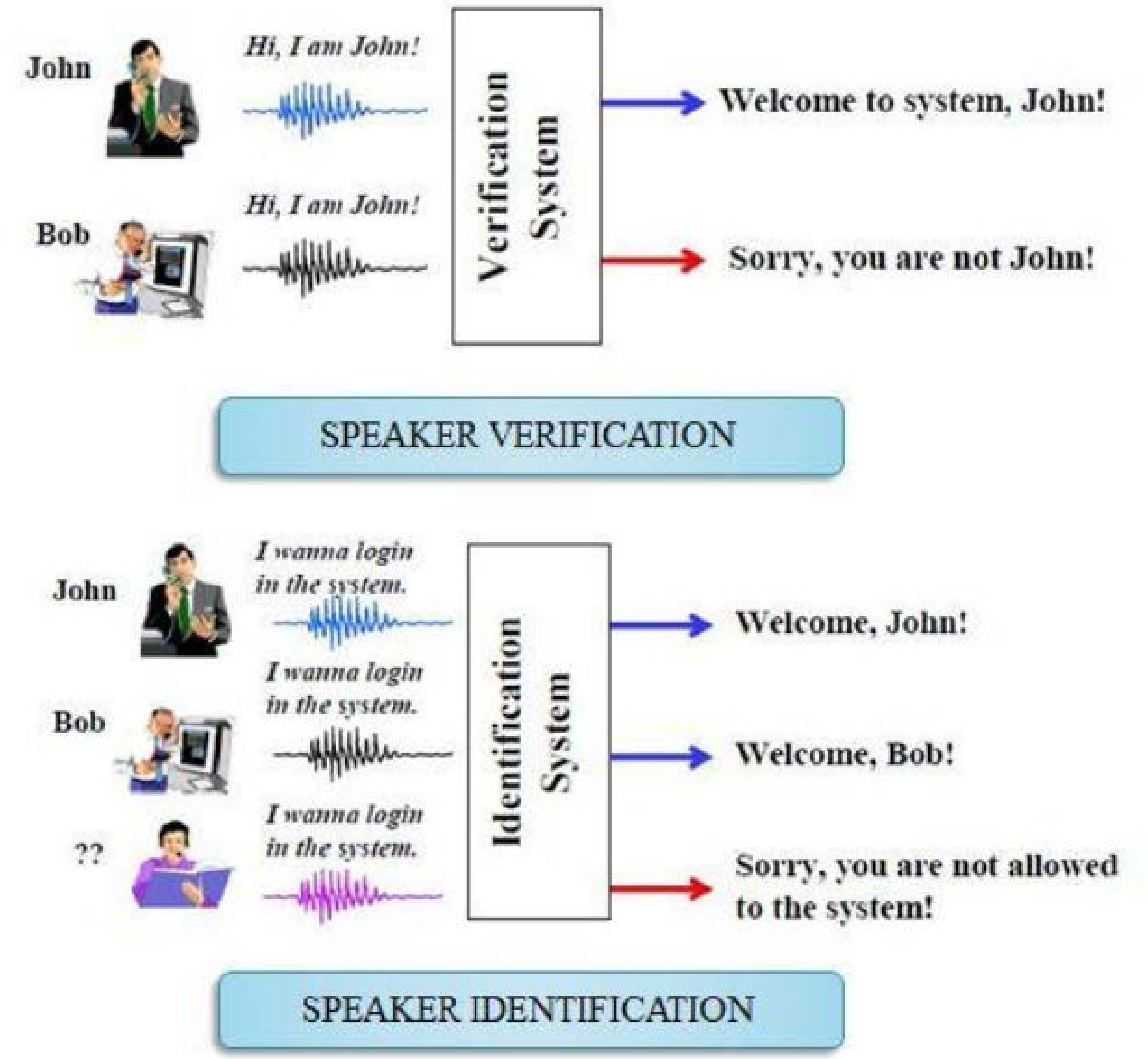


Figure 2.1 Speaker Recognition fundamental tasks (Panda et al.*, 2011).*

2.2.2 Text-dependent and Text-independent systems

Speaker recognition systems are categorised according to the constraints which are set on the input text of the speech that is used to train and test the speaker recognition system, where the categories consist of text-dependent systems and text-independent systems. For text-dependent systems, the phrase spoken or the input text used is fixed for each speaker whereas for text-independent systems, the phrase spoken or the input text is not fixed (Liu *et al.*, 2015). Text-dependent systems are mostly used in occasions where the users are known to be cooperative, while text-independent systems are generally used in occasions where users are known to be non-cooperative, since such users do not specifically wish to be recognised (Kinnunen & Li, 2010). Compared to text-independent systems, text-dependent systems are reported to achieve better recognition performances (Ramachandran *et al.*, 2002). However, the growing trend in the development of systems is to build text-independent systems because of the versatility these systems offer (Bimbot *et al.*, 2004). This study considers a text-independent speaker recognition system for closed-set identification.

2.2.3 Applications of Speaker Recognition Systems

Speaker recognition has the intent of automatically recognising a speaker given a speech sample. Speaker recognition research has continued for over 50 years now and continues to show impressive results (Sahoo & Rishi, 2014, Jain *et al.*, 2016). Although speaker recognition technologies are applicable to a wide area of applications, authentication, surveillance and forensics are the three main application areas of speaker recognition (Singh *et al.*, 2012).

2.2.3.1 Authentication

Speaker recognition for authentication enables automated systems to authenticate a person from his or her speech sample, this task is referred to as *biometric person authentication*. To enhance access control methods such as using usernames and passwords (knowledge-based) or using physical tokens such as keys and identification cards (token-based), users can be authenticated and granted access to devices, information systems or to knowledge systems safely and securely through the use of biometric person authentication (Ferbrache, 2016, Hamid, 2015).

2.2.3.2 Surveillance

Speaker recognition technology can also be used for surveillance (Kiktova & Juhar, 2015). Monitoring conversations in a communication network is one of the key tools in the counter-terrorism and espionage scenario or situations. The goal may be either tracking known criminals, terrorists, spies or even tracking suspicious internet-based conversations threatening state security (Solewicz & Koppel, 2005).

2.2.3.3 Forensic

The most important application area supported by the use of speaker recognition technologies is forensics (Ramachandran *et al.*, 2002). In telephone conversations, a lot of crucial information (evidence) can be exchanged between participants (law-abiding or not) (Gbadamosi, 2013). During a crime commission, where there is a reported speech sample, the suspect’s voice can be matched with the reported speech samples to help solve the committed crime. By proving the identification of the speaker on the reported speech sample, an innocent suspect to be discharged or a guilty suspect can help convicted in a court of law accused of committing a crime.

2.2.4 Phases of Speaker Recognition Systems

Figure 2.2 depicts the two distinct speaker recognition phases, the enrolment (or training) and recognition (or testing) phase. The speaker's voices are recorded in the enrolment phase and a number of acoustic features of speech (discussed in Section 2.4) are extracted from the recorded voice. After the acoustic features of speech are extracted, machine learning (ML) algorithms (discussed in Section 2.5) learn the feature patterns and create a classifier model that classifies among the speakers.

In the recognition phase, the test speech signal is recorded and several acoustic features are extracted and compared against the previously trained classifier model which determines the identity of the speaker the test speech is recorded signal from.

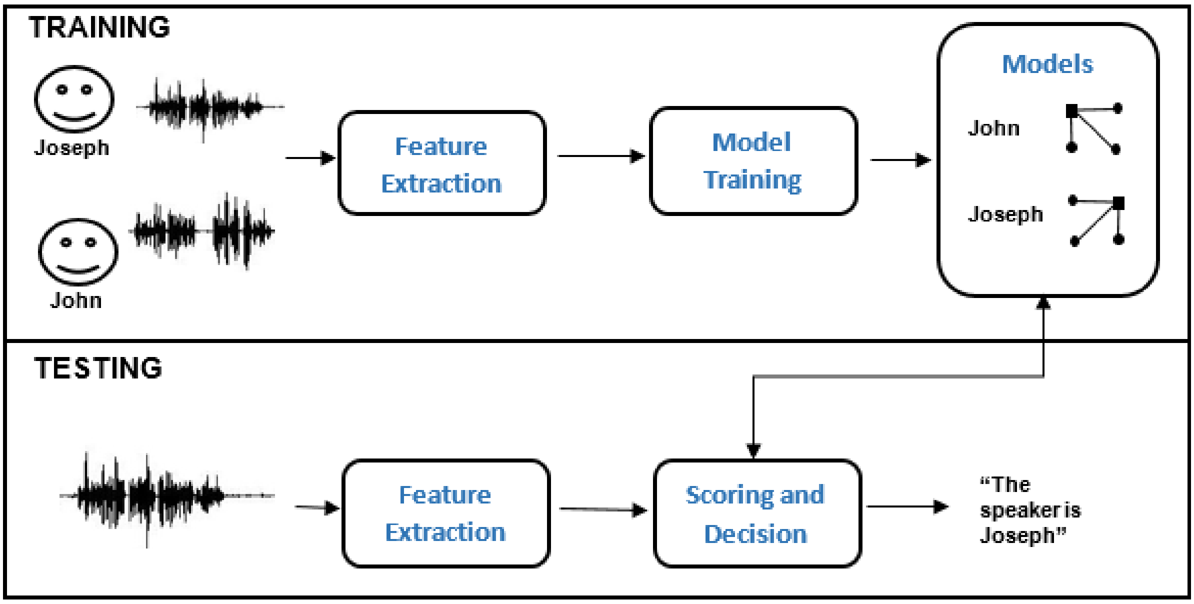


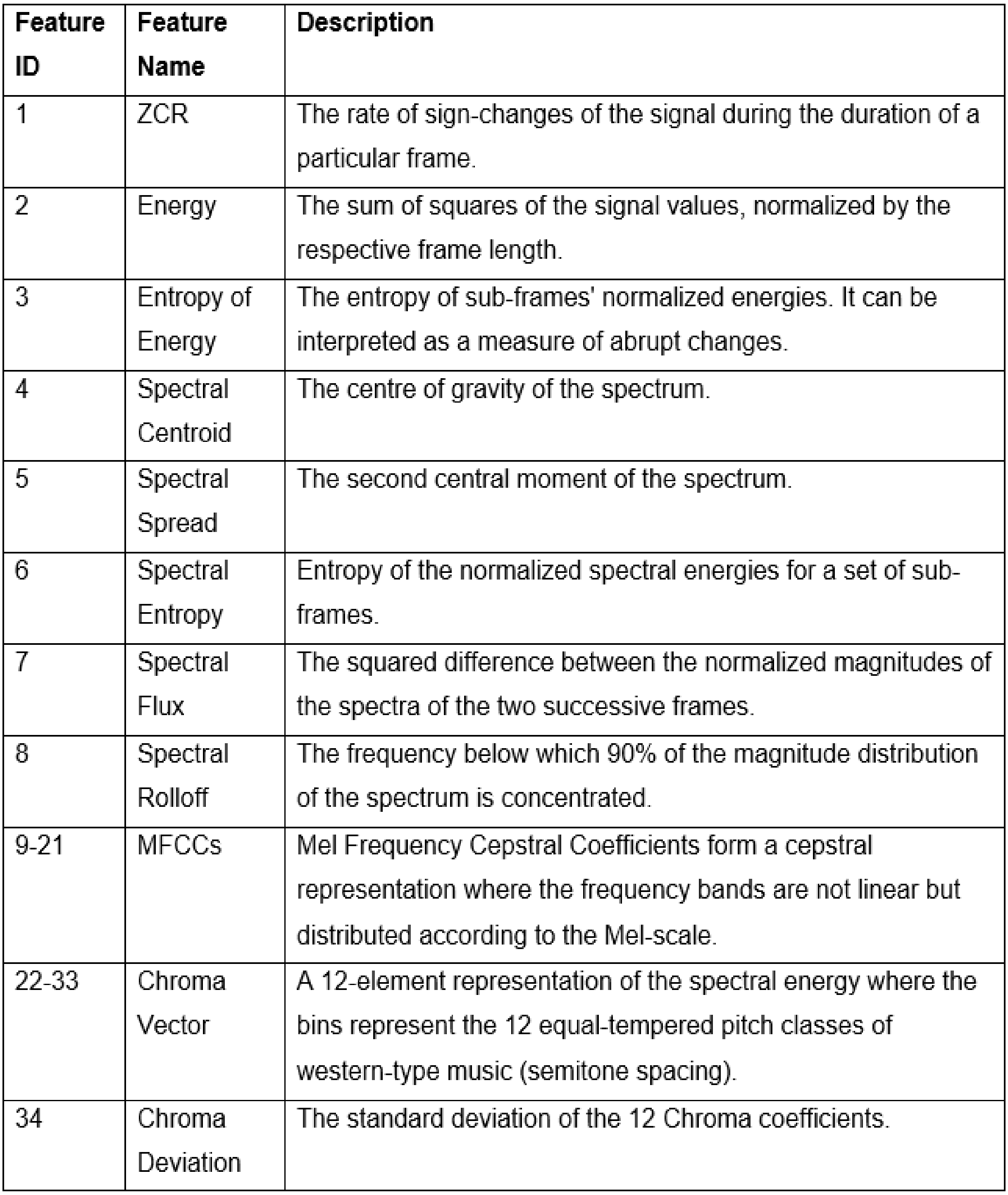
Figure 2.2 Phases of Speaker Recognition System.

2.3 Acoustic Features of Speech

Several acoustic features of speech which have the potential to uniquely differentiate among speakers are contained in the human voice. The performance (accuracy) of speaker recognition systems varies depending on the choice of acoustic features of speech that the speaker recognition system extracts from the speech signals.

Table 2.1 describes some of the available acoustic features of speech which include time-domain features, frequency-domain features and cepstral-domain features (Giannakopoulos, 2015). The time-domain features (Feature IDs 1-3) are features extracted directly from raw audio samples (Bachu *et al.*, 2010). The frequency-domain features (Feature ID 4–34, except the MFCCs) are based on the magnitude of the

Table 2.1 Acoustic features of speech on short-term windows



Feature ID	Feature Name	Description
1	ZCR	The rate of sign-changes of the signal during the duration of a particular frame.
2	Energy	The sum of squares of the signal values, normalized by the respective frame length.
3	Entropy of Energy	The entropy of sub-frames' normalized energies. It can be interpreted as a measure of abrupt changes.
4	Spectral Centroid	The centre of gravity of the spectrum.
5	Spectral Spread	The second central moment of the spectrum.
6	Spectral Entropy	Entropy of the normalized spectral energies for a set of sub-frames.
7	Spectral Flux	The squared difference between the normalized magnitudes of the spectra of the two successive frames.
8	Spectral Rolloff	The frequency below which 90% of the magnitude distribution of the spectrum is concentrated.
9-21	MFCCs	Mel Frequency Cepstral Coefficients form a cepstral representation where the frequency bands are not linear but distributed according to the Mel-scale.
22-33	Chroma Vector	A 12-element representation of the spectral energy where the bins represent the 12 equal-tempered pitch classes of western-type music (semitone spacing).
34	Chroma Deviation	The standard deviation of the 12 Chroma coefficients.

Discrete Fourier Transform. Lastly, the cepstral-domain features (Mel-Frequency Cepstral Coefficients or MFCCs) results after the Inverse Discrete Fourier Transform is applied on the logarithmic spectrum (Tiwari, 2010).

MFCCs are popular features extracted from speech signals for use in recognition tasks. In the source-filter model of speech, MFCCs are understood to represent the filter (vocal tract). MFCCs are determined with the help of a psychoacoustically motivated filter bank, followed by logarithmic compression and discrete cosine transform. Suppose the outputs of an M-channel filterbank is $Y(m), m = 1, \ldots, M$, is the MFCCs are obtained using the following equation:

$$c_n = \sum_{m=1}^{M} [LogY(m)] \cos\left[\frac{\pi n}{M}\left(m - \frac{1}{2}\right)\right] \tag{1}$$

where n is the index of a cepstral coefficient (Huang *et al.*, 2001).

2.4 Machine Learning Algorithms

The performance of each automatic speaker recognition system is highly dependent on the type of machine learning (ML) algorithm used to train its classifier model (Baharipour *et al.*, 2014). This section discusses the ML algorithms considered in this study, namely, K-nearest neighbours, logistic regression, support vector machines, random forest and multi-layer perceptrons.

2.4.1 K-Nearest Neighbours

The K-nearest neighbours (KNN) algorithm is a type of a lazy learning or instance-based learning algorithm which only approximates functions locally and defers all computation until classification (Aha *et al.*, 1991). The KNN classifier is a non-parametric classification method (or regression) that classifies unknown instances in the feature space based on the k closest training examples (k is a preferably small positive integer). If $k = 1$, the unknown instance is assigned to the class of the closest single nearest neighbour (Aha *et al.*, 1991). Several methods that apply the KNN algorithm for their speaker recognition activities are available in the literature (Charan *et al.*, 2017, Rajalakshmi & Anju, 2017, Sreelekshmi & Syama, 2017, Ranny, 2016, Kacur *et al.*, 2011).

Charan *et al.*, (2017) reports that the performances for speaker recognition is obtained when using KNN classifier and MFCC features. The authors used feature selection techniques such as LPCCs (linear predictive cepstral coefficient), MFCCs and PLPs (perceptual linear prediction), and used SVM, feed-forward, KNN and decision tree algorithms for classification blocks in speaker recognition and analysed each block to determine the best feature selection technique.

Sreelekshmi and Syama (2017) used KNN classifier employing MFCC and formants as features for speaker identification purposes. The authors proposed a new feature extraction technique for speaker identification that is based on Formants, MFCCs and KNN classifier and they have observed that, formants contributed greatly to yield a relatively good accuracy rate. The idea of using a KNN classification technique was also valid in the study of Ranny (2016). The author used MFCCs in the study and KNN classifier with one nearest neighbour. The study used 11 participants for data training and data testing where each participant's voice was recorded three times. The recognition accuracy obtained in the study was 84.85%. The recognition accuracy was further improved by applying a double distance (2 nearest neighbours) measurement which gave an almost state-of-the-art accuracy of 96.97%. Another study which used KNN with one nearest neighbour was conducted by Rajalakshmi and Anju (2017). This study was conducted on ten speakers and their voice recordings were divided into train sets and test sets. The authors used MFCC and PLP features and a recognition accuracy ranging from 80-95% was obtained for all the speakers using train sets and a recognition accuracy ranging from 50-75% was obtained for all the speakers using test sets. Kacur *et al.*, (2011) motivates using KNN classifiers for automatic speaker recognition tasks and reports that with a 6% improvement, the KNN classifier trained with $k = 4$ obtains best performances.

2.4.2 Random Forest

The random forest (RF) algorithm is a supervised classification and regression algorithm that works by constructing a number of decision trees at the training stage, generating the class which is the class mode (classification) or mean prediction (regression) of individual trees (Breiman, 2001). Due to the number of decision trees involved in the procedure, the RF algorithm is a highly precise and robust method. This algorithm does not suffer from overfitting problems in most cases, because it cancels out the biases by taking the mean of all predictions. Rao *et. al.,* (2020) has successfully applied the RF algorithm speaker recognition tasks and reports to have obtained *state-of-the-art* performances.

2.4.3 Support Vector Machine

The support vector machine (SVM) is a supervised learning model with associated learning algorithms that analyse data and recognise patterns and can be used for both regression and classification (Chang & Lin, 2011). In speaker recognition field, the SVM classifiers are very popular and are reported to have achieved the best recognition performance (Sahoo & Rishi, 2014). In addition, the SVM classifiers are common classifiers proven to be powerful pattern classification techniques which model the boundaries between one speaker and a group of impostors (Sahoo & Rishi, 2014).

The use of SVM classifiers in speaker recognition has been proven to be valid in the study of Staroniewicz and Majewski (2004) where the authors presented a test results of speaker identification system based on the Support Vector Machines. In their study, the usefulness of SVM classifier for large voice telephone quality database (1300 speakers) was examined and the authors observed that the SVM classifier showed its ability of feature generalization for large sets of classes. The study obtained high scores (around 90%) of speaker identification and reported that the results did not change significantly when the number of tested voices increased.

Kamruzzaman *et al.,* (2010) presented a technique for text-dependent speaker identification using MFCC-domain support vector machine (SVM). The authors first used sequential minimum optimization learning technique for SVM that improve performance over traditional techniques. The authors computed the cepstrum coefficients representing the speaker characteristics of a speech segment by a non-linear filter bank analysis and discrete cosine transform. The authors observed that extensive experimental results on several samples show the effectiveness of the proposed approach.

2.4.4 Logistic Regression

The logistic regression (LR) algorithm is a classification model for linear analysis instead of regression analysis and it is a very robust and accurate approach that uses *multinomial logistic regression* to generalise logistic regression to multi-class problems. (Harrell, 2015). Since LR has fewer parameters and has a regularisation parameter that handles problems of overfitting, it hardly suffers from the overfitting problems. The LR classifier and its sparse version of the kernel logistic regression method have been reported to outperform the SVMs and the Gaussian mixture models text-independent speaker recognition (Katz *et al.*, 2006).

2.4.5 Multi-layer Perceptron

A multi-layer perceptron (MLP) classifier is an artificial neural network classification model that maps sets of input data onto a set of appropriate outputs. There are multiple layers in an MLP classifier and each layer is connected to the following one. The layer nodes are neurons which use non-linear activation functions, except for the input layer nodes (Richardson *et al.*, 2015).

The recent approaches of applying neural networks for speaker recognition tasks have been reported to be successful (Wang & Lawlor, 2017). Wang and Lawlor (2017) trained a speaker recognition system using neural networks classifier with MFCC features and have observed that the recognition rate decreases when the number of speakers is increased. Therefore, as the number of speakers increased, the response was to increase the number samples per speaker. Dey *et al.*, (2012) was successful in using neural networks and Hidden Markov models to train a speech and speaker recognition system. Chauhan and Chandra (2017) applied feed forward artificial neural network to conduct a comparative study between various combinations of features for speaker identification. To build a text-independent speaker recognition system that accomodates both identification and verification tasks, Fenglei and Bingxi (2000) used binary neural networks using an MLP model.

2.5 Summary

This chapter discussed the theoretical background on automatic speaker recognition. The chapter outlined the differences between speaker identification and verification systems which are the fundamental tasks of speaker recognition, the differences between text-dependent and text-independent systems which are the categories of speaker recognition and the applications of speaker recognition (authentication, surveillance and forensics). The chapter also discussed the phases of speaker recognition, acoustic features of speech and the five machine learning algorithms considered in this study. The following chapter discusses the methodology and experimental setup of the study.

CHAPTER 3: METHODOLOGY

3.1 Introduction

This chapter discusses the research methodology and experimental setup of this study. The chapter first discusses the research design, followed by the tools and packages used to set the experiments. The data acquired is prepared accordingly within the different directories for each speaker (Section 3.4). Acoustic features of speech are extracted from each sample in the data directories and feature normalisation is performed (Sections 3.5 and 3.6). After extracting the acoustic features of speech, Scikit-learn is launched and used to train different classifier models and parameter optimisation is performed to find the best hyper-parameters (Sections 3.7). Lastly, evaluation is performed to select the best performing model. The tools discussed in Section 3.3 are installed and set up on Ubuntu 18.04 operating system.

3.2 Research Design

This study adapted an experimental design methodology encompassing both a quantitative and qualitative paradigm. This design was chosen because the study focuses on the experimentation with different machine learning techniques. Again in this study, we recruited participants to evaluate the developed speaker recognition system.

3.2.1 System Architecture

The proposed automatic speaker recognition system flow diagram is shown in Figure 3.1 which depicts two phases, enrolment and identification/verification. In the enrolment phase, audio samples for each speaker are recorded and stored in the speaker database, then acoustic features of speech are extracted from each audio sample. The extracted acoustic features of speech are the used to train machine learning algorithms which produces a speaker classifier model. The model is saved on the computer and will be deployed for prediction in the identification/verification phase. In the identification/verification phase, a new speech sample (test sample) of a single speaker is recorded and feature extraction is then performed. The extracted acoustic features of speech are compared against the previously trained speaker classifier model to identify or verify the test speaker.

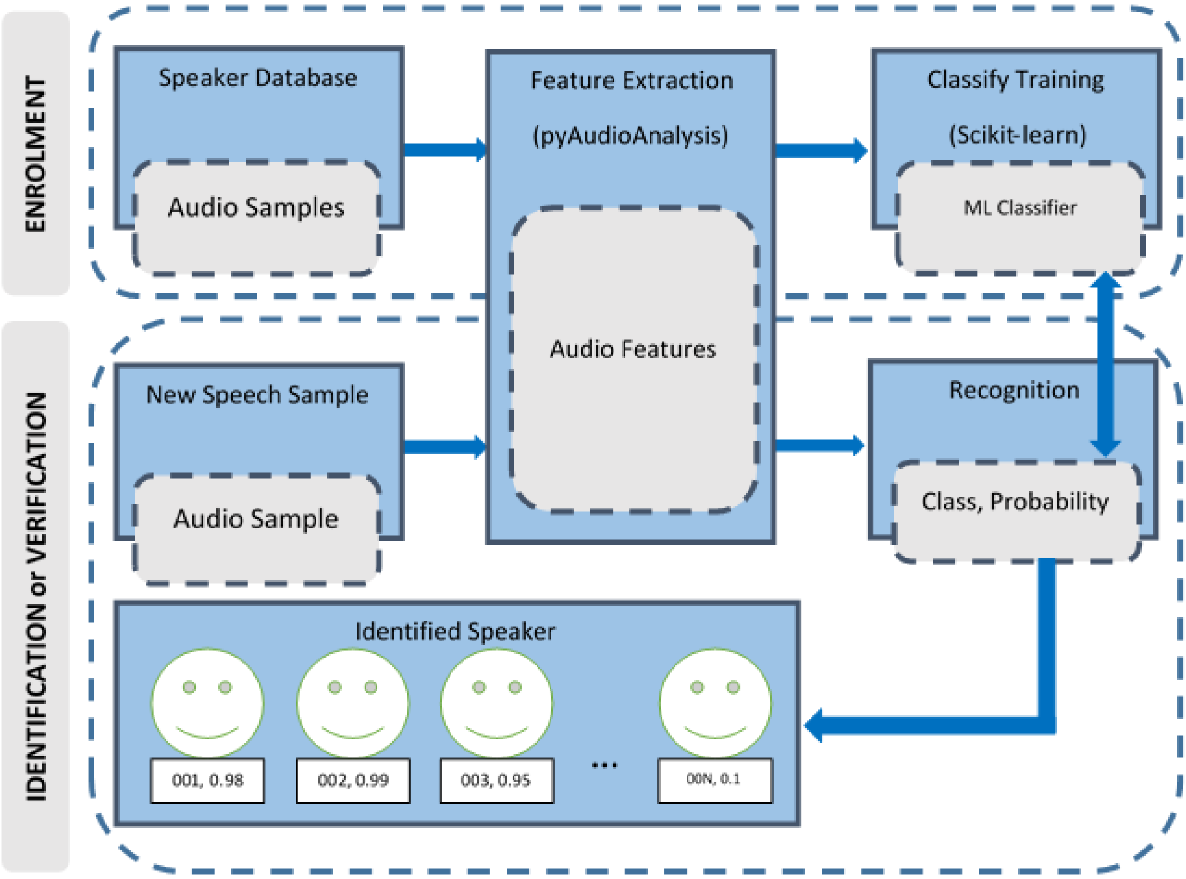


Figure 3.1 A flow diagram showing the speaker recognition system phases

3.2.2 Population specification

The criteria used to select participants was as follows:

- The individual should be a Sepedi language native speaker.
- The individual must voluntarily be willing to participate.

3.2.3 Data Collection

Data collection refers to a procedure followed, in a defined systematic manner, to collect and quantify information on variables of interest. The two types of data are, *primary data* which is the raw data or data collected from the original source and *secondary data* which is the data that is already collected by someone else and not the user, that is, data already available and analysed by someone. Both primary data and secondary data were examined in this research study.

3.2.3.1 Primary data

This study collected primary through a questionnaire (evaluation form) distributed to the respondents. The respondents were informed about the research work, and trained to answer the questions. This was conducted at the stage of evaluation (See Chapter 4).

3.2.3.2 Secondary data

This study acquired secondary data of pre-recorded voices from the National Centre for Human Language Technology project (See Section 3.4).

3.2.4 Data Analysis

Python statistical techniques were used to analyse the responses (primary data) and findings are presented in the form of pie charts plotted with *matplotlib* library. Acoustic features of speech are extracted from the pre-recorded voices (secondary data) and ML algorithms are used to study the patterns on these features.

3.3 Tools and Packages

The following tools and packages (installed and set up on Ubuntu 18.04 operating system) have been used to perform the experiments in this study:

- **Anaconda** - Anaconda[1] is a Python and R data science distribution and a package manager that puts together over 1,500+ open source packages. This study uses Python3[2] as the main programming language. From the Anaconda's collection of open source packages, we use the following packages: *pandas*, *numpy*, *Scikit-learn*, *matplotlib* and *PyQt.*
- **pyAudioAnalysis** - pyAudioAnalysis is an open-source library developed in Python. This library offers a wide variety of audio-related functions such as feature extraction, segmentation, classification and visualisation (Giannakopoulos, 2015). This library was acquired for feature extraction and visualisation capabilities.
- **Scikit-learn** - Scikit-learn (sklearn) is a Python programming language open-source machine learning platform that offers an easy way to conduct data

[1] https://docs.anaconda.com/anaconda/

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

mining and analysis (Pedregosa *et al.*, 2011, Shashidhara *et al.*, 2015). This tool was acquired for training the classifier models.

- **PyQt4** - PyQt4 is a toolkit used for creating application's graphical user interfaces. It is a blending of Python programming language and the successful Qt[3] library. This tool is acquired for the GUI development (see Chapter 4).
- **SQLite** - SQLite[4] is an in-process library implementing a transactional SQL database engine that is self-contained, zero-configuration and server-less. The GUI runs SQLite3 database in the background to store the user's biographical information.

3.4 Dataset

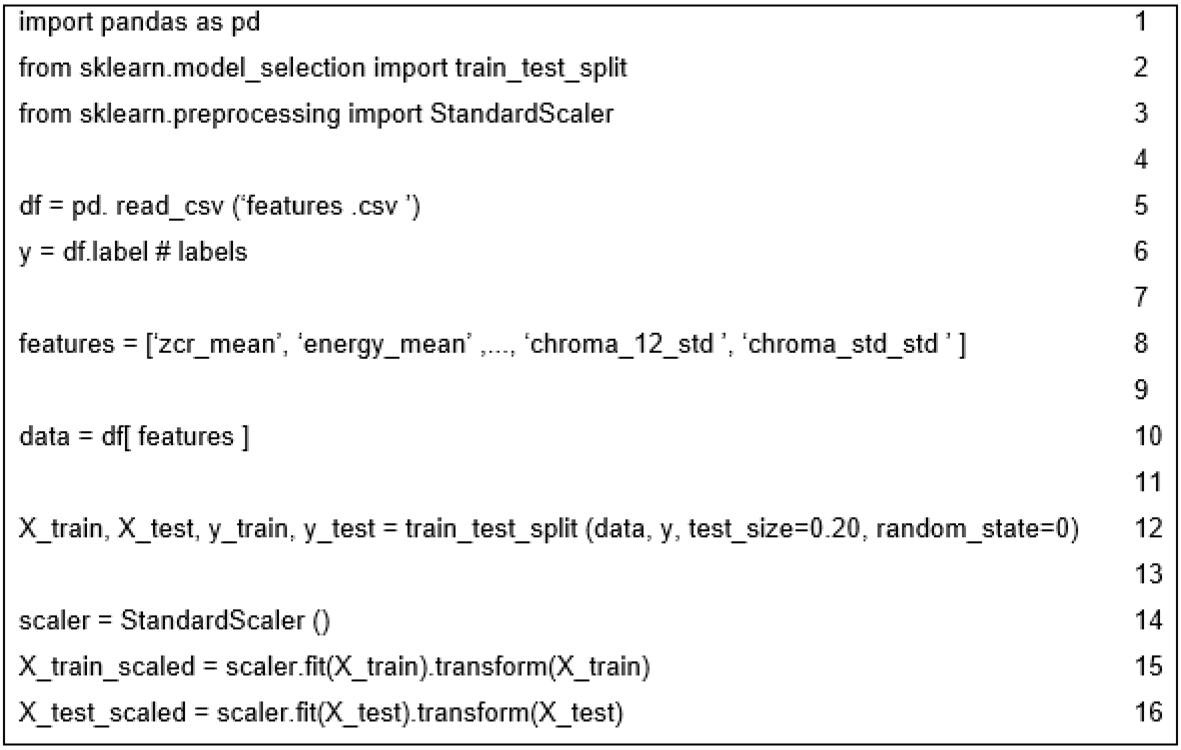
The problem of acquiring adequate speech data to train and test a speaker recognition system can be overcome through the use of a pre-recorded speech corpus. By using a popular or readily available speaker database, results can be compared directly to those published previously by other researchers. The readily available speaker corpora include the RSR2015, YOHO, TIMIT and ANDOSL corpora (Larcher *et al.*, 2014, Wildermoth & Paliwal, 2003), which have been used on well-resourced languages such as English. There is limited speech data available for most African indigenous languages. Pre-recorded voices of the Sepedi language native speakers were acquired from the National Centre for Human Language Technology (NCHLT) project (Barnard *et al.*, 2014, De Vries *et al.*, 2014). The dataset contains approximately 56 hours of recordings from 210 speakers. For this study, we selected a sample of 160 speakers with each speaker having 200 samples where each sample is a recording of a sentence that consists of 4-7 words. The selection criteria were that of speakers who have 200 samples or more. Table 3.1 shows the summary of the data used in this study, the data is divided into 80% train data and 20% test data sets. The sklearn's ***train_test_split()*** code is shown in line 12 of the Code Snippet 3.1 where X_train and X_test represent the train and test data and y_train and x_test represents the labels of the train and test data.

[3] https://www.qt.io/

[4] https://www.sqlite.org/index.html

Table 3.1 The NCHLT data used to train and test speaker recognition system

Unit	**Train Data**	**Test Data**
No. of speakers	160	160
No. of samples per speaker	160	40
Total Duration (minutes)	1422.35	35.49
Total Size (MB)	2867.20	716.80

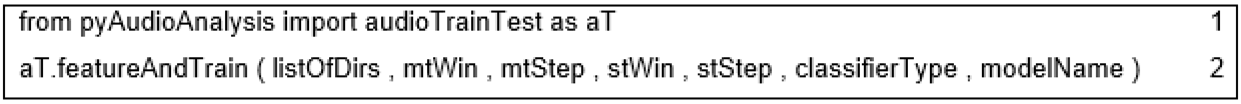


```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

df = pd. read_csv (‘features .csv ’)
y = df.label # labels

features = [‘zcr_mean’, ‘energy_mean’ ,..., ‘chroma_12_std ’, ‘chroma_std_std ’ ]

data = df[ features ]

X_train, X_test, y_train, y_test = train_test_split (data, y, test_size=0.20, random_state=0)

scaler = StandardScaler ()
X_train_scaled = scaler.fit(X_train).transform(X_train)
X_test_scaled = scaler.fit(X_test).transform(X_test)
```

Code Snippet 3.1 Data preparation code

3.5 Feature Extraction

One of the most crucial step in speaker recognition system is feature extraction. This step extracts acoustic features of speech from each speech signal. This study extracts a total number of 34 short-term acoustic features of speech (discussed in Section 2.4) using the pyAudioAnalysis library (Giannakopoulos, 2015). The 34 extracted acoustic features of speech that are extracted are divided into three domains, the time-domain, frequency-domain and cepstral-domain features. The time-domain features are features extracted directly from raw audio samples, the frequency-domain features are based on the magnitude of the Discrete Fourier Transform, and the cepstral-domain features (Mel-Frequency Cepstral Coefficients or MFCCs) results after the Inverse Discrete Fourier Transform is applied on the logarithmic spectrum. Feature extraction is performed with the $featureAndTrain()$ function imported from pyAudioAnalysis.



```
from pyAudioAnalysis import audioTrainTest as aT                                                    1
aT.featureAndTrain ( listOfDirs , mtWin , mtStep , stWin , stStep , classifierType , modelName )    2
```

Code Snippet 3.2 Features extraction function header in pyAudioAnalysis

The function *featureAndTrain(listOfDirs, mtWin, mtStep, stWin, stStep, classifierType, modelName)* (line 2, Code Snippet 3.2) from the audioTrainTest.py script in pyAudioAnalysis is used as a wrapper to segment-based audio feature extraction and classifier training. The function takes the following arguments:

- *listOfDirs*: This is a list of directories in which samples are stored within each equivalent class. This points to the location of the raw data.
- *mtWin*, *mtStep*: Represents the mid-term window size and step.
- *stWin* and *stStep*: Represents short-term window size and step.
- *classifierType*: four different classifiers (SVM, KNN, gradient boosting, and random forest) are implemented in pyAudioAnalysis. Therefore, this argument represents any one of the classifiers. However, for this study we ignore the defined classifiers.
- *modelName*: name of the model to be saved.

The output of the feature extraction step is a CSV file that contains a list of all the attributes and instances. This file is then read into the working environment (line 5, Code Snippet 3.1) and train test split is performed followed by feature normalisation (see Section 3.6).

3.6 Feature Normalisation

The aim of feature normalisation is to reduce speaker and recording variability and is an essential factor in a robust speaker recognition system. This study adopted the mean variance normalisation method where features are normalised so that they are centred around 0 with a standard deviation of 1 (Pyrtuh *et al.*, 2013, Mazibuko & Mashao, 2007). The normalised feature y_i is calculated with the following equation:

$$y_i = \frac{x_i - \mu}{\sigma}$$

where μ and σ represent mean and the variance for each feature x_i. This step is performed in line 14-16 of the Code Snippet 3.1.

3.7 Classifier Model Setup

After feature extraction is performed with feature normalisation, and the data is split into train and test partitions, *sklearn* is launched for training the speaker classifier models (speaker modelling). This section discusses parameters of the ML algorithms discussed in Section 2.5. This study used the *GridSearchCV*[5] algorithm implemented on sklearn to search for the best hyper-parameters for each algorithm. The Code Snippet 3.3 shows the training of the classifier models. Lines 19-23 shows the declaration/definition of the classifier model with the hyper-parameters obtained from the *GridSearchCV* results.

The function, $model\_train(model)$, line 8, is defined to receive one argument, which represents the speaker classifier model name. The function fits ($model\,.fit(\,X\_train\,,y\_train\,)$) the training data (line 9) to the model for training and then makes predictions with the test data (line 10). Then the evaluation metrics (discussed in Section 3.8) are calculated using the $\boldsymbol{sklearn.metrics}$ libraries, the results are then printed on to the screen (lines 12-16). The function is called in a $\boldsymbol{for\ loop}$ on line 28. For each iteration, the model name is passed on to the function.

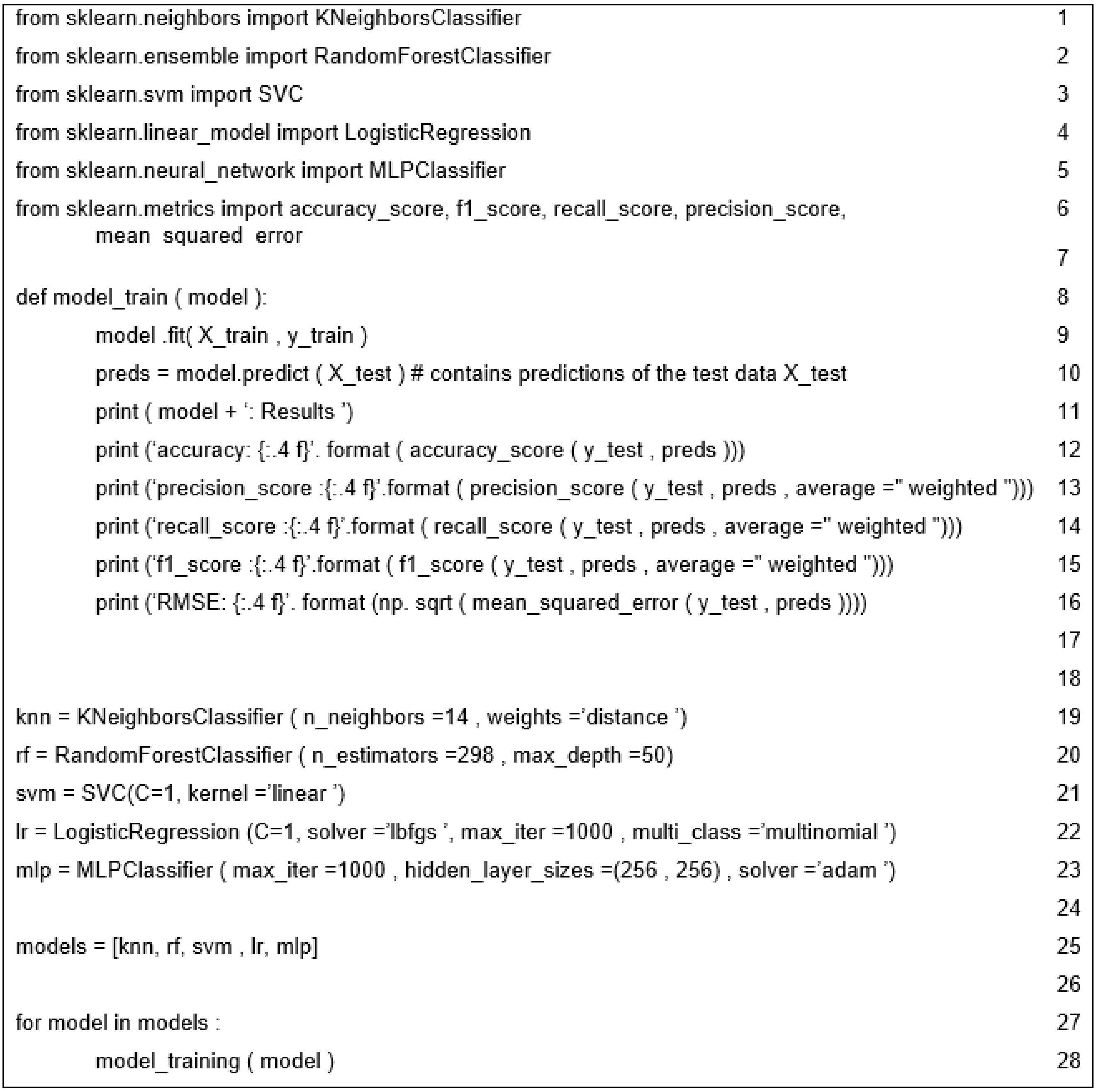
3.7.1 K-Nearest Neighbours (KNN)

The parameter 'k' representing the number of nearest neighbours and the parameter $weight$ which representing the prediction weight function are used to train the KNN classifier model. The available options for the parameter $weight$ are:

- **uniform**: every point in each neighbourhood is equally weighted.
- **distance**: every point in each neighbourhood is weighted according to the inverse of its distance, i.e., neighbours closer to a query point have a more influence than those that are further away.

[5] https://scikit-learn.org/stable/modules/grid_search.html

The GridSearchCV algorithm, depicted in Figure 3.2, suggests that given the acquired for this study, KNN classifier can be trained with the parameters $k = 14$ and $weight = distance$ to get better performances, meaning that closer neighbours have a greater influence compared to neighbours which are further away. The KNN classifier model, defined in line 19 of Code Snippet 3.3, is therefore trained with the parameters $k = 14$ and the $weight$ parameter equals to $distance$.



```
from sklearn.neighbors import KNeighborsClassifier                                              1
from sklearn.ensemble import RandomForestClassifier                                             2
from sklearn.svm import SVC                                                                     3
from sklearn.linear_model import LogisticRegression                                             4
from sklearn.neural_network import MLPClassifier                                                5
from sklearn.metrics import accuracy_score, f1_score, recall_score, precision_score,            6
        mean  squared  error
                                                                                                7
def model_train ( model ):                                                                      8
        model .fit( X_train , y_train )                                                         9
        preds = model.predict ( X_test ) # contains predictions of the test data X_test         10
        print ( model + ‘: Results ’)                                                           11
        print (‘accuracy: {:.4 f}’. format ( accuracy_score ( y_test , preds )))                12
        print (‘precision_score :{:.4 f}’.format ( precision_score ( y_test , preds , average =" weighted ")))   13
        print (‘recall_score :{:.4 f}’.format ( recall_score ( y_test , preds , average =" weighted ")))      14
        print (‘f1_score :{:.4 f}’.format ( f1_score ( y_test , preds , average =" weighted ")))            15
        print (‘RMSE: {:.4 f}’. format (np. sqrt ( mean_squared_error ( y_test , preds ))))     16
                                                                                                17
                                                                                                18
knn = KNeighborsClassifier ( n_neighbors =14 , weights =’distance ’)                            19
rf = RandomForestClassifier ( n_estimators =298 , max_depth =50)                                20
svm = SVC(C=1, kernel =’linear ’)                                                               21
lr = LogisticRegression (C=1, solver =’lbfgs ’, max_iter =1000 , multi_class =’multinomial ’)     22
mlp = MLPClassifier ( max_iter =1000 , hidden_layer_sizes =(256 , 256) , solver =’adam ’)        23
                                                                                                24
models = [knn, rf, svm , lr, mlp]                                                               25
                                                                                                26
for model in models :                                                                           27
        model_training ( model )                                                                28
```

Code Snippet 3.3 Classifier Model training on Scikit-learn

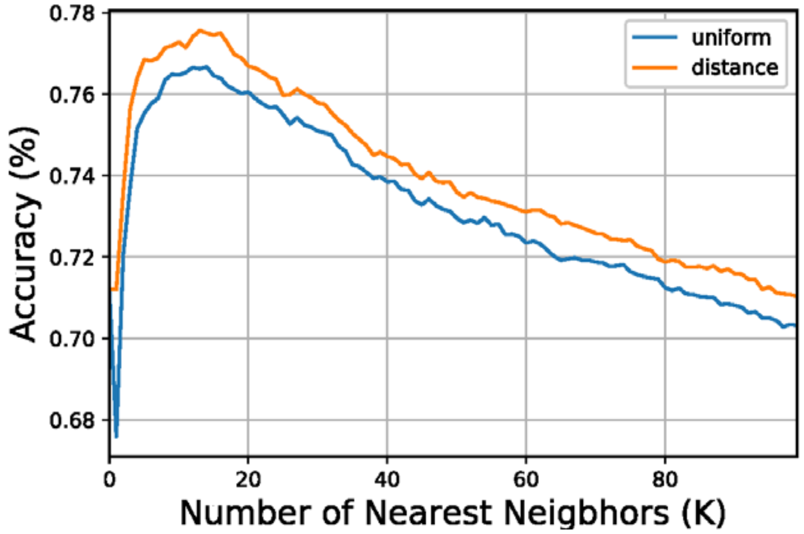


Figure 3.2 KNN Optimisation with parameters: k and weight.

3.7.2 Random Forest (RF)

The parameters that are used to train the RF classifier are the parameter max_depth representing the maximum depth of the tree and the parameter $n\_estimators$ representing the number of trees in the forest. If the max_depth parameter is not given, the nodes of each tree will be expanded until all the leaves contain less than the minimum number of samples required to split an internal node, that is, until all leaves are pure. The GridSearchCV algorithm predictions, shown in Figure 3.3, suggests that the best parameters for the RF classifier are achieved with $n\_estimators = 298$ and $max\_depth = 50$ as the best parameters. Line 20 of Code Snippet 3.3 defines the RF classifier model.

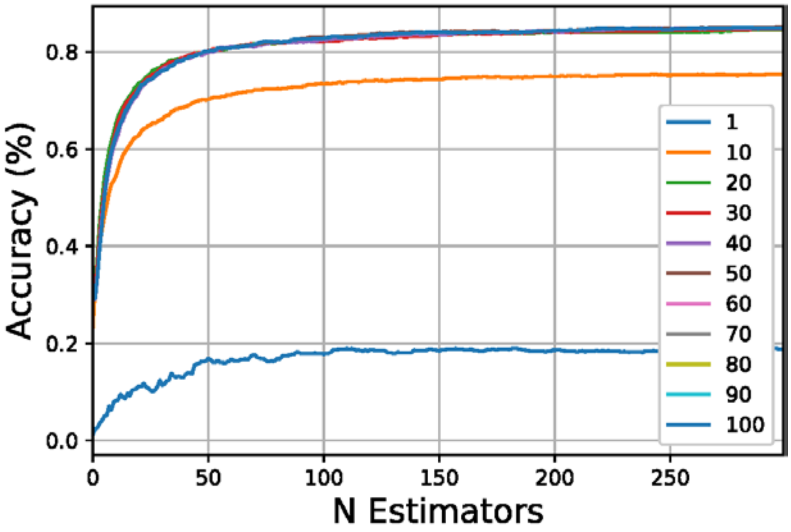


Figure 3.3 RF optimisation with parameters: number of trees and tree depth.

3.7.3 Support Vector Machine (SVM)

The SVM classifier model is trained with the penalty parameter C of the error term and different kernels are evaluated. The kernels available are the *Linear, Polynomial, Radial Basis Function (rbf),* and *Sigmoid* kernels which are defined by the following equations:

$$Linear = \langle x, x' \rangle$$

$$Polynomial = (\gamma\langle x, x' \rangle + r)^d$$

$$RBF = \exp\,(-\gamma\|x - x^1\|^2)$$

$$Sigmoid = \tanh\,(\gamma\langle x, x' \rangle + r)$$

where γ is represents a positive parameter, d represents the kernel degree, and r represents the coefficient. The SVM classifier model has the C parameter which is the penalty for misclassifying a data point. If is given as a small integer, the SVM classifier will not be entirely penalised for misclassifying data points (high bias, low variance). However, if the parameter C is given as a large integer, the SVM classifier will be penalised heavily for misclassifying data points (low bias, high variance).

Optimisation of the SVM algorithm is shown in Figure 3.4 showing the performance of SVM kernels when trained with different parameter C values. It is shown that the best hyper-parameters are the linear kernel and the penalty parameter $C = 1$. The SVM classifier model is defined in line 21 of the Code Snippet 3.3.

.

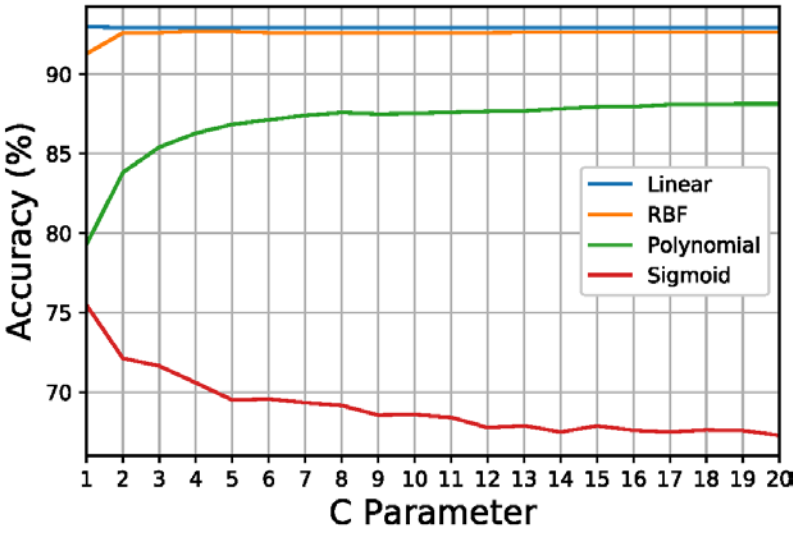


Figure 3.4 SVM Optimisation with parameters: C and kernels.

3.7.4 Logistic Regression (LR)

The LR classifier is trained for 1000 iterations with the penalty parameter C and the solver parameters. The parameter C is the inverse of regularisation strength and like in SVMs, smaller values specify stronger regularisation. The parameter solver refers to the algorithm to use in the optimisation problem. Possible values for optimisation algorithms are *newton-cg, lbfgs, liblinear, sag* and *saga*. As shown in Figure 3.5, $C = 1$ and $solver = ‘lbfgs’$ are reported as the best hyper-parameters (obtained from the GridSearchCV). The LR classifier model is defined in line 22 of the Code Snippet 3.3.

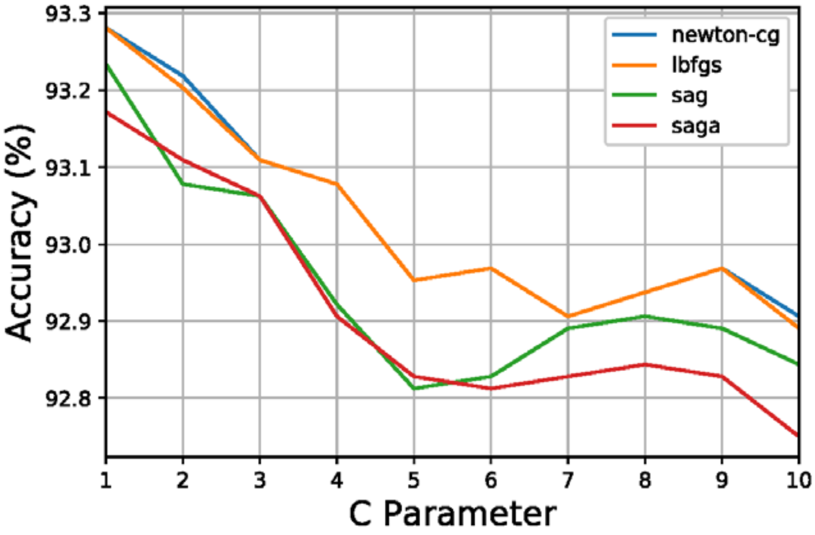


Figure 3.5 LR optimisation with parameters: C and solver.

3.7.5 Multi-layer Perceptron (MLP)

A maximum of 1000 iterations are performed to train the MLP classifier of two hidden layers where each hidden layer contains 256 neurons. The *rectified linear unit (relu)* activation function is used to activate each layer and the *softmax* function is used to activate output layer. The *softmax* function is used because of the categorical data and *softmax* takes a real number vector as an input and normalises it into a probability distribution that is composed of probabilities. The following equation defines the standard *softmax* function $\sigma : \mathbb{R}^K \to \mathbb{R}^K$:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e_{\alpha}^{z_j}}$$

where $i = 1, \ldots, K, z = (z_1, \ldots, z_K) \in \mathbb{R}^K$, and z_i is an element of the input vector z. The *adam* optimisation algorithm (Kingma & Ba, 2014) and *Stochastic Gradient Descent (SGD)* are used to compile the MLP and the learning curves are shown in Figure 3.6 (obtained from the GridSearchCV). The learning curves show that *SGD* is slow to converge and *adam* converges quicker after about 100 epochs. Therefore, *adam* optimisation algorithm was selected as the best parameter since it converges faster than the SGD optimisation algorithm. Line 23 of the Code Snippet 3.3 defines the MLP classifier model.

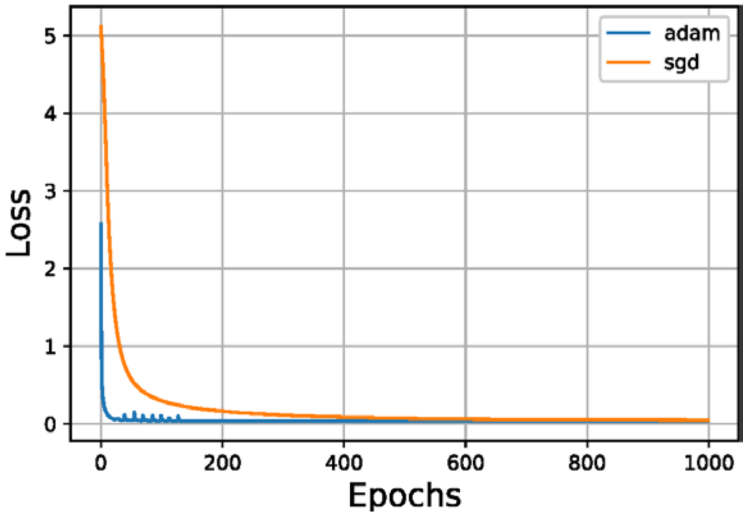


Figure 3.6 Loss function for MLP optimisation with adam and sgd algorithms

3.8 Evaluation

The performance of each classifier model is influenced by the quality of the audio files, the size of training data, and, above all, the machine learning algorithm used. The evaluation metrics used to evaluate the performance of the trained classifier models in this study are as follows:

- **Accuracy:** the percentage of samples that have been properly classified from all the samples given. It is calculated by the following equation:

$$Accuracy = \frac{t_p + t_n}{t_p + t_n + f_p + f_n}$$

- **Precision:** the percentage of samples that, among all those listed as class x, really belong to class x. It is calculated by the following equation:

$$Precision = \frac{t_p}{t_p + f_p}$$

- **Recall:** the percentage of samples that, among all samples that really have class x, were classified as class x. It is calculated by the following equation:

$$Recall = \frac{t_p}{t_p + f_n}$$

- $\boldsymbol{F_1\, score}$**:** the harmonic mean of precision and recall. It is calculated by the following equation:

$$F_1 = 2 \times \frac{precision \ \times recall}{precision + recall}$$

- **The root mean squared error (RMSE):** a quadratic scoring metric that calculates the average error magnitude. It is calculated by the following equation:

$$RMSE= \sqrt{\frac{1}{\mathrm{n}}\sum_{\mathrm{i=1}}^{\mathrm{n}}(y_i - \hat{y}_i)^2}$$

Where,

- t_p = number of positive samples that are predicted positive (true positives).
- f_p = number of negative samples that are predicted positive (false positives)
- t_n = number of negative samples that are predicted negative (true negatives).
- f_n = number of positive samples that are predicted negative (false negatives).

3.9 Summary

This chapter discussed the training and testing stages taken in this study. The stages covered data acquisition, feature extraction, feature normalisation, training the classifier models including parameter optimisation and model evaluation. The data of pre-recorded voices was acquired from the NCHLT project. The pyAudioAnalysis package was used to extract acoustic features of speech which are imported into the machine learning framework, Scikit-Learn, that trains the classifier models. The GridSearchCV algorithm is used to perform parameter optimisation. The evaluation metrics such accuracy, precision, recall, $F_1\, score$ and the RMSE are calculated to evaluate classifier models and the results are discussed in Chapter 5. The next chapter discusses the implementation of the graphical user interface (GUI).

CHAPTER 4: SYSTEM IMPLEMENTATION

4.1 Introduction

The tools and packages discussed in Chapter 3, Section 3.3 are required to train and developed the proposed speaker recognition system. This chapter discusses the proposed automatic speaker recognition system implementation covering system design (Section 4.2) which explains the database design and the graphical user interface development. The chapter also discusses the evaluation process of the developed graphical user interface by measuring performance and usability (Section 4.4.)

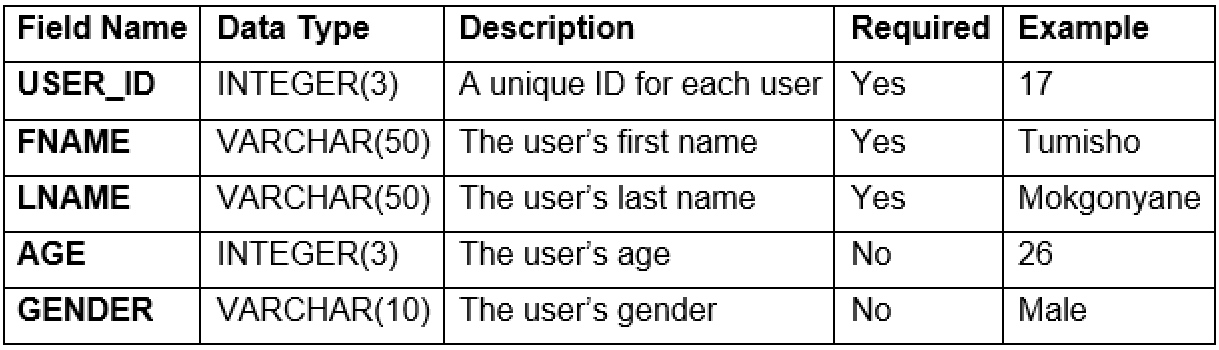
4.2 System Design

This section details the database design and development of the graphical user interface (GUI).

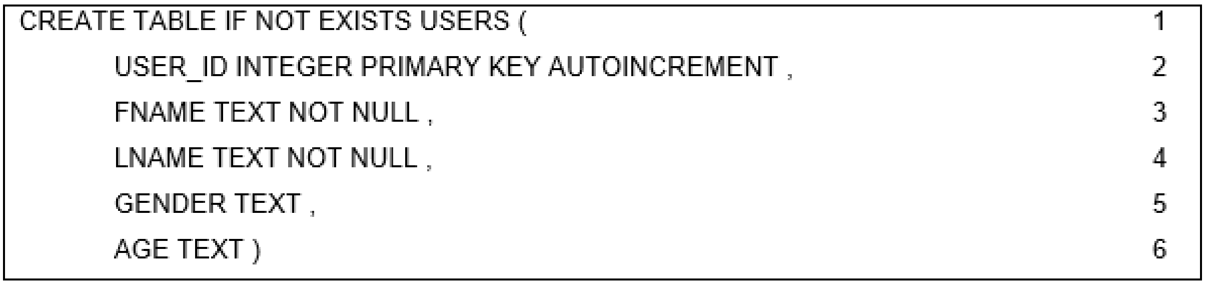
4.2.1 Database Design

The user information is stored in a SQLite3 database developed in Python. The database is designed with only one table (**USERS TABLE**), containing four attributes *(user_id, fname, lname, age, and gender)* depicted in the data dictionary show in Table 4.1. The command used to **CREATE** the database table is shown in Code Snippet 4.1 and the database queries used to **INSERT**, **UPDATE** and **RETRIEVE** the user's data stored in the database table Code are shown Snippet 4.2 to Code Snippet 4.4. The Question Mark (?) in the Code Listings represent placeholders or variables

Table 4.1 Data Dictionary

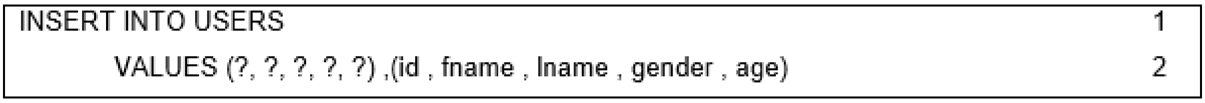


Field Name	Data Type	Description	Required	Example
USER_ID	INTEGER(3)	A unique ID for each user	Yes	17
FNAME	VARCHAR(50)	The user's first name	Yes	Tumisho
LNAME	VARCHAR(50)	The user's last name	Yes	Mokgonyane
AGE	INTEGER(3)	The user's age	No	26
GENDER	VARCHAR(10)	The user's gender	No	Male



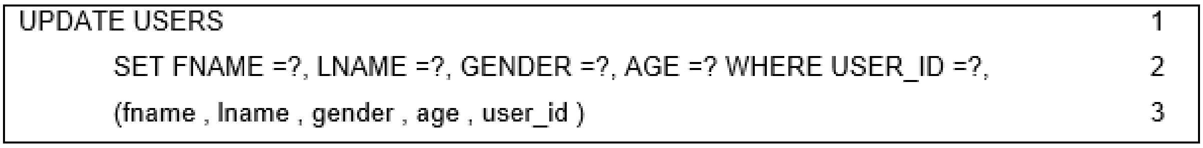
```
CREATE TABLE IF NOT EXISTS USERS (
          USER_ID INTEGER PRIMARY KEY AUTOINCREMENT ,
          FNAME TEXT NOT NULL ,
          LNAME TEXT NOT NULL ,
          GENDER TEXT ,
          AGE TEXT )
```

Code Snippet 4.1 Creating a USERS database table.



```
INSERT INTO USERS
          VALUES (?, ?, ?, ?, ?) ,(id , fname , lname , gender , age)
```

Code Snippet 4.2 Inserting data (new users) in the database table.



```
UPDATE USERS
          SET FNAME =?, LNAME =?, GENDER =?, AGE =? WHERE USER_ID =?,
          (fname , lname , gender , age , user_id )
```

Code Snippet 4.3 Updating data (existing users) in the database table.

```
SELECT * FROM USERS
          WHERE USER_ID =?, (str( user_id )
```

Code Snippet 4.4 Retrieving data from the database table.

4.2.2 The Graphical User Interface

The graphical user interface (GUI) is developed to offer easy access to the speaker recognition system and to perform speaker recognition functionalities in real-time. Figure 4.1 shows the GUI developed with *QT Creator* and *PyQT4*. The GUI contains three tabs, namely the **ENROLMENT**, **IDENTIFICATION** and **VERIFICATION** tab. The GUI runs *Python3* in the back-end.

The first tab is the *enrolment* tab which comes up as the first interface when the system is launched. This tab is for the training (enrolment) phase where the users register their biographical data (first name, last name, age and gender) and either records or upload a recording of their voice. Then clicks on the **Train** which will train a model and enrol the user in the speaker database.

The second tab is the *identification* tab which matches an unknown voice (recorded or inputted speech sample) to one of the enrolment speakers. The name, age and gender of the matched user are returned as results, accompanied by the probability of the match which has to be equal to or higher than the set threshold. If the probability of the match is less that the set threshold, the unknown voice is classified as belonging to an unenrolled speaker.

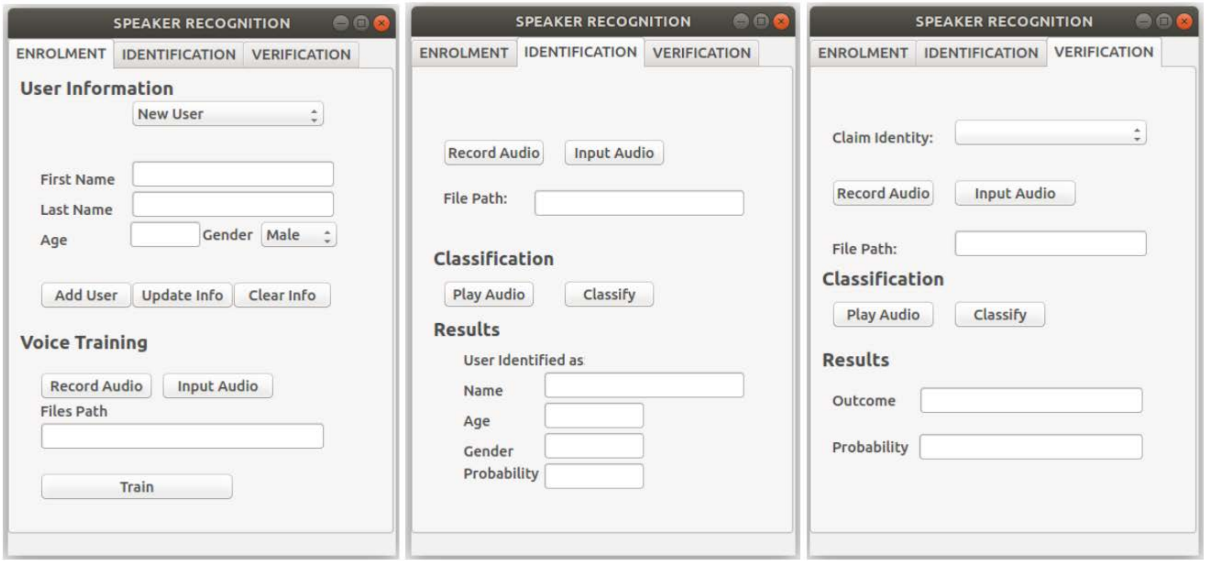


Figure 4.1 Speaker recognition Graphical User Interface.

The third tab is the *verification* tab which is used to verify whether an unknown voice (recorded or inputted speech sample) belongs to a certain enrolled speaker. An identity claim is performed and the results returned are the outcome (ACCEPT or REJECT) and the probability of match, which is also compared against the threshold. The claimed identity is rejected if the probability of the match is less than the set threshold.

4.2.3 Components of the GUI

The developed GUI consists of the following components:

4.2.3.1 Text Input

This component allows the user to input text into the system. The Enrolment tab has three text input components, for *first name, last name* and *age.*

4.2.3.2 Text Output

This component gives output to the user in the form of text. The Enrolment tab has one text output component (File Path) which shows the location of the recorded or inputted audio file.

The Identification tab has the following text output components:

- *File Path*: shows the location of the recorded or inputted audio file.

- *Name, Age,* and *Gender*: shows details of the identified speaker.
- *Probability*: shows the probability of the match for the identified speaker.

The Verification tab has the following text output components:

- *File Path*: shows the location of the recorded or inputted audio file.
- *Outcome*: results of the claimed identity (ACCEPT or REJECT) are shown here
- *Probability*: shows the probability of the match for the *identified* speaker.

4.2.3.3 OnClick Buttons

This component performs a defined action/function when clicked. The Enrolment tab has the following buttons:

- *Add User*: When clicked, this button reads details entered in the Text Input components and registers the new user into the speaker database. The details include first name, last name, age and gender of the new user to be enrolled in the system.
- *Update Info*: When clicked, this button updates the details of an already registered user in the speaker database.
- *Clear Info*: When clicked, this button clears the current information entered in the input text components and resets the drop down lists. See **Section 4.3.3.4** for dropdown list components.
- *Record Audio*: This button is used to record new audio files. As soon as the user finishes recording, the system automatically stores the recorded file to the computer and then output the path or location of this file.
- *Input Audio*: This button allows the user select an already recorded audio file from the computer and outputs the path or location of the file.
- *Train*: When clicked, this button executes the enrolment process explained in Section 4.2.

The Identification and Verification tabs have the following buttons:

- *Record Audio and Input Audio*: The buttons perform similar actions as those in the Enrolment tab.
- *Play Audio*: When clicked, this button plays the selected or recorded audio file shown in the *File Path* text output component.

- *Classify*: When clicked, this button executes the identification or verification process explained in Section 4.2. With identification, the system returns details of the identified speaker and with verification, the system verifies if the returned speaker identity matches the claimed identity. If there’s a match the user is accepted, else rejected.

4.2.3.4 Dropdown

The dropdown component allows a user to select one option from a list. The Enrolment tab has one dropdown list containing user names of enrol speakers and a *New User* option which is selected when a new user enrols in the system. The Verification tab has one dropdown list also containing a list of enrolled speakers. An identity claim is performed by selecting a user name from this dropdown list.

4.3 Evaluating the GUI

The GUI’s for performance and usability is evaluated in real-time and the results are reported in Chapter 5, Section 5.3. To determine performance, the system first determines the probability of the match for the test speaker (utterance) and then compares the probability with a predefined threshold.

The developed speaker recognition system’s performance is determined by how accurate the identified speakers reflect the actual speakers. The following evaluation metrics are calculated to measure the system’s performance:

- **True Acceptance Rate (TAR):** the rate at which the speaker recognition system accepts a valid identity claim.
- **True Rejection Rate (TRR)**: the rate at which the speaker recognition rejects a false identity claim.
- **False Acceptance Rate (FAR)**: the rate at which the speaker recognition accepts an invalid identity claim.
- **False Rejection Rate (FRR)**: the rate at which the speaker recognition rejects a false identity claim.

Table 4.2 shows the design of a confusion matrix designed for evaluation the performance of the developed speaker recognition system.

Table 4.2 Confusion matrix for evaluating a Speaker Identification System

Speaker Recognition System		**Actual Speakers**	
		Registered	**Unregistered**
Identified Speakers	**Registered**	TAR	FRR
	Unregistered	FAR	TRR

The system's *usability* is determined by the recruited speakers (respondents) with the use of an evaluation form. The evaluation form (Appendix B) includes eight (8) close-ended questions where respondents are requested to rate the system's *usability* on a 5-point Likert scale, and an optional open-ended question where respondents are requested to give reasons for the ratings given. The eight (8) close-ended questions are as follows:

- The menu items are well arranged and functions are easy to find.
- The functions of each menu item are easily understandable.
- All the functions I expected to find in the menus are present.
- The help of a technical person is needed for me to be able to use the system.
- The system was built with a simple, clean, uncluttered screen.
- When completing a task, the system keeps screen changes to a minimum.
- The system responds quickly and reduces the number of steps needed to complete tasks.
- The system's overall impression.

The mean response is calculated to determine the Mean-Opinion-Score (MOS), which is a numerical measure of the overall quality of an occurrence or experience judged by humans. The following equation is used to calculate the MOS:

$$MOS = \frac{1}{n}\Sigma_{i=1}^{n}x_i$$

where x_i is the score assigned by respondent i and *n* is the total number of subjects.

4.4 Summary

This chapter discussed the system implementation covering requirement analysis, system design and evaluation. The database and system designed are discussed in Section 4.3. SQLite3 database is used as the main database to store speaker data. QT Creator and PyQt4 are used to develop the GUI. The system runs Python 3 in the back-end. Section 4.4 discussed the evaluation process of the developed GUI by measuring performance and usability. The following chapter discusses the experimental and evaluation results.

CHAPTER 5: RESULTS AND DISCUSSION

5.1 Introduction

The results from the experiments carried out are presented and discussed in this chapter. The performance of the classifier models (KNN, RF, SVM, LR and MLP) is reported in Section 5.2. The performance of the graphical user interface (GUI) mentioned in Chapter 4 is discussed in Section 5. 3. Section 5.4 discusses the GUI's usability as determined by the respondents who participated in the evaluation and testing of the developed speaker recognition system. The GUI evaluation process is discussed in Section 4.4 of Chapter 4. Section 5.5 discusses overall results and findings and Section 5.6 summarises and concludes the chapter.

5.2 Results on Model Performance

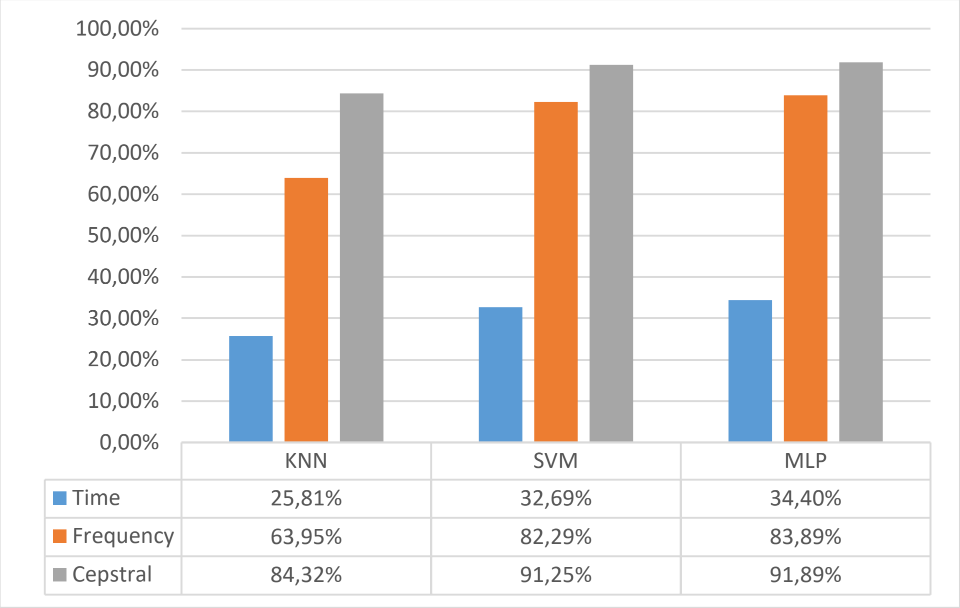
This section describes the experiments conducted and the results obtained. Two different experiments are conducted where the second experiment takes input or decisions from the first experiment. In the first experiment, the effect of acoustic features of speech is investigated and in the second experiment, different machine learning algorithms (classifier models) are compared.

5.2.1 Experiment 1: Effects of Acoustic Features of Speech

The purpose of this experiment was to determine which of the features give better performance on a given dataset. A total of 34 features were extracted as discussed in Chapter 3. The features include Time-domain, Frequency-domain and Cepstral-domain features. The features are individually trained and their performances is compared, and then the features are combined to investigate their compatibility. The following combinations are investigated:

- TF = Time + Frequency domain features
- TC = Time + Cepstral domain features
- FC = Frequency + Cepstral domain features
- TFC = Time + Frequency + Cepstral domain features

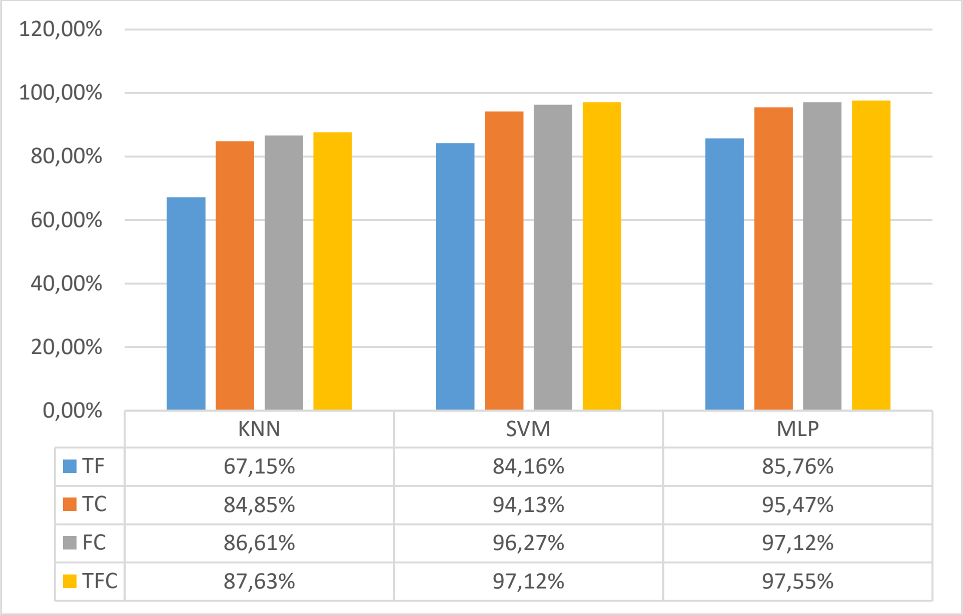
The accuracy results obtained from training three classifier models (KNN, SVM, and MLP) with different the three individual acoustic speech characteristics are shown in Figure 5.1. Time-domain features are found to offer the lowest accuracy for all classifier models (25%–34%). It is observed that the performance improves for all classifiers respectively when frequency-domain features are used (63%-83%). Cepstral-domain features improve the accuracy even further to 84.32% for KNN, 91.25% for SVM and 91.89% for MLP. From these results, we see that the performance is affected by different acoustic features of speech and that Cepstral-domain features give better performances.



	KNN	SVM	MLP
🟦 Time	25,81%	32,69%	34,40%
🟧 Frequency	63,95%	82,29%	83,89%
■ Cepstral	84,32%	91,25%	91,89%

Figure 5.1 Accuracy scores obtained from three acoustic features of speech

We combined Time-domain with Frequency-domain features (TF) to investigate the compatibility of the features. As depicted in Figure 5.2, the performance has increased by 3.20% for KNN and improved by 1.87% for both SVM and MLP classifier models, resulting in an average improvement of 2.31%. The performance improved by improved by 2.33% average when combining Time-domain features with Cepstral-domain features. Lastly, combining Frequency-domain features with Cepstral-domain features has improved the performance by higher average of 4.81%. From this results, we conclude Frequency-domain features and Cepstral-domain features (FC) are more compatible with each other as compared to Time-domain features combined with Frequency-domain (TF) features, and Time-domain features combined with Cepstral-domain (TC) features.



	KNN	SVM	MLP
🟦 TF	67,15%	84,16%	85,76%
🟧 TC	84,85%	94,13%	95,47%
⬜ FC	86,61%	96,27%	97,12%
🟨 TFC	87,63%	97,12%	97,55%

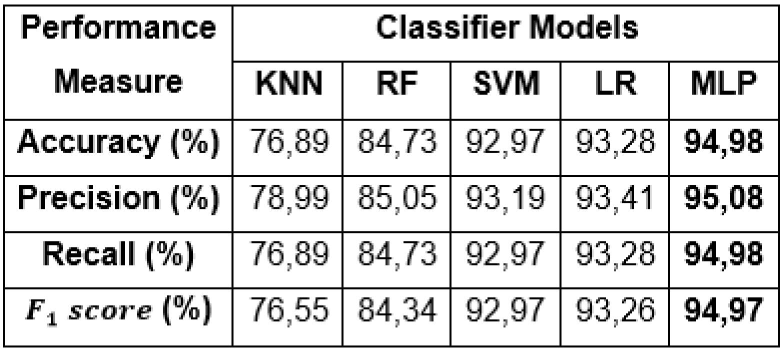
Figure 5.2 Accuracy scores obtained from combining features.

Figure 5.2 shows that when all three acoustic speech features (TFC) are combined, the performance improves even further. The MLP classifier model performs best with the highest accuracy of 97.55% and KNN achieves the lowest accuracy of 87.63%, followed by SVM with an accuracy of 97.12%. MLP outperforms SVM by a difference of only 0.43% and outperforms both KNN and SVM classifier models regardless of the features in use.

5.2.2 Experiment 2: Performance of Classifier Models

In this experiment, ML algorithms (classifier models) are compared. The intent of this experiment was to determine which classifier model gives the best performances on the given dataset. Since we observed that we get a better performance by combining the acoustic features of speech (Experiment 1), this experiment uses a combination of the features to train classifier models. Table 5.1 reports the classifier model's performance results on *accuracy, precision, recall,* and $F_1\ score$. The results show that the MLP classifier outperforms all the classifiers by achieving the highest accuracy of 94.98% whereas KNN has the lowest accuracy of 76.89% followed by RF with an accuracy of 84.73%. As reported in the literature that LR performs better than

Table 5.1 Performance of the classifier models.



Performance Measure	Classifier Models				
	KNN	RF	SVM	LR	MLP
Accuracy (%)	76,89	84,73	92,97	93,28	**94,98**
Precision (%)	78,99	85,05	93,19	93,41	**95,08**
Recall (%)	76,89	84,73	92,97	93,28	**94,98**
F_1 *score* (%)	76,55	84,34	92,97	93,26	**94,97**

SVM (Katz *et al.*, 2006), it is observed in Table 5.1 that LR outperforms SVM by a difference of 0.31% in this study.

Looking at precision, it is observed that it is slightly higher than both accuracy and recall for all the five classifier models. With precision, we calculate the percentage of speakers that, among all those listed as class x, really belong to class x and we observe the precision of 78.99% for KNN, 85.05% for RF, 93.19% for SVM, 93.41% for LR and 95.08% for MLP. This means that the majority of the correctly recognised speakers are truly the recognised the speakers.

With *Recall* we calculate the percentage of samples that, among all samples that really have class x, were classified as class x. It is observed that recall is similar to accuracy results for all the five classifier models, meaning that the speakers are correctly recognised without overfitting the models.

There are several studies in the literature reporting that accuracy may be misleading when there is high difference between recall and precision (Brownlee, 2014). As such, $F_1\ score$ is a viable solution as it finds the harmonic means of both the recall and precision. Therefore, we have calculated the $F_1\ score$ of the classifier models and achieved $F_1\ scores$ of 76.55%, 84.34%, 92.97%, 93.26% and 94.97% for KNN, RF, SVM, LR and MLP respectively. The results for the $F_1\ score$ are almost similar to the observed accuracy for all the classifier models and thus we conclude that accuracy is enough to evaluate the classifier models and that the classifier models are not overfitted.

The standard deviation of the prediction (recognition) errors, the RMSE, is depicted in Figure 5.3. With the highest RMSE of 42.70, KNN misclassifies most of the data followed by RF with a RMSE of 34.61. The SVM and LR classifiers performed slightly better with RMSE of 22.88 and 21.04 respectively and a difference of only 1.84. MLP had the lowest RMSE of 17.76 which suggests that MLP gives better predictions with lower classification errors.

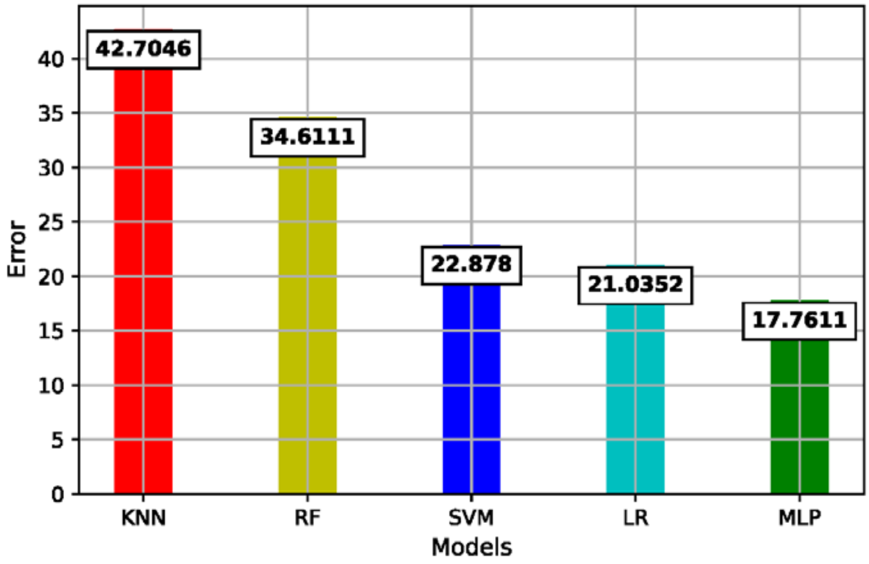


Figure 5.3 The RMSE of the trained classifier models

In terms of the results discussed above, it is seen that the MLP classifier model gives best performances and therefore it is selected as the best classifier model and is deployed to further the research project.

5.3 Results on GUI Performance

The best performing model (MLP) is implemented in the GUI for real-time speaker recognition. The GUI is evaluated as discussed in Section 4.4 and the results are reported in Figure 5.2-5.3. and in Table 5.2. Fifteen (15) Sepedi language native speakers were recruited to help evaluate the performances and usability of the developed speaker recognition system. Twelve (12) of the 15 participants were enrolled into the system to test whether the system would correctly identify and verify them as enrolled users. The three (3) remaining participants did not enrol in the system to test whether the system would reject them as invalid or unregistered users.

As shown in Table 5.2, the system was able to correctly identify 10 of 12 registered speakers and misidentify 2 speakers, leading to a **TAR** of 66.67% and a **FRR** of 13.33%. The results in Table 5.2 also show that 1 of the 3 unregistered users was identified as registered and 2 of 3 unregistered users were correctly identified as unregistered users, therefore obtaining a **TRR** of 13.33% and **FAR** of 6.67%.

The performance of the best performing model (MLP) is also evaluated based on the standard performance metrics discussed in Chapter 3 and the results are reported in Table 5.3. We model performs well with an 80% accuracy and a higher precision of 83.33% and an even higher recall percentage of 90.91%. It is reported in the literature that accuracy can be misleading when there is high difference between recall and precision (Brownlee, 2014) and we observe here that recall is higher than precision with a difference of 7.58% meaning that indeed is misleading in terms of evaluating the deployed model. We however calculated the F_1 score which is the harmonic mean between precision and recall and it is observed to be 86.96% and therefore we conclude that our model performs best at 86.96% accuracy.

Table 5.2 The GUI performance.

Speaker Recognition System		**Actual Speakers**	
		Registered	**Unregistered**
Identified Speakers	**Registered**	10	2
	Unregistered	1	2

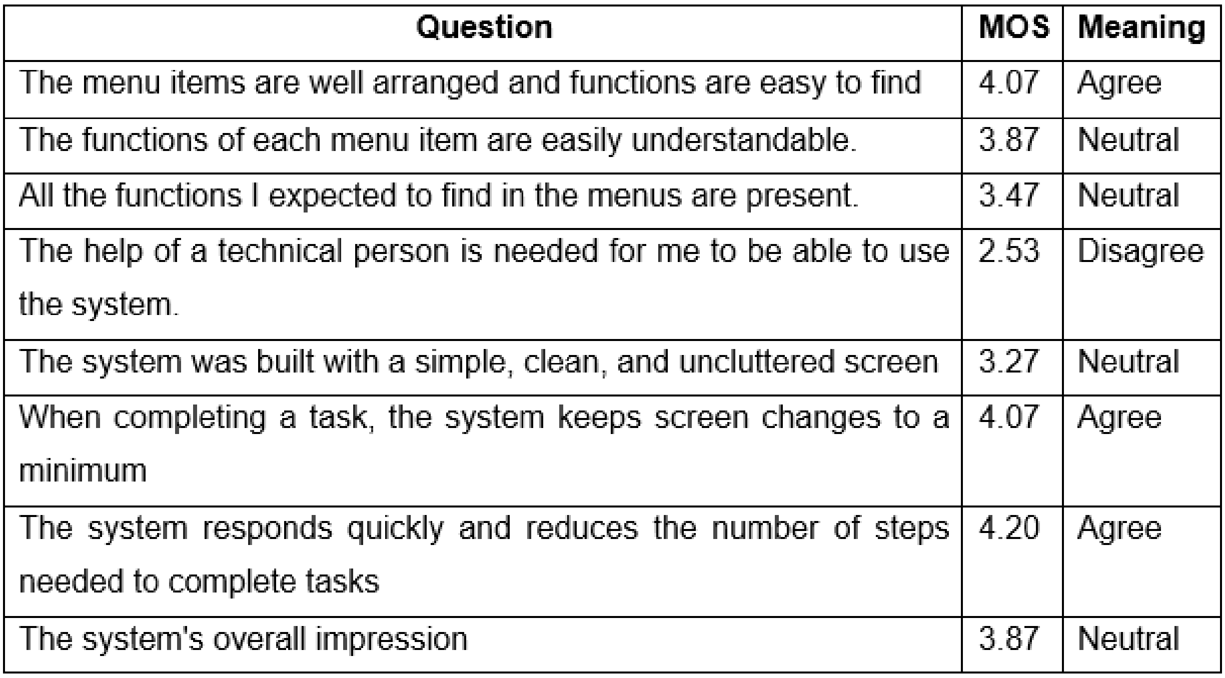
Table 5.3 The performance deployed model on GUI.

Registered	**Unregistered**
Accuracy	80%
Precision	83.33%
Recall	90.91%
F_1 score	86.96%

5.4 Results on GUI Usability

This section discusses the feedback from the evaluation forms regarding functional requirements and usability testing. The questions asked in the evaluation form are given in Appendix A and the responses are reported in Table 5.4 (the MOS) and in Figure 5.4 to Figure 5.11.

Table 5.4 Mean Opinion Scores calculated from the responses



Question	MOS	Meaning
The menu items are well arranged and functions are easy to find	4.07	Agree
The functions of each menu item are easily understandable.	3.87	Neutral
All the functions I expected to find in the menus are present.	3.47	Neutral
The help of a technical person is needed for me to be able to use the system.	2.53	Disagree
The system was built with a simple, clean, and uncluttered screen	3.27	Neutral
When completing a task, the system keeps screen changes to a minimum	4.07	Agree
The system responds quickly and reduces the number of steps needed to complete tasks	4.20	Agree
The system's overall impression	3.87	Neutral

- **The menu items are well arranged and functions are easy to find**

Figure 5.4 shows that 46.67% of the participants agree that the menu items are arranged well and that it is easy to functions, 33.33% strongly agree in favour of the menu items, 13.33% didn’t agree nor did they disagree, and 6.67% disagreed and none of the participants strongly disagree. This question obtained an MOS of 4.07 meaning the majority of the participants do agree with the arrangement of the menu items.

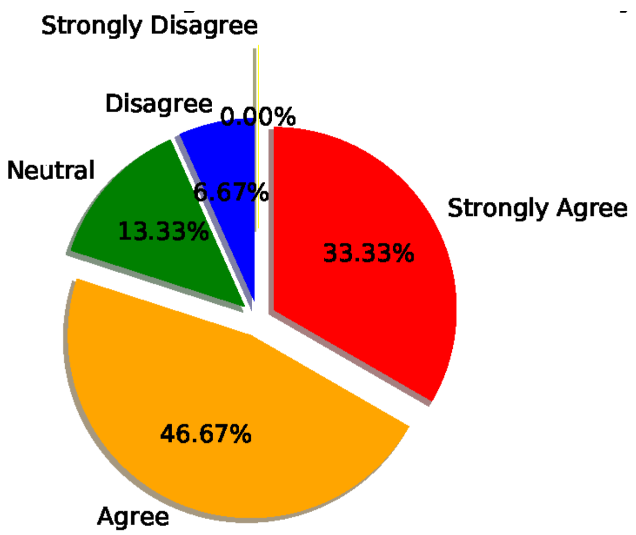


Figure 5.4 The menu items are well arranged and functions are easy to find

- **The functions of each menu item are easily understandable**

In attempt to discover whether the participants were able to understand the functions of each menu item, Figure 5.5 reports that 2 (13.33%) respondents understood all the functions of each menu item immediately, 10 out 15 (66.67%) understood the majority of the functions, 1 out of 15 (6.67%) was not sure of the functions of the menu items and only 1 respondent did not understand the functions of each menu.

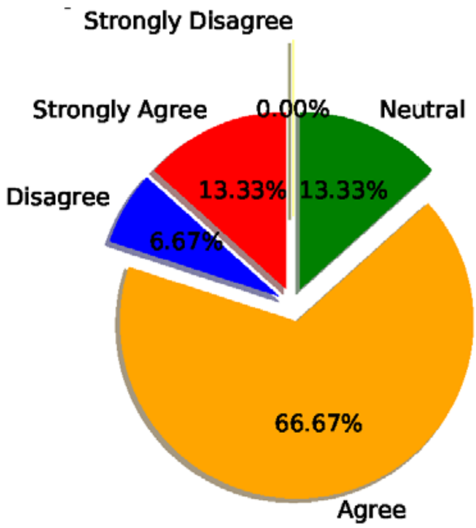


Figure 5.5 The functions of each menu item are easily understandable

- **All the functions I expected to find in the menus are present**

Figure 5.6 shows 13.33% of the participants were a bit disappointed not to find their expected functions and 6.67% of the participants were extremely disappointed (strongly disagree). The results also show that 46.67% (agree) were very happy (strongly agree) that the expected functions were present in the system an 13.33% participants found some of the functions they expected. 20% of the participants voted neutral as they were not sure if what to what functions to expect inn the speaker recognition system. However, this question obtained an MOS of 3.4 which falls under the Neutral category and thus we can conclude that the participants are happy with the functions present in the system but still expected even more functions.

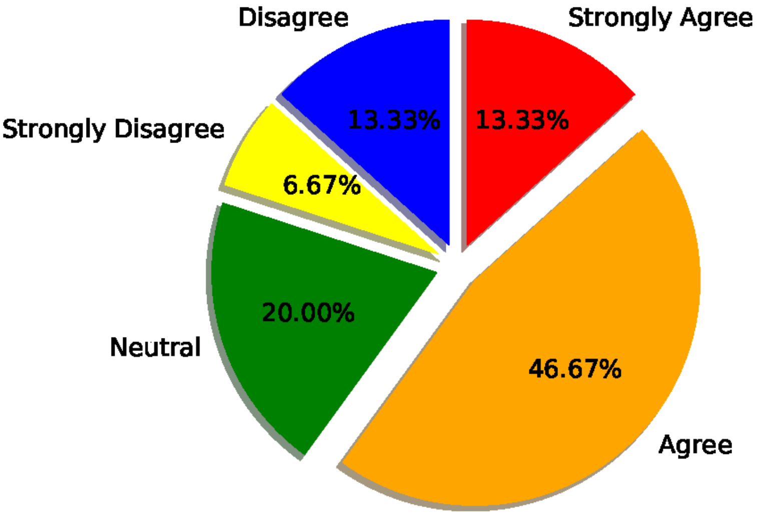


Figure 5.6 All the functions I expected to find in the menus are present

- **The help of a technical person is needed for me to be able to use the system**

It is observed that the system was easy to use as 46.67% (7 out 15) participants managed to use the system without the help of a technical person and a further 13.33% of the participants found their way around the system. It was 20% of the participants who were able to use the system with limited help (Neutral). The results show that 6.67% of the participants strongly needed help on every step in using the application and 13.33% of the participants agreed that they needed help on some of the functions. This question obtained an MOS of 2.53 meaning the majority of the participants did not need help from a technical person for them to be able to use the system.

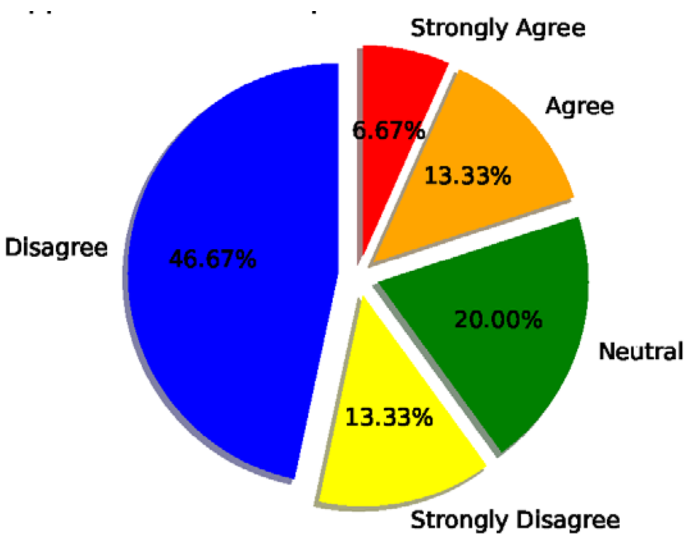


Figure 5.7 The help of a technical person is needed for me to be able to use the system

- **The system is built with a simple, clean, and uncluttered screen**

Figure 5.8 shows that 13.33% of the participants strongly agreed that the mobile application is equipped with clear and clean screen design. Out of the 15 participants, 20% disagreed on the cleanness and clarity of the screen design whereas 33.33% of the participants agree and only 26.67% were not sure where to classify the screen design's cleanness and clarity. As shown in Table 5.6, this question obtained a MOS of 3.27 (Neutral) which means the participants could not determine whether the system had a clear, clean, uncluttered screen design.

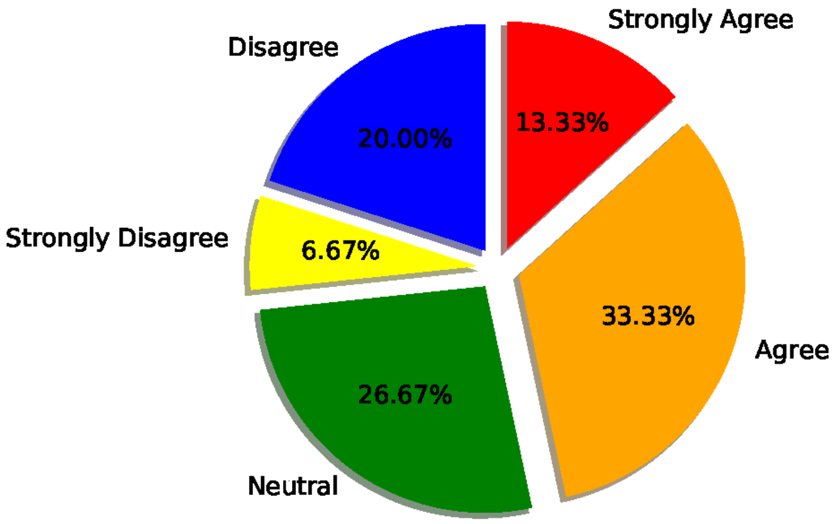


Figure 5.8 The system is built with a simple, clean, and uncluttered screen

- **When completing a task, the system keeps screen changes to a minimum**

The results depicted in Figure 5.9 show that 13.33% of the participants strongly agree that screen changes are kept to a minimum during the completion of a task. The results also show that a further 80% of the participants also agree that the system keeps screen changes to a minimum. Figure 5.9 also depicts that 6.67% of the participants were not sure on this matter and thus did they agree nor disagree. With an MOS of 4.07, we conclude that the system does keeps screen changes to a minimum during the completion of a task and participants are happy.

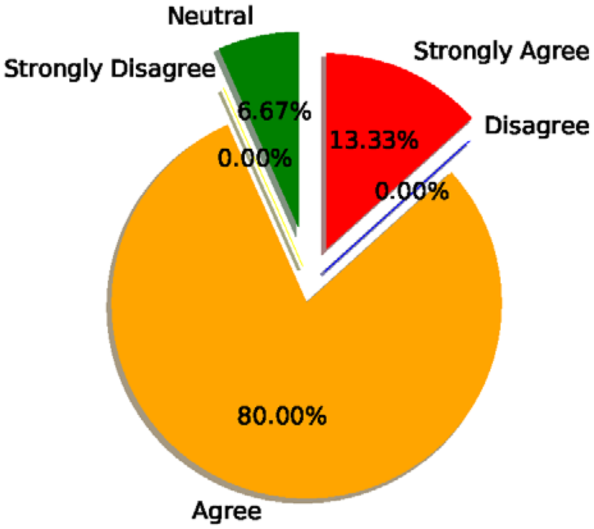


Figure 5.9 When completing a task, the system keeps screen changes to a minimum

- **The system responds quickly and reduces the number of steps needed to complete tasks**

As shown in Figure 5.10, it is observed that 46.67% of the participants strongly agree that the number of screen changes or steps taken to complete a specific task. The results also show that 33.33% of the participants also agree that the number of steps taken to complete a specific task are kept to a minimum. However, 13.33% voted Neutral and 6.67% of the participants disagree and think that the steps taken to complete a specific task can be reduced even further. Some of the participants who disagreed commented that although the system minimizes the number of steps needed to complete a specific task, the system does not respond quickly. This question obtained a MOS of 4.20 meaning that the majority of the participants agree that the system minimizes the number of steps it took to complete tasks and responds quickly.

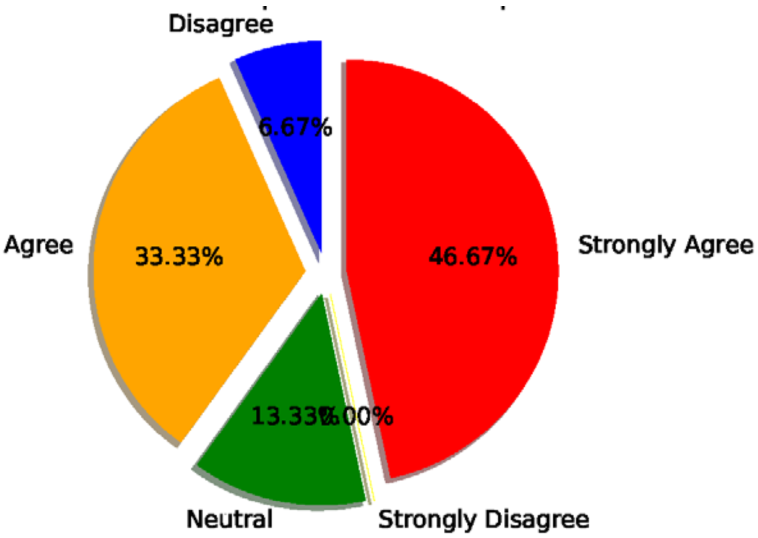


Figure 5.10 The system responds quickly and reduces the number of steps needed to complete tasks

- **The system's overall impression**

It is shown in Figure 5.11 that 33.33% of the 15 participants were very impressed of the system and a further 40% had a positive impression about the system. It is also shown that 13.33% of the participants had both a positive and negative impression (Neutral) about the system whereas only 6.67% of the 15 participants, that is only 1 out of 30 participants had a negative impression, and 1 other participant (6.67%) was not impressed at all (very negative) about the speaker recognition.

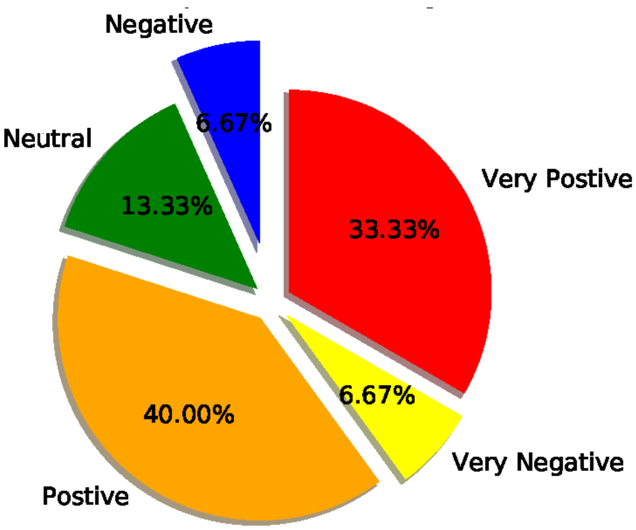


Figure 5.11 The system's overall impression

5.5 Discussion

This section summarises the results and answers the research questions stated in Chapter 1. The main research questions of the study are:

a) Can a speaker recognition system give a significant performance if trained with data collected from speakers of low-resourced languages?

b) What is the effect of a particular spoken language towards the performance of a speaker recognition system?

In Section 5.2, we compare the acoustic features of speech to determine which features perform best given the Sepedi speech data and have found that combining the time, frequency and cepstral domain features gives the best performances. We there use the combined features to train the five classifier models and compare the results. We report the results on classifier model performances and observe that the models that are trained achieve significant and acceptable performances with accuracies of over 75% for all the classifier models discussed and that the MLP classifier model outperforms the other classifier models. We also report in Section 5.3 that the deployed classifier model achieves an accuracy of 80% meaning that it is possible to develop and deploy a speaker recognition system with data collected from speakers of low-resourced languages achieve significant results.

The answers to the above questions are therefore as follows:

Yes, a speaker recognition system can give significant performances if trained with data collected from speakers of low-resourced languages and the performances can be improved if speakers use the system by speaking their native languages.

5.6 Summary

This chapter reported and discussed the results obtained from the experiments performed. Two experiments are described in Section 5.2 where the first experiment investigates the effects of acoustic features towards the performance of the classifier models. The three acoustic features of speech that are investigated are the Time, Frequency and Cepstral domain features and it was reported that Cepstral-domain features give better results in comparison the Time and Frequency-domain features.

It is also reported that the performance improves significantly when these three acoustic features of speech are combined. The second experiment described in Section 5.2 compares the performance of five classifier models (KNN, RF, SVM, LR and MLP). This experiment used the combined acoustic features of speech to train the classifier models and it was reported that the best performances are obtained from the MLP classifier model. The SVM and LR models performed better than KNN and RF.

From the results of the two experiments conducted, the best performing model (MLP) was selected and implemented on the GUI for real-time recognition. Section 5.3 discussed the performance of the graphical user interface and it was reported that the system performed better with **TAR** of 66.67% and a **FRR** of 13.33%. Section 5.4 discussed the GUI’s usability and reported an overall impression MOS of 3.87 implying that most of the participants were impressed by the developed speaker recognition system.

CHAPTER 6: CONCLUSION AND FUTURE WORK

6.1 Introduction

This chapter summarises the conducted research, followed by the challenges and study limitations. The chapter concludes by discussing the contributions and suggestions for future extensions to the study.

6.2 Research Summary

In Chapter 1, we introduced the aim of this research study as to train and develop a speaker recognition system uses the speaker's voices to verify and identify the speaker's identities in order to allow only the speakers who are identified or verified the right to access information systems, devices or services that have to be secured from unauthorized users. In response to the aim of the study, we have set out the following research objectives:

- To acquire pre-recorded Sepedi speech data from publicly available speaker recognition databases.
- To extract acoustic features of speech from the acquired speech data.
- To train speaker classifier models using machine learning algorithms and compare their performances to select the best performing model.
- To deploy the best performing speaker classifier model that determines speaker identities and to verify the claimed speaker identities.
- To develop a graphical user interface that performs real-time automatic speaker recognition capabilities.

The objectives were achieved as follows:

- **Objective 1**: Pre-recorded voices were acquired from the NCHLT, where the Sepedi NCHLT Speech Corpus was selected.
- **Objective 2**: The open-source comprehensive pyAudioAnalysis python library was used for feature extraction of three types of acoustic features of speech from the acquired pre-recorded voices (Section 3.5). An experiment was conducted to investigate which of these features have a significant effect on speaker recognition system's performance (Section 5.2.2). It was observed that cepstral domain

features perform better compared to time and frequency domain features, however combining all three acoustic features of speech gives the best performances.

- **Objective 3**: Five machine learning algorithms (KNN, RF, LR, SVM and MLP) implemented on Scikit-learn where used to train different classifier models and GridSearchCV, also implemented on Scikit-learn, was used to determine the hyper-parameters for each of the five algorithms (Section 3.7). The classifier models were evaluated and the results show that MLP classifier outperforms KNN, RF, LR and SVM classifiers (Section 5.2.2).
- **Objective 4 and Objective 5**: A GUI was developed (Section 4.3.1) and the best performing classifier model, MLP, was implemented on the developed GUI to perform real-time automatic speaker recognition capabilities. The GUI's performance was evaluated and the results showed better performances (Section 5.3).

6.3 Challenges and study limitations

In the past decades, research in the speaker recognition field has focused mainly on smaller population sizes with speech signals recorded telephones resembling real-life conditions. Several studies have achieved good results achieving over 90% in accuracy. Past studies were conducted on different speaker databases, in which some of those speaker databases were privately owned while others were not designed specifically for speaker recognition. Similarly, the speaker database (NCHLT) used in this research was not primarily collected for speaker recognition.

6.4 Future Work and Recommendation

As an extension to the study, the following can be considered:

- Collect more speaker recognition data from native Sepedi language speakers to improve the system's performance.
- Explore other types of acoustic features of speech like Linear Predictive Cepstral Coefficients.
- Explore various deep neural network types, such as deep belief networks and long short-term memory which are deep neural network architectures that have a more efficient training scheme than standard MLPs.

- The GUI can be extended to include multiple speaker recognition functionalities that can be executed in parallel.
- The GUI is developed to run only on Linux/Unix and can be extended to run on multiple platforms such as Windows and Mac operating systems. The Interface can also be extended to run as an online framework in which structured speaker recognition system functions can be performed remotely.

Appendices

Appendix A: Evaluation Form

DEVELOPMENT OF A TEXT-INDEPENDENT AUTOMATIC SPEAKER RECOGNITION SYSTEM

Name and Student number: Mr. TB Mokgonyane (201211351)

To who it may concern

Please assist in evaluating the Graphical User Interface (GUI) for a **Speaker Recognition System** developed as part of a Master of Science Project for postgraduate student Mr. TB Mokgonyane, student number 201211351. The data gathered from this questionnaire is for research purpose only and participants are to remain anonymous throughout this research.

Instructions: Please tick or complete with the appropriate answers on the questionnaire.

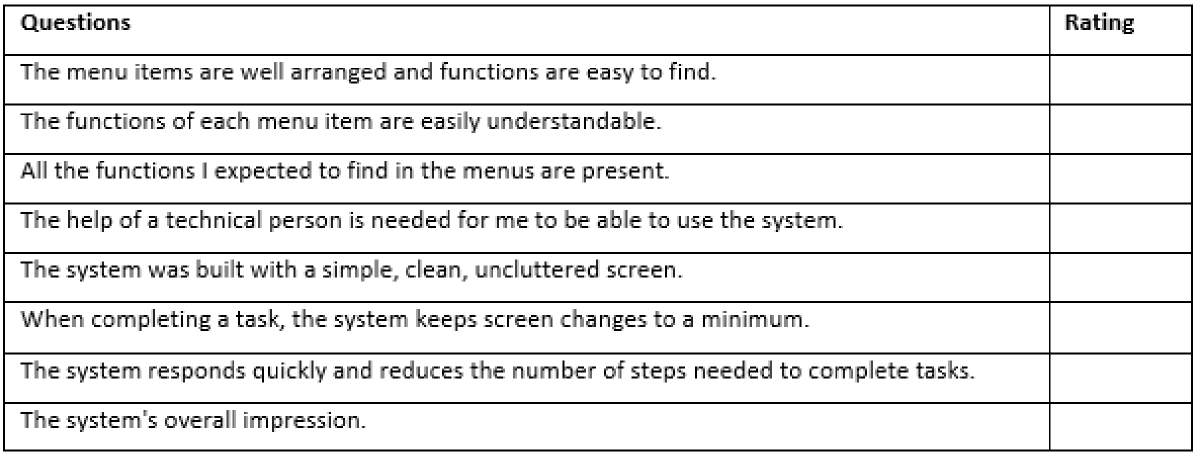
Section 1: General Questions

Gender	
Age	
Home Language	

Section 2: Usability Questions

Question 1: Please rate the system usability based on a scale of 1-5, where:

1 = Strong Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, and 5 = Strongly Agree



Questions	Rating
The menu items are well arranged and functions are easy to find.	
The functions of each menu item are easily understandable.	
All the functions I expected to find in the menus are present.	
The help of a technical person is needed for me to be able to use the system.	
The system was built with a simple, clean, uncluttered screen.	
When completing a task, the system keeps screen changes to a minimum.	
The system responds quickly and reduces the number of steps needed to complete tasks.	
The system's overall impression.	

Question 2: Comment on given ratings (optional):

REFERENCES

Adamski, M.J., 2013. *A speaker Recognition Solution for Identification and Authentication*. M.Com. (Informatics) Unpublished: University of Johannesburg.

Aha, D.W., Kibler, D. & Albert, M.K., 1991. Instance-based learning algorithms. *Machine Learning*, 6(1), pp. 37–66.

Bachu, R., Kopparthi, S., Adapa, B. & Barkana, B.D., 2010. Voiced/unvoiced decision for speech signals based on zero-crossing. In *Advanced Techniques in Computing Sciences*., Springer, pp. 79–282.

Baharipour, H., Ahmadabadi, M.E. & Mosleh, M., 2014. A Study of Speaker Recognition Approaches Based on Feature Selection and Classification Methods. *International journal of Computer Science & Network Solutions*, 2(9), pp. 61-67.

Barnard, E., Davel, M.H., van Heerden, C., de Wet, F. & Badenhorst, J., 2014. The NCHLT Speech Corpus of the South African languages. In *Fourth International Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU-2014)*. St. Petersburg, Russia, pp. 194-200.

Bimbot, F. *et al.*, 2004. A Tutorial on Text-Independent Speaker Verification. *EURASIP Journal on Applied Signal Processing*, 2004(4), pp. 430–451.

Breiman, L., 2001. Random forests. *Machine Learning*, 45(1), pp. 5-32.

Brownlee, J., 2014. *Classification Accuracy is Not Enough: More Performance Measures You Can Use*. [Online] Available at: https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/ [Accessed 27 Augustus 2020].

Casserly, E.D. & Pisoni, D.B., 2013. Speech perception and production. *Wiley Interdisciplinary Reviews: Cognitive Science*, 1(5), pp. 629–647.

Census, 2011. *Statistics South Africa (STATS SA)*. [Online] Available at: http://www.statssa.gov.za/publications/Report-03-01-78/Report-03-01-782011.pdf [Accessed 14 June 2018].

Chang, C.-C. & Lin, C.-J., 2011. LIBSVM : A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3), pp. 1-27.

Charan, R., Manisha, A., Karthik, R. & Kumar, M.R., 2017. A text-independent speaker verification model: A comparative analysis. In *2017 International Conference on Intelligent Computing and Control (I2C2)*., pp. 1-6.

Chauhan, N. & Chandra, M., 2017. Speaker recognition and verification using artificial neural network. In *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*., pp. 1147-1149.

De Vries, N.J., Davel, M.H., Badenhorst, J., Basson, W.D., de Wet, F., Barnard, E. & de Waal, A., 2014. A smartphone-based ASR data collection tool for under-resourced languages. *Speech Communication*, 56, pp. 119–131.

de Wet, F., Badenhorst, J. & Modipa, T., 2016. Developing Speech Resources from Parliamentary Data for South African English. In *SLTU-2016 5th Workshop on Spoken Language Technologies for Under-resourced languages*. Yogyakarta, Indonesia

Dey, N.S., Mohanty, R. & Chugh, K.L., 2012. Speech and Speaker Recognition System Using Artificial Neural Networks and Hidden Markov Model. In *2012 International Conference on Communication Systems and Network Technologies*., pp. 311-315.

Fenglei, H. & Bingxi, W., 2000. An integrated system for text-independent speaker recognition using binary neural network classifiers. In *WCC 2000 - ICSP 2000. 2000 5th International Conference on Signal Processing Proceedings. 16th World Computer Congress 2000*., pp. 710-713 vol.2.

Ferbrache, D., 2016. Passwords are broken – the future shape of biometrics. *Biometric Technology Today*, 2016(3), pp. 5-7.

Furui, S., 2005. 50 years of progress in speech and speaker recognition. *ECTI Trans. on Computer and Information Technology*, 1(2), pp. 67-74.

Gbadamosi, L., 2013. Text Independent Biometric Speaker Recognition System. *International Journal of Research in Computer Science*, 3(6), pp. 9-15.

Giannakopoulos, T., 2015. pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis. *PLoS one*, 10(12) Available at: https://doi.org/10.1371/journal.pone.0144610.

Hamid, L., 2015. Biometric technology: not a password replacement, but a complement. *Biometric Technology Today*, 2015(6), pp. 7-10.

Harrell, F.E., 2015. Ordinal logistic regression. In *Regression modeling strategies*. Springer. pp. 311-325.

Hashimoto, K., Yamagishi, J. & Echizen, I., 2016. Privacy-Preserving Sound to Degrade Automatic Speaker Verification Performance. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Shanghai, China, IEEE

Huang, X., Acero, A., Hon, H.-W. & Reddy, R., 2001. *Spoken language processing: A guide to theory, algorithm, and system development*. Upper Saddle River: Prentice hall PTR.

Jain, A.k., Nandakumar, K. & Ross, A., 2016. 50 years of biometric research: Accomplishments, challenges, and opportunities. *Pattern Recognition Letters*, 79, pp. 80-105.

Kacur, J., Vargic, R. & Mulinka, P., 2011. Speaker identification by K-nearest neighbors: Application of PCA and LDA prior to KNN. In *2011 18th International Conference on Systems, Signals and Image Processing*., pp. 1-4.

Kamruzzaman, S.M., Karim, A.M.N.R., Islam, M.S. & Haque, M.E., 2010. Speaker Identification using MFCC-Domain Support Vector Machine. *International Journal of Electrical and Power Engineering*, 1(3), pp. 274–278.

Katz, M., Schaffoner, M., Andelic, E., Kruger, S.E. & Wendemuth, A., 2006. Sparse Kernel Logistic Regression using Incremental Feature Selection for Text-Independent Speaker Identification. In *2006 IEEE Odyssey - The Speaker and Language Recognition Workshop*., pp. 1-6.

Kekre, H.B. & Kulkarni, V., 2013. Closed set and open set Speaker Identification using amplitude distribution of different Transforms. In *2013 International Conference on Advances in Technology and Engineering (ICATE)*., pp. 1-8.

Kiktova, E. & Juhar, J., 2015. Speaker Recognition for Surveillance Application. *Journal of Electrical and Electronics Engineering*, 8(2), pp. 19-22.

Kingma, D.P. & Ba, J., 2014. Adam: A Method for Stochastic Optimization.

Kinnunen, T. & Li, H., 2010. An Overview of Text-Independent Speaker Recognition: from Features to Supervectors. *Speech Communication*, 52(1), pp. 12-40.

Larcher, A., Lee, K.A., Ma, B. & Li, H., 2014. Text-dependent speaker verification: Classifiers, databases and RSR2015. *Speech Communication*, 60, pp. 56-77.

Liu, Y., Qian, Y., Chen, N., Fu, T., Zhang, Y. & Yu, K., 2015. Deep feature for text-dependent speaker verification. *Speech Communication*, 73, pp. 1-13.

Marciniak, T., Weychan, R., Stankiewicz, A. & Dąbrowski, A., 2014. Biometric speech signal processing in a system with digital signal processor. *Bulletin of the Polish Academy of Sciences, Technical Sciences*, 62(3), pp. 589-594.

Mazibuko, T. & Mashao, D., 2007. Feature Normalization in SVM Speaker Verification using Telephone Speech. In *Southern Africa Telecommunication Networks and Applications Conference (SATNAC)*.

Mokgonyane, T.B., Sefara, T.J., Manamela, M.J. & Modipa, T.I., 2018. Development of a Text-Independent Speaker Recognition System for Biometric Access Control. In *Southern African Telecommunication and Networks and Application Conference (SATNAC) 2018*. Arabella, Western Cape, South Africa, pp. 128-133.

Mokgonyane, T.B., Sefara, T.J., Modipa, T.I., Mogale, M.M., Manamela, M.J. & Manamela, P.J., 2019. Automatic Speaker Recognition System based on Machine Learning Algorithms. In *2019 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA)*. Bloemfontein, South Africa, pp. 141-146.

Panda, K.A., Sahoo, A.K. & Meher, S., 2011. *Study of speaker recognition systems*. Thesis. Rourkela: National Institute of Technology, Rourkela.

Pedregosa, F. *et al.*, 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12(Oct), pp. 2825--2830.

Pyrtuh, F., Jelil, S., Kachari, G. & Joyprakash Singh, L., 2013. In *2013 Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*., pp. 1-4.

Rajalakshmi, P. & Anju, L., 2017. Feature Extraction and Speaker Identification in Automatic Speaker Recognition System. *International Journal of Innovative Research in Computer and Communication Engineering*, 5(3), pp. 143-148.

Ramachandran, R.P., Farrell, K.R., Ramachandran, R. & Mammonec, R.J., 2002. Speaker recognition - general classifier approaches. *Pattern Recognition: The Journal of pattern recognition society*, 35, pp. 2801-2821.

Ranny, 2016. Voice Recognition Using k Nearest Neighbor and Double Distance Method. In *2016 International Conference on Industrial Engineering, Management Science and Application (ICIMSA)*., pp. 1-5.

Rao, M.S., Lakshmi, G.B., Gowri, P. & Chowdary, K.B., 2020. Random Forest Based Automatic Speaker Recognition System. *The International journal of analytical and experimental modal analysis*, 8(4), pp. 526-535.

Reynolds, D.A., 1995. Automatic Speaker Recognition Using Gaussian Mixture Speaker Models. *The Lincoln Laboratory Journal*, 8(2), pp. 173-191.

Richardson, F., Reynolds, D. & Dehak, N., 2015. Deep neural network approaches to speaker and language recognition. *IEEE Signal Processing Letters*, 22(10), pp. 1671-1675.

Sahoo, J.K. & Rishi, D., 2014. Speaker Recognition using Support Vector Machines. *International Journal of*, 2(2), pp. 01-04.

Shashidhara, B.M., Jain, S., Rao, V.D., Patil, N. & Raghavendra, G.S., 2015. Evaluation of machine learning frameworks on bank marketing and higgs datasets. In *2015 Second International Conference on Advances in Computing and Communication Engineering*., pp. 551-555.

Siddique, K., Akhtar, Z. & Kim, Y., 2017. Biometrics vs passwords: a modern version of the tortoise and the hare. *Computer Fraud & Security*, 2017(1), pp. 13-17.

Singh, N., Khan, R.A. & Shree, R., 2012. Applications of Speaker Recognition. *Procedia Engineering*, 38, pp. 3122-3126.

Solewicz, Y.A. & Koppel, , 2005. Selective Fusion for Speaker Verification in Surveillance. In *Intelligence and Security Informatics: IEEE International Conference* *on Intelligence and Security Informatics*. Dept. of Computer Science, Bar-Ilan University, Ramat-Gan, Israel, IEEE

Sreelekshmi, S.K. & Syama, R., 2017. Speaker Identification Using K-Nearest Neighbors (k-NN) Classifier Employing MFCC and Formants as Features. *International Journal of Advanced Scientific Technologies ,Engineering and Management Sciences*, 3(1), pp. 234-238.

Staroniewicz, P. & Majewski, W., 2004. SVM based text-dependent speaker identification for large set of voices. In *2004 12th European Signal Processing Conference*. Vienna, Austria, IEEE, pp. 333-336.

Tiwari, V., 2010. MFCC and its applications in speaker recognition. *International Journal on Emerging Technologies*, 1(1), pp. 19-22.

Wang, Y. & Lawlor, B., 2017. Speaker recognition based on MFCC and BP neural networks. In *2017 28th Irish Signals and Systems Conference (ISSC)*., pp. 1-4.

Wildermoth, B.R. & Paliwal, K.K., 2003. GMM based speaker recognition on readily available databases. In *Microelectronic Engineering Research Conference, Brisbane, Australia*., p.55.