André Filipe Magalhães

VOICE RECOGNITION OF USERS FOR VIRTUAL ASSISTANT IN INDUSTRIAL ENVIRONMENTS

1 2 9 0 UNIVERSIDADE D COIMBRA

1 2 9 0

UNIVERSIDADE D
COIMBRA

André Filipe da Silva Magalhães

VOICE RECOGNITION OF USERS FOR VIRTUAL ASSISTANT IN INDUSTRIAL ENVIRONMENTS

VOLUME 1

Dissertation in the context of the Master in Informatics Engineering, Specialization in Intelligent Systems advised by Professor Doutor João Nuno Gonçalves Costa Cavaleiro Correia and Professor Doutor Tiago José dos Santos Martins da Cruz presented and to the Faculty of Sciences and Technology / Department of Informatics Engineering.

September de 2021

Faculty of Sciences and Technology

Department of Informatics Engineering

Voice recognition of users for virtual assistant in industrial environments

André Filipe da Silva Magalhães

Dissertation in the context of the Master in Informatics Engineering, Specialization in Intelligent Systems advised by Professor Doutor João Nuno Gonçalves Costa Cavaleiro Correia and Professor Doutor Tiago José dos Santos Martins da Cruz presented and to the Faculty of Sciences and Technology / Department of Informatics Engineering.

September 2021

UNIVERSIDADE D
COIMBRA

This page is intentionally left blank.

Abstract

With a growth in the number of devices with a greater computational capacity, the need to innovate the human-machine interaction was necessary. Furthermore, with the current technological advances in speech processing and natural language processing, the possibility of interacting with devices has been created in the most natural way human beings have to communicate, the voice.

In the context of this internship, we analyse virtual assistants and techniques for recognising the sound produced to authenticate and authorise user commands. To pursue these objectives, we have explored Mycroft AI and extended its framework. Furthermore, was developed an algorithm for creating models for user recognition. In addition, to perform user recognition through Mycroft AI, a REST Server API was created to provide the necessary resources for that purpose. With this, the recognition is carried out through the communication of these two systems (Mycroft AI and API REST Server).

For the creation of the speaker identification system, the main component of the API Server, the set of features used were the combination of MFCC, Chroma, Spectral (centroid, contrast and rolloff), RMS and Zero Crossing Rate. Additionally, as preprocessing, a trimming technique was used. Finally, as modelling techniques, we use Neural Network (Multilayer Perceptron) and Linear Discriminant Analysis (LDA). The public datasets used to validate this approach are TIMIT, NOIZEUS, LibrisSpeech ARS. As a result, Multilayer Perceptron (MLP) was slightly superior to Linear Discriminant Analysis (LDA), being able to recognize a set of 462 different users.

Keywords

Voice Assistance; Authentication; Authorisation; Speaker Recognition Systems; Machine Learning;

This page is intentionally left blank.

Resumo

Com o crescimento do número de dispositivos e aumento da sua capacidade computacional, a necessidade de inovar a interação com os diferentes dispositivos a aplicações surge. Com os atuais avanços tecnológicos no processamento da fala e no processamento natural da linguagem, tornou-se possível de interagir com os dispositivos da forma mais natural que os seres humanos têm para se comunicar: a voz.

No contexto deste estágio, analisamos alguns assistentes virtuais assim como técnicas de reconhecimento dos sons produzidos para autenticar e autorizar os comandos do utilizador. Para atingir esses objetivos, explorados o Mycroft AI e estendemos a sua framework. Foi desenvolvido um algoritmo para a criação dos modelos de reconhecimento dos utilizadores. Adicionalmente, para realizar o reconhecimento dos utilizadores através do Mycroft AI, foi criado um servidor API REST que fornece os recursos necessários para esse propósito. Com isto, o reconhecimento é realizado através da comunicação desses dois sistemas (Mycroft AI e servidor REST API).

Para a criação do Speaker Recognition System, a principal componente do servidor REST API, o conjunto de features utilizadas foi a combinação das MFCC, Chroma, Spectral (centroid, contrast and rolloff), RMS and Zero Crossing Rate. Como preprocessamento foi utilizada uma técnica de trimming. Por fim, como técnicas de modelação, foram utilizadas as redes neuronais (Multilayer Perceptron) e Linear Discriminant Analysis (LDA). Os datasets públicos TIMIT, NOIZEUS e LibrisSpeech ARS. Como resultados finais, as redes neuronais (Multilayer Perceptron) saírem ligeiramente superior em comparação ao Linear Discriminant Analysis (LDA) e é capaz de reconhecer um conjunto de 462 diferentes de utilizadores.

Palavras-Chave

Assistente de Voz; Autenticação; Autorização; Assistente de Reconhecimento de Voz; Aprendizagem Computacional

This page is intentionally left blank.

Contents

This page is intentionally left blank.

Acronyms

API Application Programming Interface. iii, v, 28, 29

CNN Convolutional Neural network. 16

DNN Deep Neural Network. 16, 17

GMM Gaussian Mixture Model. 7, 15, 16

LDA Linear Discriminant Analysis. iii, v, 7, 10, 25, 34–37, 39

MFCC Mel-Frequency Cepstral coefficients. iii, v, 16, 27

MLP Multilayer Perceptron. iii, v, xiii, 25, 34, 35, 39

MVC Model-Controller-View. 29

NLP Natural Language Processing. 11

NN Neural Network. iii, xiii, 7, 8, 15, 34–36

PLDA Probabilistic Linear Discriminant Analysis. 17

REST Representational State Transfer. iii, v, 29

RMS Roor-Mean-Square. iii, v, 27

SIS Speaker Identification System. 13, 14, 28–30

SRS Speaker Recognition System. 5, 13–16, 39

STT Speech To Text. 11

SVM Support Vector Machines. 16

SVS Speaker Verification System. 13, 14

TTS Texto To Speech. 11, 12

VQ Vector Quantization. 7, 9, 15, 16

This page is intentionally left blank.

List of Figures

This page is intentionally left blank.

List of Tables

This page is intentionally left blank.

Chapter 1

Introduction

The growth in the number of devices and technological advances creates a need to explore and improve new ways for human-machine interaction. With this, the research and development conducted in Artificial Intelligence made possible more natural forms of inputs, such as voice. Consequently, and with technological advances in speech processing and natural language processing, systems are offered the ability to interact with them through the voice and understand what is intended to be done. These systems are called voice assistants and have grown exponentially in their use. Currently, these systems' capabilities are significantly untapped in industrial environments. Therefore, given this highly automated context, the easy interaction with these systems will focus on streamlining and supporting certain processes. Within an entity, it is normal for people to perform different tasks, and as such, they may be associated with a type of role. Good data management and control must offer the permissions for accessing and handing the systems to the correct roles. The overall process is called authorisation, typically defined by a system that controls the different users' permissions/accesses. To perform the proper authorisation, it is necessary to identify the user correctly, which is done by authentication. In short, combining the technology of virtual voice assistants with some AI techniques, creating an authentication method using people's voices is encouraged. Under an industrial context, artificial intelligence topics can offer a much more natural and intuitive interaction with systems, leading to the optimisation of industrial processes.

1.1 Objectives

Regarding the defined proposal, its main objectives and expected results, after a first clarification and discussion meeting, were defined as follows:

- Explore open-source voice assistants;
- Study and develop techniques for the creation of machine learning models for the recognition of users through their voice;
- Integrate the machine learning model in the voice assistant to identify users;

With this in mind, the expected and result is a virtual voice assistant with an integrated machine learning model for user identification. Once the final objective is known, to better prepare the planning of this work, these objectives are presented at a more detailed level. The following objectives can be classified as mandatory and optional, that is, the success

of this project means that the main (mandatory) objectives were successfully completed, and the rest are considered as extra (optional) objectives. Thus, the goals are presented as:

- (Mandatory) Datasets: identify data to be used in the creation of machine learning models;
- (Mandatory) Technologies: identify and explore technologies for the development of this system;
- (Mandatory) Speaker Recognition System: to develop a system capable of identifying users through their voice, this includes data processing techniques (speechs) and modelling;
- (Extra) Performance: depending on the problem of identifying users through their voice, explore the main and possible sub-problems;
- (Mandatory) Virtual voice assistants: understand their functioning and use;
- (Mandatory) Integration: integrate the pre-trained model into a virtual voice assistant for user identification;
- (Extra) Creation of a dataset;
- (Extra) Implementation of user permission management depending on their identification;

In short, the project is summed up in the implementation and validation of a speaker recognition system integrated into a virtual assistant.

1.2 Main Contributions

Part of this work (thesis) received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017226 – Project 6G BRAINS. Its main contributions identified with the completion of this project are:

- Base of a Server API using REST with an MVC architecture and PostgreSQL database configured for future projects development
- Development of an algorithm for speech processing
- Development of modelling techniques to classify users using their biometric data (voice)
- Two new features (Authentication and Authorisation) added to Mycroft AI. Integration of the speaker recognition system.

1.3 Outline

The remainder of this document is structured as follows:

- **Chapter Two State of the Art** - An overview of research work related to the project is presented. Additionally, concepts about the leading technologies and methods to be used are described and introduced.
- **Chapter Three Methodology and Planning** - The methodology used for the development of the project is introduced, and an analysis of the planning is carried out before the beginning of the project as a function of the work carried out.
- **Chapter Four System Overview** - A description for the implementation of the system, as well as its functionalities, workflow, architecture and technologies.
- **Chapter Five Experimentation** - The experiment performed is detailed in this chapter. First, its setup is explained, that is, the datasets used, the algorithms and respective parameters and, finally, the obtained results are presented together with analysis.
- **Chapter Six Conclusion** - Final conclusion on the development of this project as well as future work is presented

This page is intentionally left blank.

Chapter 2

State of the Art

In this chapter, the main topics covered are Virtual Voice Assistants, authentication and authorisation systems, and finally, the Speaker Recognition System (SRS). First, some definitions and concepts regarding possible approaches to processing and machine learning to speech data are introduced. Then, the concepts of authentication and authorisation are explained, showing their advantages and disadvantages. After that, the Virtual Voice Assistants are described, their functionalities and a comparative study of the different assistants selected. Before introducing the related work, the Speaker Recognition System, its traditional architecture and different pre-processing, processing and modelling techniques are presented. Finally, the research related to this project, where its approaches and respective results are mentioned, is presented. The chapter ends with a conclusion of the main aspects taken from the research carried out.

2.1 Concepts and Definitions

Overall, human intelligence follows a simple, straightforward, and typically predictable pattern. It collects information, processes this information and uses it to decide how to act, in other words, learning is possible through the observation of something, identification of a certain pattern, construction of a theory that explains this pattern and the validation of this theory by checking if the same is identified in other observations. In this way, the main aspects of human intelligence are very similar to artificial intelligence, that is, in the same way that the human collects information, processes, and determines an output, the machine is also capable of doing this [5, 21]. Thus, given this classification problem, the data processing process and computational learning models, combining their approaches, are intended to simulate the same learning behaviour that human beings perform. Studying algorithms/architectures to define the model is fundamental, however, data processing is an essential and indispensable process that influences the choice of the model and its performance. Therefore, this section introduces some concepts and definitions regarding data processing and machine learning, considering the problem explained in Chapter 1.

2.1.1 Speech Data Processing

Simply put, this is a process that, given the context of the problem, intends to convert the speech waveform into a parametric representation (unique for each user) and used to the identification of the different users. This process is known as feature extraction. Having

the data in a format that can be interpreted by the system, there are a set of techniques that can be applied in order to optimise the training process as well as improve your performance. These approaches are called feature selection, feature reduction, and feature scaling. Feature selection is defined by the reduction of the initial set of features without the existence of any type of transformation. Feature reduction there is also a reduction of features, however, there is a transformation in the features relative to the initial set. The idea of these two approaches is represented in Figure 2.1.

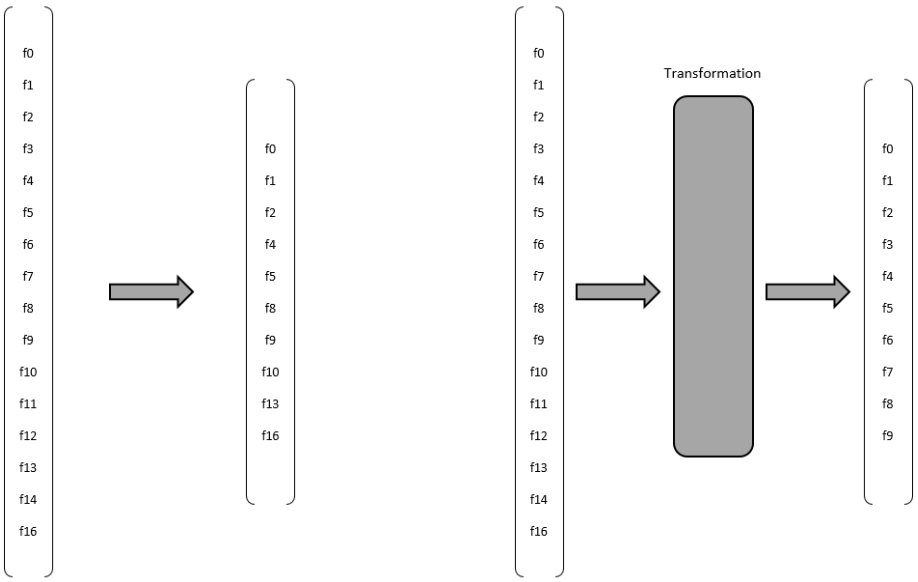


Figure 2.1: Feature Selection vs Feature Reduction

As you can see, in the feature selection process, we select the initial set of features. In feature reduction, we apply a transformation to the initial set of features getting a new set of features. Feature scaling is a process that aims to normalise the range of feature values. For example, considering a dataset composed of age, salary, and height, we would have values between 18 to 80 years, 20.000 to 100.000 Euros, and 1 to 2 meters, respectively, feature scaling would transform the values to be in the same range of values. To perform data scaling there are several methods, however, of all existing approaches, the most common is normalisation and standardisation. Normalisation is the process of scaling feature data to a range between two values, and standardisation causes the values of each feature to be centred around the mean with a certain standard deviation. The normalisation, unlike standardisation, is used when the data does not follow a Gaussian distribution [23, 57, 67, 68]. Additionally, standardisation, unlike normalisation, is not affected by possible outliers present in the data. With this, the data processing workflow, generically, can be translated by Figure 2.2.

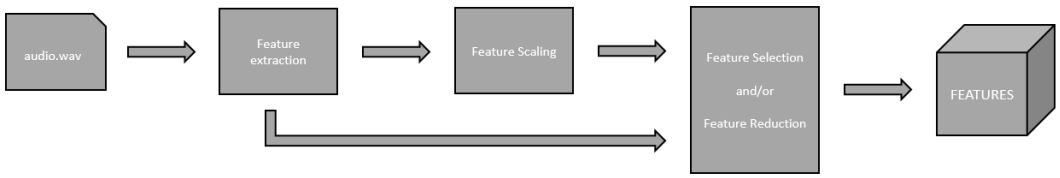


Figure 2.2: Speech Data Processing Workflow

In short, it is possible to visualise the main data processing techniques that can be applied before any kind of algorithm. However, given the context of the problem, there is a pre-processing phase. Its main objective is to overcome problems, such as speech enhancement, channel compensation, and voice activity detection, which improve the final model's performance.

2.1.2 Machine Learning to Speech Data

Applying all the necessary approaches regarding data processing, it is needed to use the algorithms to train/build the part of the system capable of receiving data never seen before and correctly identifying the users. Regarding modelling techniques, we have two distinct methods, which are called supervised learning and unsupervised learning [18, 63]. In supervised learning techniques, its objective is to learn a function that, given a data sample and respective label, best approximates the relationship between observable input and output in the data. On the other hand, unsupervised learning has no label outputs, so its goal is to infer the natural structure present in a set of data points. Therefore, the main difference between the two types is that supervised learning is done using ground truth, or in other words, we have prior knowledge of what the output values for our samples should be. This section will introduce and explain algorithms for modelling capable of identifying users through their voice, such as Gaussian Mixture Model (GMM), Neural Network (NN), Vector Quantization (VQ) and Linear Discriminant Analysis (LDA).

Gaussian Mixture Model (GMM) [6, 13, 33, 56] is a probabilistic model that attempts to represent usually distributed sub-populations inside a general population. Since they do not require and do not know which sub population a data point belongs to, this approach allows the model to learn the sub populations automatically, thus being unsupervised learning. This model is parameterised by two types of values, the mixture component weights and the components means and variances/covariances. If the number of components is known, the most used technique to estimate the mixture model's parameters is the expectation maximisation, a numerical method for maximum likelihood estimation otherwise, if the number of components is unknown, the most common approach its try to guess that number fit that model to the data using expectation maximisation as well.

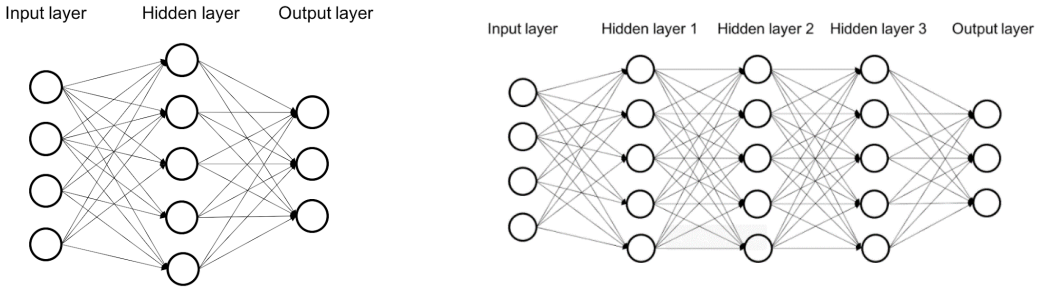
Expectation maximisation, an iterative algorithm, is guaranteed to approach a local or saddle maximum point due to its property that the maximum probability of the data strictly increases with each subsequent iteration. For mixture models, expectation maximisation consists of two steps known as expectation and maximisation.

- First, Expectation (E), which consists of calculating the expectation of the component for each data point given the model parameters.

- Second, Maximisation (M) consists of maximising the expectation calculated in the first step concerning the model parameters, updating the model parameters values.

The expectation step corresponds to the latter case, while the maximisation step corresponds to the former. Thus, by alternating between which values are assumed fixed or known, maximum likelihood estimates of the non-fixed values can be calculated efficiently. Once the expectation maximisation algorithm has finished, the model can perform various forms of inference. The two most common are density estimation and clustering.

Neural Network (NN) [7] is a computer program that tries to simulate similarly to the human brain. Their objective is to perform those cognitive functions our brain can perform, like problem-solving and being teachable. The power of neural networks comes from their ability to learn from a given training data and how to best relate it to the output variable you want to predict. Typically, a neural network is composed of an input layer, an output layer and **N** hidden layers where **N** should be equal or more than one. Depending on the hidden layers, the neural network can be classified as deep or shallow.



(a) Shallow Neural Network (b) Deep Neural Network

Figure 2.3: Example: Shallow and Deep Neural Networks [7]

As you can see in Figure 2.3(a) and 2.3(b), each layer are composed of neurons. Neuron forms the basic structure of a neural network. Receives an input, processes it, and generates an output that is either sent to other neurons for further processing or the final output. For better understanding, the following Figure 2.4, represent a simple neuron with a three inputs.

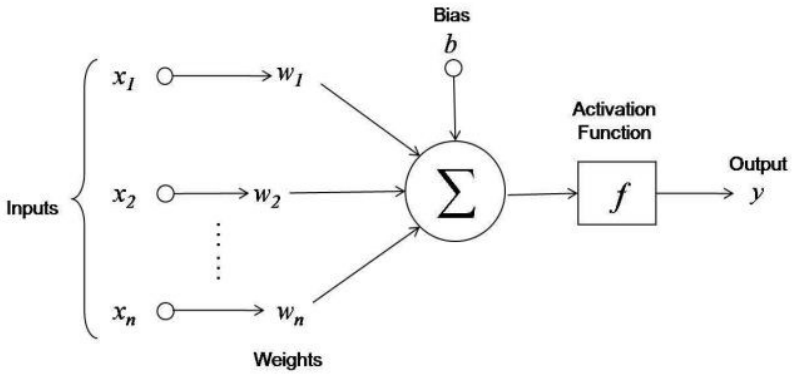


Figure 2.4: Example: Neuron

First, we need to understand which operations are performed and variables are part of the equation that results in the neural network's final output. Considering that, we need to

consider four thinks: **Inputs**, **Weights**, **Bias** and **Active Functions**. The neuron get the information (Input) and those inputs (X_1, X_2 and X_3) will multiply for their respective weights (W_1, W_2 and W_3), that result will be express by summation of those operations, $\sum_{i=1}^{n} X_i W_i$. In addition, another value is added to that operation, the Bias. Finally, the result is passed a function, active function, given the output of the neuron. However, the neuron output is given to a function that will give the final output. These functions are called active functions [59] and determine, depending on the function, its output. With that in mind, other important aspects need to be known. Cost/Loss function is the function that measures the accuracy of the network. The goal of the network is to predict a value as close as possible to the real value, and with this, the cost/lost function have the job of penalising the network, then the prediction is wrong. Given this, the objective during the training phase is to increase the accuracy of the forecast and reduce that error. Also, during the training process, the model structure, its mutable parameters are updated in response to the estimated error, however, this update can be controlled by the hyperparameter knows as the learning rate. Setting this hyperparameter is one of the most important steps for the successful training of a good model, and this is because the choice of a minimal value can result in a long training process that gets stuck, whereas a value too large may result in learning a sub-optimal set of weights too fast or an unstable training process. Then, to improve the training process and network performance and solve problems such as overfitting, we can perform the training in batch, making the model more generalised. Instead of sending all the input at once, we randomly split the input into several pieces of equal size. Dropout can also be used to prevent overfitting, which is a layer parameter that will randomly drop some neurons in the hidden layer.

Vector Quantization (VQ) [8] is a technique where the models are formed by clustering the feature vectors into K non-overlapping clusters, and each that cluster represents a different class. For each cluster, it is represented by a code vector which in this case is it's the centroid. Having this, a set of code vectors is called codebook and their objective to classify the new data through the squared error between the new data and the codebook. The one with the smallest error belonging to the respective cluster. As an example, Figure 2.5 is presented.

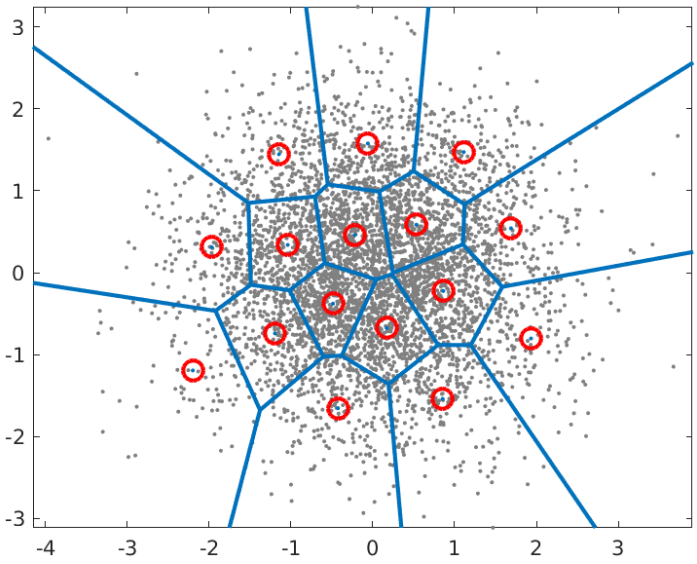


Figure 2.5: Example: Vector Quantization Clusters and Codebook [8]

In the present Figure 2.5, we can identify the cluster as well their centroid, the code vectors. The key point of this approach is the techniques to find the best codebook, which will reflect the data classification performance.

Linear Discriminant Analysis (LDA) [64], it is traditionally used as dimensionality reduction technique in processing step for pattern-classification and machine learning applications. However, it can also be used as a supervised classification algorithm. Considering a variable **X** comes one of **K** classes, with some class-specific probability densities **f(x)**. A discriminant rule tries to divide the data space into K disjoint regions that represent all classes. Classification means that we allocate **x** to class **j** if **x** is in region **j**. To know which region the data belongs, two allocation rules can be follow:

Maximum likelihood rule: If we assume that each class could occur with equal probability, then allocate data **x** to class **j** if

$$j = arg\, max_i f_i(x)$$

Bayesian rule: If we know the class prior probabilities, π, then allocate data **x** to class **j** if

$$j = arg\, max_i\, \pi_i f_i(x)$$

2.2 Authentication and Authorisation

With the evolution of technology and the emergence of systems that deal with people's private information, there was a need to make those systems as secure as possible and make the correct information available to the right people. With that in mind, two important concepts are introduced [53]: Authentication and Authorisation. Authentication is the process that analyses the user's identity before granting access to the system, that is, verifying the user's identity. This functionality has become essential in the development of most systems and is currently present in any application. We can identify the existence of six types of authentication methods, and they can be listed as [29]:

- Password-Based Authentication - The user accesses the system through an email/username and password after registering with the same details in the system in question.
- API Authentication - This is a process of certifying the user's identity who tries to access the services of a server.
- Passwordless Authentication - The user accesses the system through a link or an OTP sent by email or text message.
- Social Authentication - Through this method, the user uses the credentials existing in one of his social accounts to access the system.
- Biometric Authentication - The system validates his identity by using the individual's unique biological characteristics (fingerprint, face recognition, iris or voice).

Given these methods, it is necessary to understand that each has its advantages and disadvantages. Those advantages and disadvantages are presented in Table 2.1.

Authentication method	Advantages	Disadvantages
Password-Based Authentication	Easy to implement; Simple to deploy;	Security is entirely based on the confidentiality and complexity of the password; User identification based on password only;
API Authentication	Easy to implement;	The absence of encryption makes the security risk relatively high;
Social Authentication	Faster registration; Social login for social features; Possible fewer failed logins;	Social networks/logins are sometimes blocked; Loss of control for thrid parties;
Passwordless Authentication	Enhanced user experience; Greater security;	Potentially increased costs; More difficult to implement;
Biometric Authentication	Greater authenticity; Accessibility; Safer; Simplicity and convenience for the user;	Bias and imprecision; False positive; Permanent digital tracking and records;

Table 2.1: Authentication methods - Advantages and Disadvantages

With this, only one authentication method in the system (single-factor authentication) may not offer the desired security, that is, to offset the disadvantages of the different methods, a 2nd-factor authentication and Multi-Factor Authentication were introduced. As the name implies, single-factor authentication uses only one authentication method, 2nd-factor authentication, two authentication methods are used, and, finally, multi-factor authentication is used for at least three authentication methods.

Regarding authorisation, the associated methods validate the roles, permissions and privileges associated with a user in question. It is performed after the successful authentication, and its main objective is to control the actions that the user can act before the system. For example, it would make no sense for a user to access another user's information and be able to edit that information.

In short, we can identify the importance of these features in any system, especially when there is a need for control from the perspective: who can see or access what and how?; We can also, given the above descriptions, resume that authentication answers the question "are you the person you claim to be?" and authorisation "are you allowed to do this action?"

2.3 Virtual Voice Assistants

The interaction between humans and devices through the dialogue can be a simple definition to voice assistants. Once activated, that is, the user gives the intention to use the voice assistant, the assistant software records the user's voice and sends it to the responsible system that processes and interprets it as a command. Depending on this same command, the system will provide the essential information as needed by the user. Each voice assistant, depending on its context and development, may have unique functionalities. However, some of the most common features in an assistant can be sending or reading text messages, making calls, launching applications, controlling Internet-of-Things-enabled devices (thermostats, lights and alarms), and answering questions like "What day is it today?" among many others [27].

Four essential modules support an intelligent voice assistant: Wake Word, Speech To Text (STT), Natural Language Processing (NLP) and Texto To Speech (TTS). Wake Word and Speech To Text are speech recognition system-based [50]. Usually, the systems are previously configured with a set of words spoken to the system to activate it, that is, a Wake Word that represents the part of the system responsible for identifying the user's intention to place a request to the system. After that, the user makes his request, and it is made up of the voice, sent to the Speech To Text system, which, as the name implies, his goal is to transform the set of words spoken to text. However, for the system to recognise the user's intention, it is necessary to use Natural Language Processing, that is, to make

the user's text into an object capable of being perceived by the system so that it can process the correct answer for the user. Finally, having the answer, Texto To Speech is used to notify the user, that is, transform the solution that is typically given into text to voice. For a better understanding of the previous explanation, Figure 2.6 is presented.

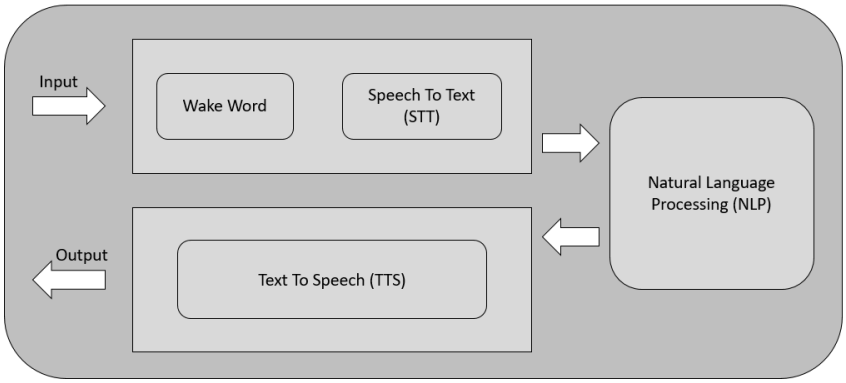


Figure 2.6: Generic Voice Assistant Workflow

Worldwide known companies like Google, Samsung, Amazon, Apple and Microsoft have developed and invested in their voice assistants. Some of these assistants also won their own devices like Google's Google Home, Amazon's Alexa and Apple's HomePod. Moreover, these assistants offer an extensive integration with other devices of these Headphones, SmartPhones, Cars, Some household appliances, SmartWhatch, SmartTV's among many more. Unfortunately, these voice assistants are not open source which does not allow access to their source code. However, given the evolution of technologies, some open source solutions have emerged. In order to list the main open-source virtual voice assistant and compare them, there is a comparison table. First, the evaluation parameters and a brief description are listed.

- Documentation - This point aims to evaluate the documentation that the project offers. This makes its understanding and subsequent development easier. It will be rated at four levels (**Fair**, **Normal**, **Good**, and **Very Good**).
- Support - It concerns the support that the team responsible for the project's development offers when there are some difficulties concerning possible bugs, integration's and/or development. It is also rated in levels which are **Weak**, **Normal** and **Good**.
- Community - Community is an important point. It concerns how much the project is being used by other external teams and the possible existence of future contributions. Rating will consist of **Low**, **Normal** and **High**.
- Product - This refers to projects that, despite being open source, already a developed and stable product can be purchased. It will only be mentioned whether or not there are products, and it intends to show extra credibility and future continuity of the project's development.
- Extensibility - This is one of the most important topics as it concerns when the project is prepared to integrate different technologies in the project. This component is evaluated in three levels (**Fair**, **Normal** and **Very Good**)
- Integration - The capacity to integrate the project into different environments. It is rated **Low**, **Normal** and **High**.

- Observations - Some observations will be made that distinguish the project from the others. Here a description is used.

With this in mind, Table 2.2 shows the comparison between the different assistants.

Voice Assistant	Documentation	Support	Community	Product	Extensibility	Integration	Observation
Mycroft AI [1, 2]	Very Good	Good	High	Yes	Very Good	High	Have partnerships with other development teams to improve some parts of the voice assistant relative to AI components.
Leon [22, 45]	Very Good	Good	High	No	Normal	Normal	None
Kalliope [35, 36]	Good	Good	High	No	Very Good	High	Specifically developed for home automation.
Stephanie [24, 25]	Normal	Fair	Normal	No	Normal	Normal	None
Open Assistant [4]	-	Good	Normal	No	Fair	Normal	It lacks of developer documentation.

Table 2.2: Voice Assistant Comparison

In conclusion, through Table 2.2 we observe that the assistants that stand out the most are Mycroft AI and Kalliope. However, given that Kalliope is more dedicated to house automation and this project is inserted in the industrial environment, we were able to conclude that, among those listed, Mycroft AI is the one that stands out the most.

2.4 Speaker Recognition System

Due to technological advances, voice recognition has gained some prominence in his research, where he gained several and different approaches. A Speaker Recognition System (SRS) for user authentication can be distinguished into two systems [60]: Speaker Identification System (SIS) and Speaker Verification System (SVS). Theoretically, its final purpose is the user's recognition; however, their approaches and how the system recognises the user are different. The main difference of the Speaker Identification System concerning a Speaker Verification System is that one knows which user is undergoing the recognition of his identity while the other does not have this information. As the name indicates, the Speaker Identification System tries to identify the user (does not know who is speaking). The Speaker Verification System checks whether the user is who he says he is (knows who is speaking). Figure 2.7 presents a practical example.

Figure 2.7: Speaker Identification vs Speaker Verification

However, the Speaker Identification System has characteristics that influence its approach in recognising users. First, from the point of view of content, the system can be characterised as text-dependent or text-independent. The text-independent system allows the user to recognise the user, depending on the system's implementation's logic, with any speech he may offer to the system. Regarding the text-dependent system, user recognition is already with predefined speeches. These text-dependent systems can be static or dynamic. Text-dependent static only recognises a determined speech, and the dynamic, the speech can be sent by the system for the user to speak.

For the user's identification, there is a need for the user to be already registered in the system database. Another characteristic of this system is whether or not it allows identifying users who are not registered in the system database. In other words, a system that does accept unregistered users is called a Closed-set system. A system that allows unregistered users is called an Open-set System. Another essential aspect to mention is that in the Speaker Verification System, as they only verify the user's identification, the system compares with a speaker model, that is, with the data from the database that the user is referring to be. This means that it is a 1 to 1 relationship. Since the Speaker Identification System does not know which user is trying to identify, it will be necessary to compare all the different speaker models present in the database. Given this, it means that it is a 1 to N relationship.

In short, despite the existence of two types of Speaker Recognition System, typically, the main steps can be identified as [3]:

- Receive a voice sample;
- Voice signal treatment, normalisation and feature extraction;
- Quality checks (they may reject the sample or inappropriate signals for comparison requiring the submission of additional samples);
- Comparison with a specific or all database models to determine the degree of similarity;
- Decision making regarding identity;

Considering these points, the final system capable of correctly recognising users can be divided into Front-end Processing and Speaker Modeling Techniques. In Figure 2.8, the typical architecture of a Speaker Recognition System is presented.

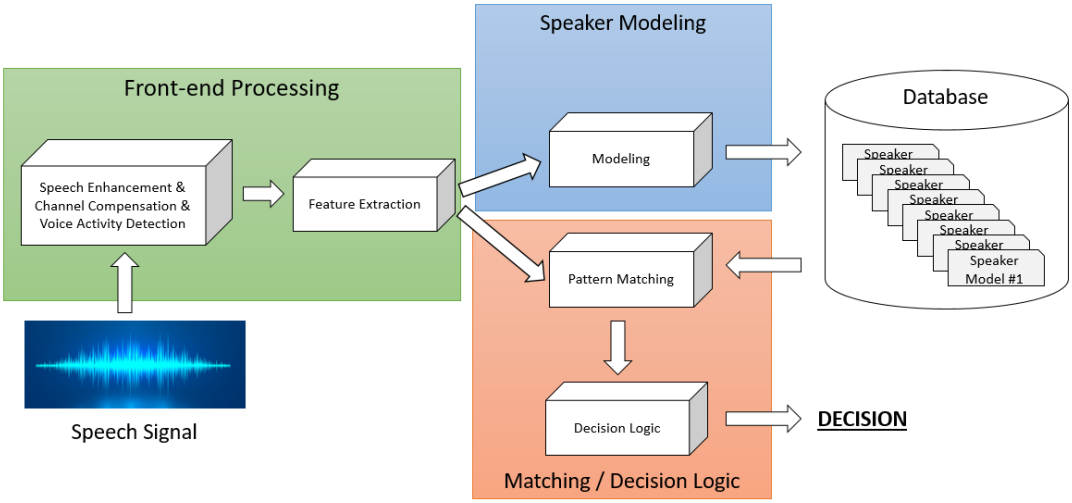


Figure 2.8: Speaker Recognition System [61]

Regarding Speaker Modeling Techniques, generally speaking, these methods can be divided into four categories [3]: Systems based on Vector Quantization (VQ), systems based on Gaussian Mixture Model (GMM), systems based on factor analysis, and, more recently, systems based on Neural Network (NN).

Concerning Front-End Processing, the voice signal sent to the Speaker Recognition System sometimes contains some unwanted background noise and parts with no voice activity. In general [39], the main factors causing signal distortion are the user's condition, software, and hardware. In addition, channel effects are also a major cause of errors in voice recognition systems. Having this in mind, the main techniques consists mainly of voice activity detection, speech enhancement, channel compensation, and feature extraction.

2.4.1 Front-end Processing

Since the user's recognition problem requires recording their voice, several factors significantly degrade the voice signal. Because of that, this component of the system is crucial for its excellent performance. As mentioned before, the main techniques are called: voice activity detection, speech enhancement, channel compensation, and feature extraction.

The goal of voice activity detection [9], is to determine whether a signal contains speech or not. In order to do speech enhancement, due to the loss of signal quality at the recording time, some approaches [15] such as Periodic Noise Removal, Wide Band Noise Removal, and Interfering Speech aim to remove the different types of noise and improve the quality of the recorded voice emerge. The periodic noise can be removed using stationary filters, adaptive filters, or transform domain filters. Wideband noise can be removed using the Spectral Subtraction method (SS) and adaptive cancellation, and for interfering speech, a comb filter or transform domain technique can be used. Channel Compensation [39] which in turn, by default, includes Feature Domain Compensation, Model Domain Compensation and Score Domain Compensation. Feature Domain Compensation aims to remove channel mismatch when feature vectors are being extracted. Most standard techniques are cepstral mean subtraction, RASTA filtering, cepstral subtraction and MAP (Maximum A Posterior Probability). Model Domain compensation modifies the models to minimise channel incompatibility. In this case, we have speaker model synthesis, where you learn how the model parameters change between different channels and apply a transformation to synthe-

sise models under invisible registration conditions. Finally, scoring Domain compensation attempts to remove scales from the model and changes caused by channel mismatch. For this can be used, the techniques are Hnorm and Tnorm.

Already introduced in section 2.1.1, one of the most used feature extraction techniques is called Mel-Frequency Cepstral coefficients (MFCC) and given that it is possible to categorise the methods for extracting features as Mel-Frequency Cepstral coefficients based and non Mel-Frequency Cepstral coefficients based [11]. In summary, a description of the approaches proposed and studied by some researchers are presented in Table 2.3.

Features	**References**
Mel-Frequency Cepstral Coefficients Based	
Combination of Mel-Frequency Cepstral Coefficients with Perceptual Linear Predictive Coefficients	[10]
Mel-Frequency Cepstral Coefficients combined with Residual Phase Cepstrum Coedfficients, Glottal Flow Cepstrum Coefficients and Teager Phase Cepstrum Coefficients	[17]
Mel-Frequency Cepstral Coefficients with Inverted Mel-Frequency Cepstral Coefficients	[14, 61]
Mel-Frequency Cepstral Coefficients incorporated into the histogram transformation features	[47]
Non Mel-Frequency Cepstral Coefficients Based	
Auditoy-Based Time-Frequencey Transform	[48]
Mean Hilbert Envelope Coefficient	[20, 58]
Power-Normalised Cepstral Coefficient	[51, 58]
Using Fisher Vector	[31]
Convolutional Neural Network	[11, 28]

Table 2.3: Feature Extraction : MFCC and Non-MFCC-based

2.5 Related Work/Systems

Known in generic terms the architecture of the Speaker Recognition System, identified its biggest barriers to its development, as well as the different steps and consequently different techniques related to each step, this section describes the different systems, that is, what type of techniques used for Front-end Processing and Speaker Modeling and the results obtained. First, a Table 2.4 of the techniques used is presented. Finally, for each work presented, the results obtained are described.

Title	**Front-End Processing**	**Speaker Modeling**
Speech Recognition and Verification Using MFCC & VQ [55]	MFCC using Triangular Shaped Bins	VQ with K-Means and Euclidean Distance for Clustering
VQ Approach for Speaker Recognition using MFCC and Inverse MFCC [61]	MFCC and/or Inverse MFCC using Guassian Shaped Fileter	VQ
Improved Text-Independent Speaker Identification using Fused MFCC & Inverse MFCC Feature Sets based on Guassian Filter [14]	MFCC and/or Inverse MFCC using Guassian Shaped Fileter and Triangular Shaped Bins	GMM
Speaker Model Clustering for Efficient Speaker Identification in Lager Population Applications [41]	MFCC	GMM based using K-Means for Clustering

Table 2.4: Related Work : Techniques

The use of an approach using Convolutional Neural network (CNN) through spectrogram images of short speech phrases is also explored by the authors, in [11]. Each signal wave sample is transformed into a spectrogram and used as a grayscale input image for the network. This approach is compared with the traditional MFCC feature extraction classified by Support Vector Machines (SVM) and the Convolutional Neural network that uses the signal image as an input. Its results show a much higher performance when using its approach than the others. In [3], the authors also discuss the use of Deep Neural Network (DNN) in speaker verification. The authors explain DNN is generally used to extract the specific features of speak and the baseline system with deep network research for speaker verification is the GMM-UBM system. More specifically, we have GMM-UBM and i-Vector based system where the i-vector approach shows considerable improvements

in speaker verification and consists of three sequential steps. First, the extraction of information/statistics through data, then calculate the i-vectors and apply the Probabilistic Linear Discriminant Analysis. Features Extracted from DNN Used by GMM-UBM or i-Vector Based System where DNN's are used to extract frame by frame speaker information and calculate its utterance-level information. Finally, the authors perform a combination of different architectures based on Deep Neural Network applied to speaker verification.

Regarding the results/tests carried out in order to clarify them better, two tables are presented. Table 2.5 presents the results of the works described in Table 2.4.

Title	Results & Datasets
Speech Recognition and Verification Using MFCC & VQ	The authors obtained an error rate of 13 % for a system that considers users' verification through commands such as "Hello".
VQ Approach for Speaker Recognition using MFCC and Inverse MFCC	Results show that the use of this approach to extract features, both individually and in combination, improves the speaker identification performance from very short samples, obtaining an efficiency of 100% for a speech of 6 seconds, 98.57 % for 3sec, 98.49 % for 2sec and 94.28 % for 1sec. For YOHO dataset the results were between 94-95% for system using Triangular Shape Bins based on MFCC and Inverse MFCC. Combining MFCC and Inverse MFCC had a performance of 96-97%.
Improved Text-Independent Speaker Identification using Fused MFCC & IMFCC Feature Sets based on Guassian Filter	Regarding the approach using Gaussian Shaper Filter, the results were 94-96% (MFCC and Inverse MFCC) and 96-97% (MFCC and Inverse MFCC combined).For the POLYCOST database, the results using Triangular Shaped Bins based on MFCC and Inverse MFCC as well as MFCC and Inverted MFCC combined were 76-77% and 81%, respectively. Using Gaussian Shaped Bins they had 76%-80% (MFCC and Inverse MFCC) and 82 (MFCC and Inverse MFCC combined).
Speaker Model Clustering for Efficient Speaker Identification in Lager Population Applications	For TIMIT/NTIMIT dataset 24s training and 6s testing signals were used and for NIST 2002 90 training and 30s testing signals were used. The identification system had an accuracy of 99.68%, 69.37% and 89.39% for the TIMIT,NTIMIT and NIST 2002 Courpus datasets respectively.

Table 2.5: Related Work : Results and Datasets

Finally, the related works are presented in a summarised way, as well as the respective results using deep neural networks through Table 2.6 [30]. Additionally, it also shows their architectures and datasets.

Title	Type of System	Input Features	DNN Type	Score Function	Baseline System	Datasets	Score (%ERR)
[26]	Text-dependent	40 log Mel-filter bank coefficients	7-layered, full-connected	PLDA	UBM/i-vector	NIST SRE'12 Noisy narrowband	1.39
[62]	Text-independent	39 dimensions PLP	7-layered RBM	Cosine Distance	GMM-UBM	NIST SRE'05-06	0.88
[38]	Text-independent	60 dimensions MFCCs	5-layered	Probabilistic Linear Discriminant Analysis	UBM/i-vector	NIST SRE'12 Switchboard I,II,III	1.58
[66]	Text-dependent	39 dimensions PLP	4-layered, fully-connected	Cosine Distance	UBM/i-vector	Self-Created	1.21
[40]	Text-dependent	20 dimensions MFCCs	4-layered, fully-connected	Probabilistic Linear Discriminant Analysis	UBM/i-vector, GMM-DTW	RSR 2015	0.2
[37]	Text-dependent	40 dimensions MFCCs	7-layered, multi-splice time delay	GPLDA	GMM-UBM	NIST SRE'10	7.2
[42]	Text-independent	Phone-blind & Phone-aware 40 dimensions d-vectors	7-layered, time-delay	PLDA	UBM/i-vector	Fisher dataset, CSLT-CUDGT2014	8.37
[62]	Text-independent	20 dimensions MFCCs	4-layered, temporal pooling	PLDA	UBM/i-vector	US English telephonic speech	5.3

Table 2.6: Related Work : DNN Results and Datasets

2.6 Conclusion

Considering the main objective of this work, voice authentication in a virtual assistant, we were able to identify how this type of system works and its main characteristics. Furthermore, the main barriers to implementing this system have also been identified and how they can be overcome. Thus, the typical architecture of a speaker recognition system and the preprocessing, processing and modelling problems for the construction of this speaker recognition system are identified. We can conclude that one of the most complex challenges is cleaning the signal transmitted by the user. It is then noted the importance of development and investment in biometric systems that are differentiated by the fact that they use unique characteristics of the user. On the other hand, its most significant challenge is developing approaches that can extract/identify these characteristics with a very low margin of error.

This page is intentionally left blank.

Chapter 3

Methodology and Planning

In this chapter, the main topics will be the description of the development methodology and the planning. Regarding planning, an analysis of what was planned with the work performed is carried out.

3.1 Methodology

The methodology used for this work is based on the logic of the traditional cascade model, a methodology used for software development. However, given that this project requires the development of machine learning approach/techniques to solve the problem in question, the methodology can be presented by the Figure 3.1.

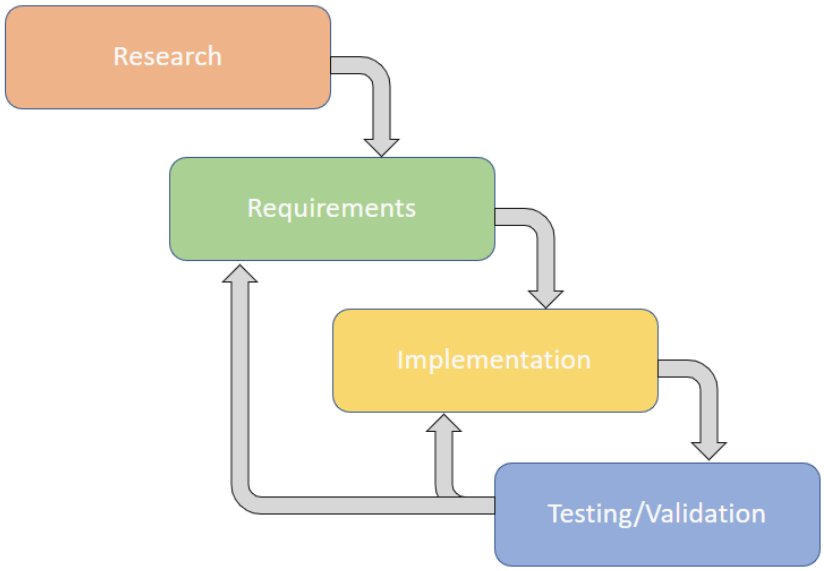


Figure 3.1: Methodology

As it is possible to visualise in the previous figure, given the nature of this project to be of a more investigative component, after the first version of this system has been developed, it is possible to return to the previous steps to improve this same system. This same project, having the objectives already identified, the research is carried out, and as a result, it is implemented, tested and validated.

3.2 Work Planning

In this section, an analysis is made of the relationship between planned vs performed. Considering the previous planning, the Table 3.1 presents the relation between that planning and semester's work. Additionally, the weight of each task is shown as a percentage. In summary, planning carried out was divided into 3 phases. First, the implementation of the system which includes speaker voice recognition and their integration in the virtual voice assistant, the respective tests and validation, and finally, the writing of the report.

PLANNED	REAL	START	DUE	PERCENTAGE
Implementation	Report (Correction)	1 February	19 February	11,5%
Implementation	Implentation (Data Processing)	22 February	19 March	15,4%
Implementation	Implementation (Modeling)	22 Mach	30 April	19,2%
Implementation	Implementation (Pre Processing)	3 May	14 May	7,7%
Testing/Validation	Modeling Testing	17 May	28 May	7,7%
Report Writting	Implementation (Server + Database)	31 May	11 June	7,7%
Report Writting	Integration (Server + Assistent)	14 June	25 June	7,7%
	System Testing	28 June	2 July	3,8%
	Report Writting	5 July	30 July	23,3%

Table 3.1: Planning vs Real

Looking at the table, we can see that it was initiated by an unplanned task, the correction of the report, and some additional research. After that, and since the integration resulted in the development of a server, database, and an extension to the Mycroft AI framework, it occupied the part planned for writing the report. In short, these were the two main differences between planning and work carried out.

This page is intentionally left blank.

Chapter 4

System Overview

This chapter is where the solution is detailed, from its design to its implementation. First, a survey of the characteristics and respective descriptions is carried out. Then, the system's workflow is detailed, from the data collection to the speaker recognition model. Finally, the architectures of the selected virtual assistant, speaker recognition system, integration, and deployment are described.

4.1 System Characteristics and Design

For the successful development of the system, this section presents and details its design and features. Thus, this system is designed around three components:

- The system must be able to save the biometric (voice) signals of the user.
- The system must recognise the user's identity by their voice.
- The system must manage user authorisation levels regarding their requests.

These same ones can be reflected and easily described in the following use cases presented in the use case diagram in the Figure 4.1.

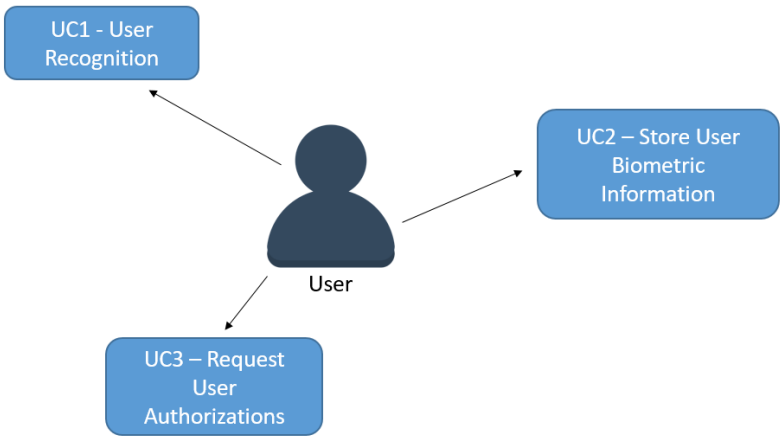


Figure 4.1: Use Cases Diagram

The three use cases are briefly described. Additionally, to provide more detail used a table where the main actor, objective, pre and post conditions, main scenario and possible secondary scenarios are identified. Use Case 1 is about user recognition. Through a command characterised by the voice signal, the user asks the system to be identified. Table 4.1 gives a more detailed description of this Use Case 1.

Main actor	User
Goal	Be identified by the system
Pre-Conditions	Biometric information (voice signal) stored in the system Internet connection
Post-Conditions	-
Main scenario	1. User activates system using the wake word 2. The system emits a sound that is translated by the start of recording of the request 3. The user places his request through a command 4. The system emits a sound signifying the end of the recording, processes your command and identifies the user through his full name
Alternative scenarios	4. System cannot understand command 4.1. The system notifies the user of the error and asks to perform the command again 4. The system cannot identify the user correctly 4.1. The system notifies the user that it was not possible to identify him and asks again to try again

Table 4.1: Complete Description of Use Case 1

In Use Case 2, the user, through a command, saves your provided biometric information (voice signal) for future identification. A complete description of Use Case 2 is shown in Table 4.2.

Main actor	User
Goal	Store biometric information from specific user
Pre-Conditions	Internet connection
Post-Conditions	Biometric information (voice signal) stored in the system
Main scenario	1. User activates system using the wake word 2. The system emits a sound that is translated by the start of recording of the request 3. The user places his request through a command 4. The system emits a sound signifying the end of the recording, processes the command and asks the user some recordings 5. The user makes some recordings 6. The system validates the recordings and notifies of the success of the operation
Alternative scenarios	4. System cannot understand command 4.1. The system notifies the user of the error and asks to perform the command again 6. The system identifies problem in recording 6.1. The system notifies the user of the error and asks to perform the command again

Table 4.2: Complete Description of Use Case 2

Use Case 3, the user asks the system, using a command, of authorisations he has on it. Table 4.3 show the full description of this Use Case 3.

Main actor	User
Goal	Store biometric information from specific user
Pre-Conditions	Internet connection
Post-Conditions	Biometric information (voice signal) stored in the system
Main scenario	1. User activates system using the wake word 2. The system emits a sound that is translated by the start of recording of the request 3. The user places his request through a command 4. The system emits a sound signifying the end of the recording, processes the command and list the different types of authorisations that have
Alternative scenarios	4. System cannot understand command 4.1. The system notifies the user of the error and asks to perform the command again 4. The system cannot identify the user correctly 4.1. The system notifies the user that it was not possible to identify him and asks again to try again

Table 4.3: Complete Description of Use Case 3

4.2 Machine Learning Workflow

This section presents, in a detailed way, the data flow, implementation, and consequently the used techniques of the speaker recognition system. It starts by describing the workflow as well the implementation pipeline and the pre-processing techniques. It follows the description of the processing carried out (feature extraction) and finishes by presenting the technologies used. This system is categorised into two and, consequently, divided into two very common workflows called the training and testing phases. Its main difference is in the last step, where in training, the data is sent to the modelling algorithms, and the test phase sends it to an already trained model for its data classification. Figure 4.2 represents these same two phases.

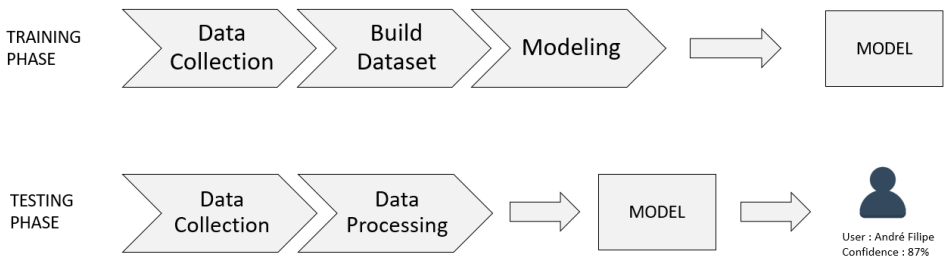


Figure 4.2: Speaker Recognition System : Training and Testing Phase

As we can see in Figure 4.2, the training phase ends with a trained model and the test phase with results relative to a model. It is also important to mention that the Data Processing phase is a sub-module of the Build Dataset, that is, Build Dataset also includes Data Processing. Therefore, to better describe both phases in detail and show their common parts, Figure 4.3, it is possible to observe the workflow at a much more detailed level.

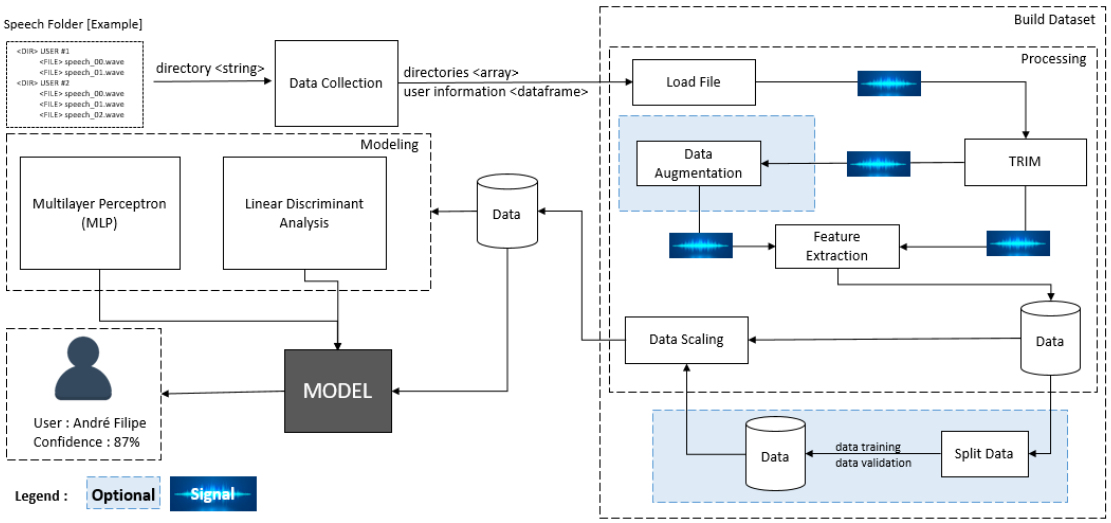


Figure 4.3: Speaker Recognition System : Workflow

1. **Data Collection** – This is the initial process where information from users is collected, namely the amount of speeches they have as well as a list corresponding to the speeches' directorates.

2. **Processing Phase -** After Loading the speech file, four approaches are applied:

   (a) Trimming

   (b) Optionally, and if necessary, Data Augmentation [19, 46]

   (c) Feature Extraction

   (d) Depending on the used algorithm, that is, Multiplayer Perception or Linear Discriminant Analysis, optionally, we can split the Dataset in two: Data Training and Data Validation

3. **Modeling or Testing -** After that, we can follow two "paths", one for the modelling where the model is trained for future identification of users or for the pretrained model where the data will be classified that is, training and testing phases, respectively.

After describing the data flow and identifying the algorithms/approaches used, these descriptions are described in more detail. As far as data construction (processing) is concerned, we have Trim, Data Augmentation, Feature Extraction, Data Scaling, and Split Data. For modelling neural networks (Multilayer Perceptron) and Linear Discriminant Analysis is used.

Trim is an approach that aims to cut silent signal spaces, that is, without any speech. This will have an impact on the total speech time. For example, Figure 1 is presented where we can visualise the same speech signal in its first phase (Normal) and after applying this technique (Trimmed). To identify the silence zones, a threshold is defined, in which case defined that below 20 decibels, it is considered a silent zone (without speech).

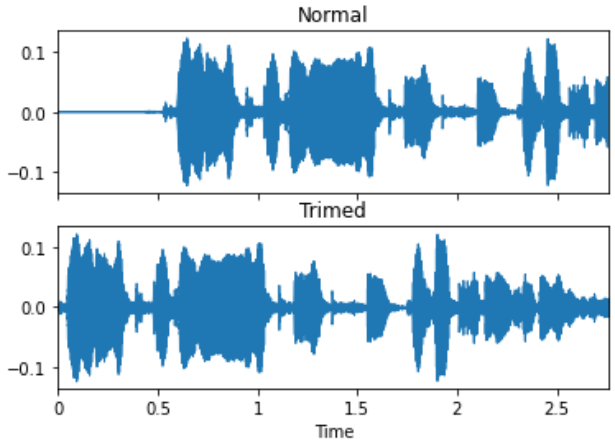


Figure 4.4: Example: Normal Speech vs Trimmed Speech

Concerning this example show in Figure 4.4, was removed 1.1 seconds of their total duration. As a result, the normal speech was 3.86 seconds, reducing it by 2.76 seconds through processing. A common problem is the lack of data for model training. In other words, when working with small datasets, there are techniques (they vary according to the type of data) that artificially, through the original data, can create new data. This approach is called data augmentation, and hence it is an optional process that is only used depending on the need for such data augmentation to existing. Knowing that we are dealing with voice signals, some more common approaches are: injecting noise, changing the pitch and changing the speed. However, given the context of the problem, changing the pitch and velocity would change the characteristics of the voice, and therefore the most appropriate and used technique is noise injection. In this way, having the original voice signal, its speech time is calculated. After that, the noise is created and integrated into the original signal, thus duplicating the data. The Figure 4.5 shows this transformation.

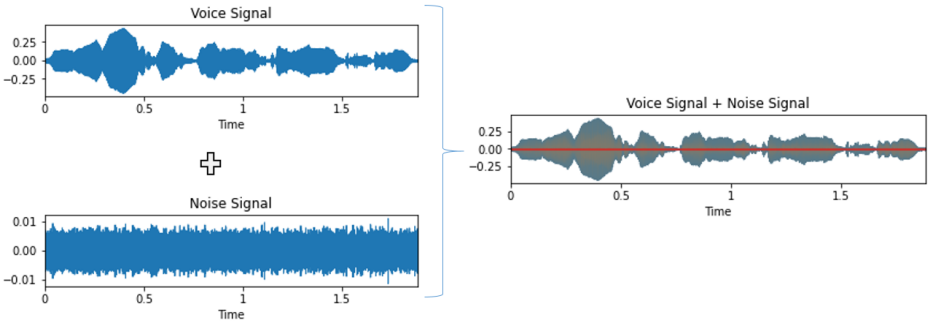


Figure 4.5: Example: Data Augmentation

In short, background noise is added from the original voice signal, such as visible in red in the signal "Voice Signal + Noise Signal". Finally, regarding data scaling, already explained in 2 Section 2.1.1, uses the Standard Scaling approach. Transforms the data so that its distribution will have a mean value of zero and a standard deviation of one.

4.2.1 Feature Extraction

This process, explained in Chapter 2 Section 2.1.1, is one of the most important steps and allows extracting the unique characteristics of the voice signal. There are two types of features [12]. Temporal features (time-domain features) are simple to extract and have easy

physical interpretation (energy of signal, zero-crossing rate, maximum amplitude, minimum energy). Spectral features (frequency-based features) are obtained by converting the time-based signal into the frequency domain using the Fourier Transform, like fundamental frequency, frequency components, spectral centroid, spectral flux, spectral density, spectral roll-off. These features can be used to identify the notes, pitch, rhythm, and melody. In short, the extracted features were as follows [32, 65].

Chroma features relate to the twelve different pitch classes are a powerful tool for analysing music whose pitches can be meaningfully categorised (often into twelve categories) and whose tuning approximates to the equal-tempered scale. One main property of chroma features is that they capture harmonic and melodic characteristics of music while being robust to changes in timbre and instrumentation. Two types of features use STFT to generate the spectrogram and then project that spectrum down to a single-octave (chroma) representation. The others are similar, however, they use CQT instead of STFT.

MFCC any sound generated by humans is determined by the shape of their vocal tract (including the tongue, teeth). If this shape can be determined correctly, any sound produced can be accurately represented. Thus, the envelope of the time power spectrum of the speech signal is representative of the vocal tract and MFCC, which is nothing but the coefficients that make up the Mel-frequency cepstrum, accurately represents this envelope.

Computing the **Roor-Mean-Square (RMS)** value from a spectrogram will accurately represent energy over time because its frames can be windowed.

Zero Crossing Rate measures the number of times in a given time interval/frame that the amplitude of the speech signals passes through a value of zero. The simplest method to distinguish between voiced and unvoiced speech is to analyse the zero crossing rate. A large number of zero crossings implies that there is no dominant low-frequency oscillation.

Spectral Centroid, each frame of a magnitude spectrogram is normalised and treated as a distribution over frequency bins, from which the mean (centroid) is extracted per frame.

Spectral Contrast, each frame of a spectrogram is divided into sub-bands. For each sub-band, the energy contrast is estimated by comparing the mean energy in the top quantile (peak energy) to the bottom quantile (valley energy). Thus, high contrast values generally correspond to clear, narrow-band signals, while low contrast values correspond to broad-band noise.

Spectral Rolloff frequency is defined for each frame as the center frequency for a spectrogram bin such that at least 85% (by default) of the energy of the spectrum in this frame is contained in this bin and the bins below.

4.2.2 Technologies

To develop the workflow detailed above, the language used is Python, however, the most outstanding libraries are Librosa, Scikit-learn, and Keras. Librosa [43, 49] is a Python package for audio and music signal processing. It provides the building blocks necessary to create music information retrieval systems. In general, librosa functions tend to expose all relevant parameters to the caller. While this offers great flexibility to expert users, it can be overwhelming to novice users who need a consistent interface to process audio files. Therefore, a set of general conventions and standardised default parameters values shared across many functions were defined to satisfy both needs. This library was used to process the audio signal. It was mostly used to read audio files, perform signal processing, and extract features to train/test AI models for user identification. Scikit-learn [56] is an open-

source machine learning library that supports supervised and unsupervised learning. It also provides various tools for model fitting, data pre-processing, modelling and evaluation. In this case, it is used both for data processing and machine learning. Finally, Keras [16] is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. This library was used mainly for the modelling of neural networks.

4.3 System Architecture

This section discusses the architectures, that is, the voice assistant (Mycroft AI), the Speaker Identification System (SIS) (server), and these two systems integration. Finally, once the entire system is known, the sequence diagrams referring to the use cases illustrated in Figure 4.1 are introduced.

4.3.1 Virtual Voice Assistant - Mycroft AI

Mycroft AI [1, 2] offers freedom and control over the assistant, making it easy to integrate this voice system in specific contexts. It also provides a level of software with a wide range of hardware compatibility. It is possible to integrate Mycroft either on a Raspberry Pi or an Android device or even on a Linux system. The architecture of this system consists of four main parts [52]: Enclosure, Voice, Skills and Service/Message Bus.

Service/Message Bus is responsible for creating a WebSocket to establish communication with the other parts of the system and ensure communication between them. In **Skills**, a "skill" can be seen as system functionality. These are loaded into the system only when necessary and independently, with their process. This approach was able to bring some advantages, such as the lower use of memory and the conflict of bookstores, that is, the existence of skills with the same bookstores but different versions. This part of the architecture has the Natural Language Processing method. **Voice** is where the connection to the microphone and speakers is established. It is also where the Wake Word, Speech To Text and Text To Speech methods are found. **Enclosure** is the component that interacts with different parts of the system's hardware or software.

In Figure 4.6 it is possible to view the architecture previously explained.

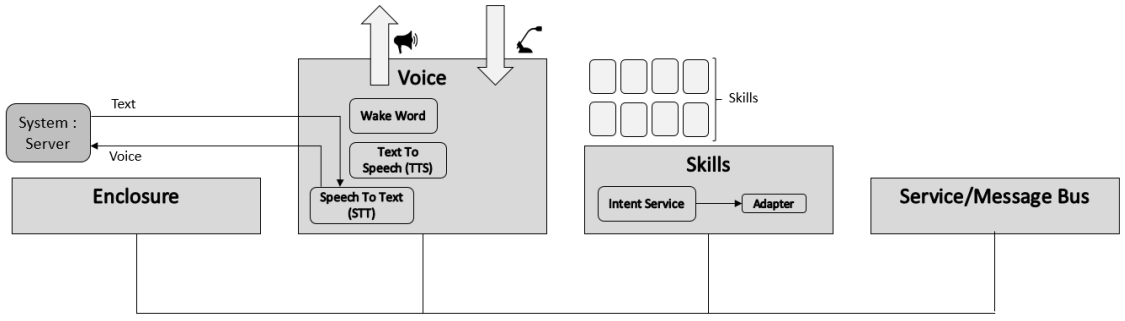


Figure 4.6: Mycroft Architecture

4.3.2 Speaker Identification System

This system is an API REST server based on MVC architecture that stores the machine learning model to identify users. API REST means that this server has a web standards-based architecture that uses the HTTP protocol for data communication. Model-Controller-View (MVC) is a software architecture standard that separates applications into three layers. Each layer is responsible for performing certain actions, thus making the application much more organised. In Figure 4.7, this architecture is presented.

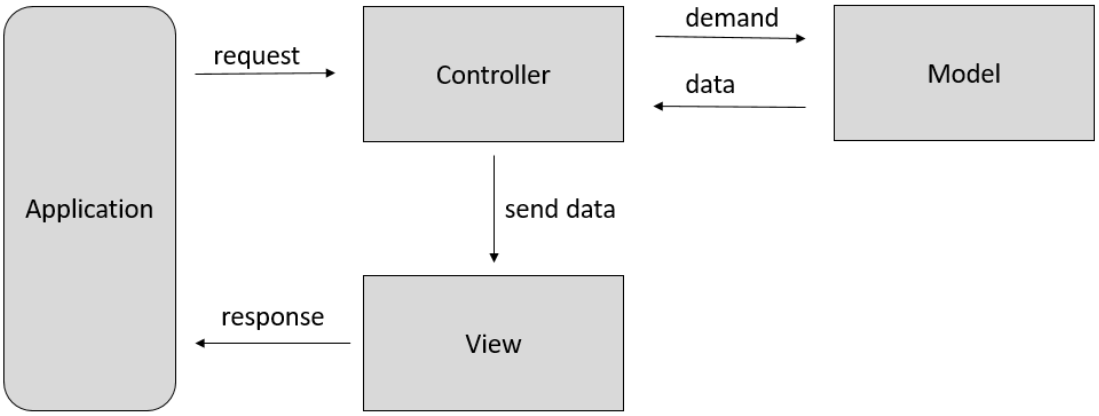


Figure 4.7: Model-View-Controller Architecture

Each layer is responsible for the following actions:

- **Model:** contains the application logic, is responsible for validations, business rules and database operations
- **View:** manages the interaction with the user, that is, formulates the response to the user's request.
- **Controller:** works as an intermediary between Model and View. Defines the application's behaviour and controls the sending of data between the View and Model.

With this in mind, Figure 4.8 represents the architecture of the Speaker Identification System (SIS).

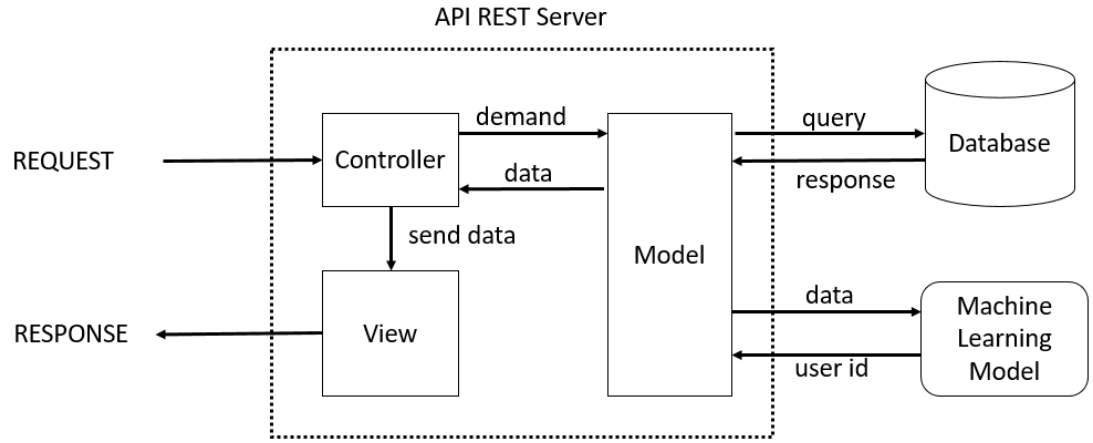


Figure 4.8: Speaker Recognition System Server Architecture

As we can see, the controller is the intermediary of the model and view. Model is responsible for processing the data (trim, feature extraction and scaling), requesting recognition from the machine learning model, and querying the database. The view that, after receiving the data, formulates the response to send to the assistant. Regarding the database, it only addresses the need to identify users and manage their permissions. Given this, three tables are created (users, roles and skills). Figure 4.9 shows their tables and their relationships. As their relations are N to N, two more tables are created (users_roles and roles_skills).

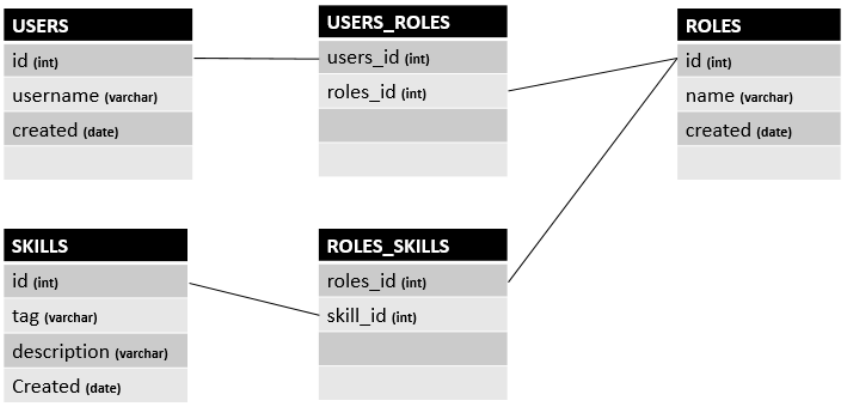


Figure 4.9: Database Architecture

4.3.3 Integration

Once the two architectures are known, Mycroft AI and Speaker Identification System (SIS), final integration is summed up in creating communication between them. Mycroft uses the server's implemented resources to identify users. Data is sent and received in JSON format. As such, its architecture is illustrated in Figure 4.10.

Figure 4.10: High-Level System Integration Architecture

As a function of detailing the functioning of the integration of this system and considering the use cases identified and illustrated in Figure 4.1, a sequence diagram of the system is presented that demonstrates its functioning in greater detail. However, it is important to point out that, except for Use Case Two, we can generalise the system sequence diagram, that is, the variables that change are: Command, Server Response, Database Response and Mycroft Response. Table 4.4 identifies the different commands and their responses for the remaining use cases.

	COMMAND	MYCROFT_RESPONSE	SERVER_RESPONSE	DATABASE_RESPONSE
UC1	"can you identify me?"	"i think you are {username} "	{"username":"André Filipe", "confiance":92}	username
UC3	"what permissions do i have?"	"you have {roles} permissions"	"roles":[{"tag":"admin", "name":"administrator"}]	username list<roles>

Table 4.4: Use Cases 1 and 3 Variables

The system sequence diagram that illustrates the operation of Use Cases 1 and 3, considering the values in table 4.4, is shown in Figure 4.11.

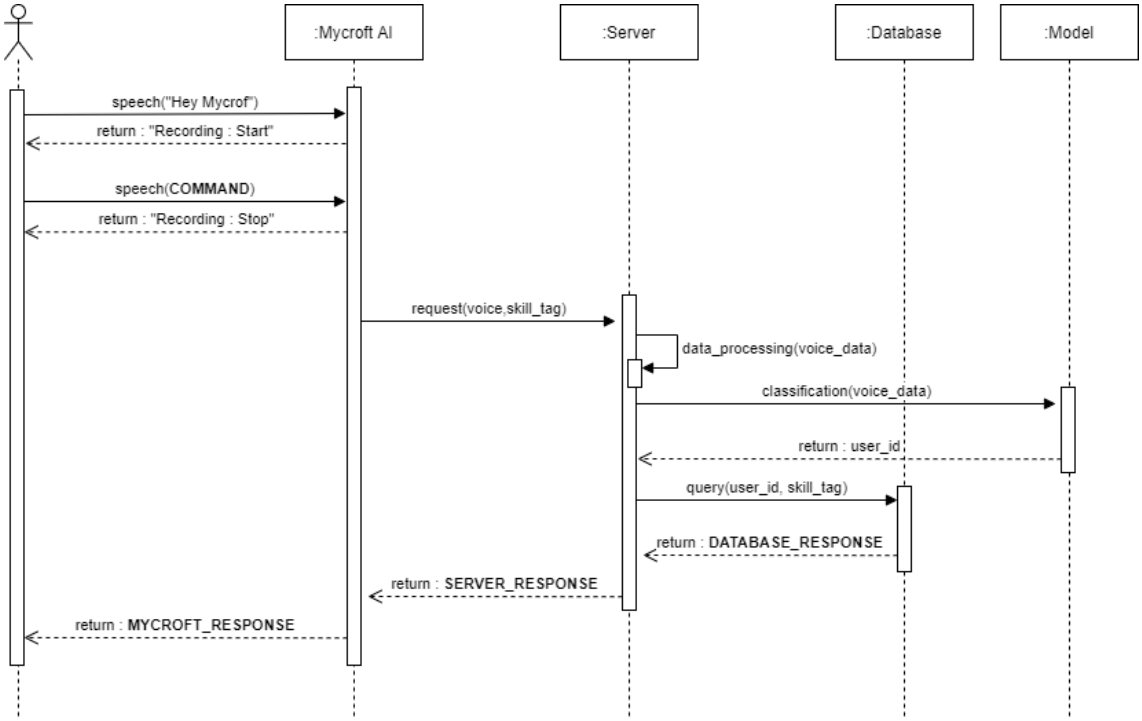


Figure 4.11: Sequence System Diagram : Use Case 1 and 3

By describing this flow, the user activates the virtual assistant using their default wake word, “Hey Mycroft”. After being alerted that the recording of his command started

through a sound, the user communicates the desired command. The command is received on Mycroft and sent to the server via an HTTP request. The server receives the voice data and respective tag that identifies which skill was triggered (user's action on the system). Next, voice data is processed (trim, feature extraction and scaling) and sent to the pre-trained model to identify. Finally, a database query is performed to fetch information about the user and, depending on his action, and his response will match those previously identified in the Table 4.4. Use case 2 is more complex, as its objective is to store your biometric information on the server. Additionally, by using scripts, the server automates the process of creating a new neural network, replacing the old one. This process can be better described in Figure 4.12.

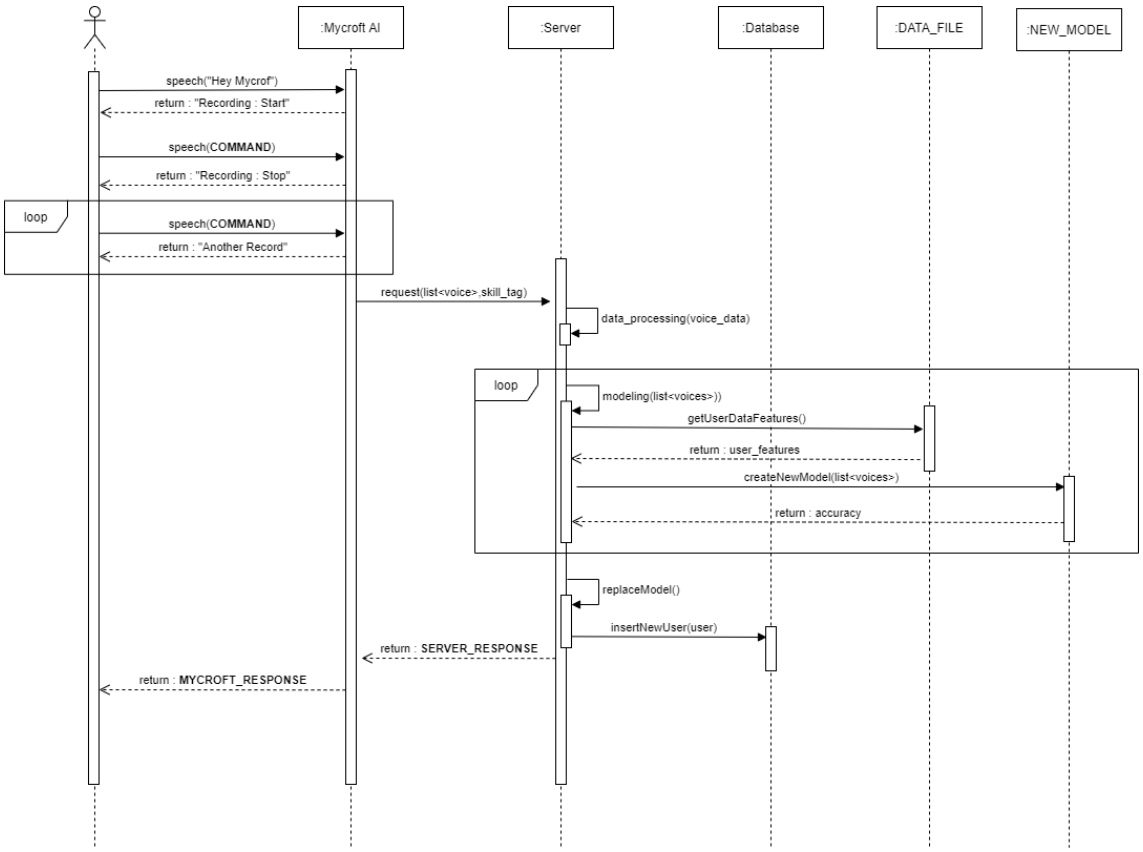


Figure 4.12: Sequence System Diagram : Use Case 2

4.4 Summary

In summary, the design and main features of the system are identified, and to better describe these same features, a use case diagram and respective individual descriptions of each use case are presented. This is followed by an explanation of the workflow for creating user identification models, the front end processing and modelling techniques used, and the possible visualization of the data flow. The principal technologies used are also briefly explained. Finally, the architecture of the chosen virtual voice assistant is explained, the architecture of the server responsible for hosting the model, and finally, it is described how the integration is done, accompanied by sequence diagrams of the system.

This page is intentionally left blank.

Chapter 5

Experimentation

This chapter analyses the final experiments carried out to measure the performance of the algorithms used. It starts with the experimental setup where the properties of the different datasets used, the techniques/architectures and their respective parameters are detailed. Then, the metrics used to measure the performance are defined. After that, the results obtained are presented, always offering a comparative and critical analysis about them. Finally, some conclusions that can be drawn when observing the obtained results are presented.

5.1 Experimental Setup

The section will be elaborated in a more detailed way as conditions of the experiment, its data sets, and the parameters of the used algorithms. Regarding datasets, four different datasets are used: **NOIZEUS** [44], **TIMIT Acoustic-Phonetic Continuous Speech Corpus** [34], **LibrisSpeech ASR** [54] and a created dataset called **SAFC**. In the Table 5.1 we can analyse these same datasets through the following characteristics: Dataset size, Number of speeches made by each user and average speech time.

Name	**Size**	**Number of Speech's**	**Average Time of each Speech**
NOIZEUS [44]	6	5	2,6680s
TIMIT [34]	462	8	2,6726s
LibrisSpeech ARS [54]	156	77 (average)	6,8771s
SAFC	25	5	3,1416s

Table 5.1: Dataset Properties

As we can see in the previous table, their characteristics allow us to study the scalability performance of the two chosen algorithms concerning the growth in the number of users. Additionally, we can also analyse the performance of the approaches taking into account the amount of data per user. The chosen approaches are known as Neural Network - Multilayer Perceptron and Linear Discriminant Analysis (LDA).

The architecture of the Neural Network Multilayer Perceptron (MLP), as well as its parameters, is defined by:

- **Input Layer -** Input is equal to the number of features in the dataset;
- **Five Hidden Layers -**
  - First Hidden Layer
    - Neuros: 1205; Active Function: Tanh; Dropout: 5%
  - Second Hidden Layer
    - Neuros: 1045; Active Function: Tanh; Dropout: 15%
  - Third Hidden Layer
    - Neuros: 823; Active Function: Tanh; Dropout: 25%
  - Fourth Hidden Layer
    - Neuros: 633; Active Function: Tanh; Dropout: 35%
  - Fifth Hidden Layer
    - Neuros: 423; Active Function: Tanh; Dropout: 45%
- **Output Layer -** Its output is equal to the number of users to classify; Active Function: Softmax;

After being built following the architecture-defined abode, the model is compiled using the parameter identifier in Table 5.2.

Name	**Value**
Adam	0,001
Loss function	Sparse Categorical Crossentropy
Epochs	150
Batch Size	512
Callbacks	Early Stop

Table 5.2: Neural Network - Multilayer Perceptron Parameters

As we can see in the previous table, the model is compiled with the optimiser that implements the Adam algorithm with a learning rate of 0.001. Adam optimisation is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments. The loss function used is Sparse Categorical Crossentropy because we always classify more than two users where they are categorised by ID (integers). It is important to note that validation data is used, 150 Epochs and with a Batch Size of 512. Additionally, an early stop callback is implemented. Early stop and dropout are used in order to make the network as widespread as possible, trying to prevent its overfitting. Overfitting means that the training of this network was very good, but when exposed to new data its performance is very low.

Concerning Linear Discriminant Analysis, as this algorithm is simpler about its parameterization, the parameters used to concern the solver in which Eigenvalue decomposition was used with a value of 0.1 Shrinkage.

Thus, the metric chosen to measure the performance of the algorithm is accuracy. This accuracy is given by the amount of correctly identified speeches as a function of the total number of tested speeches. In addition, its performance regarding the use of the data augmentation technique in datasets is also studied.

5.2 Experimental Results

The primary study measures the performance of the different algorithms chosen as a function of the growth in the number of users. Its performance is also analysed with and without the use of data augmentation. Starting with the dataset named NOIZEUS, with six users and five speeches per user, the results are presented in the Table 5.3.

	with Data Augmentation	**without Data Augmentation**
Neural Network (Multilayer Perceptron)	100%	100%
Linear Discriminant Analysis	100%	100%

Table 5.3: NOIZEUS dataset results

In this first phase, we can conclude that both approaches have good performance in a system with a minimal number of users and speeches per user. We also note that using data augmentation is unnecessary. Moving on to the dataset created and composed of 25 users and five speeches per user, Table 5.4 shows the results obtained from selected approaches.

	with Data Augmentation	**without Data Augmentation**
Neural Network (Multilayer Perceptron)	100%	100%
Linear Discriminant Analysis	100%	100%

Table 5.4: SAFC dataset results

Analysing these results, despite an increase in users of approximately 400% concerning the NOIZEUS dataset, we draw the same conclusions. Therefore, to better study the scalability of the total number of users of these approaches, the following dataset, TIMIT, composed of 462 users and eight speeches for each user, sub-datasets consisting of 38, 77 and 156 users are created. With this, in the table 5.5, the obtained results are presented.

		with Data Augmentation	**without Data Augmentation**
TIMIT (38)	Neural Network (Multilayer Perceptron)	100%	94,74%
	Linear Discriminant Analysis	100%	100%
TIMIT (156)	Neural Network (Multilayer Perceptron)	100%	94,74%
	Linear Discriminant Analysis	100%	100%
TIMIT (156)	Neural Network (Multilayer Perceptron)	100%	91,28%
	Linear Discriminant Analysis	100%	100%
TIMIT (462)	Neural Network (Multilayer Perceptron)	100%	87,88%
	Linear Discriminant Analysis	92,21%	95,67%

Table 5.5: TIMIT dataset results

Regarding the experiments performed with the TIMIT dataset, we can conclude that both algorithms scale well with the increase of users and the use of data augmentation positively influences the performance of neural networks. We also observed that, unlike LDA, only Neural Network managed to obtain an accuracy of 100% in the experiment with 462 users. Finally, and to analyze the relation of the number of speeches per user, a final test of the SpeechLibris dataset is carried out. This has a total of 156 users, however, each user has an average of 77 speeches. It is also necessary to be aware that each speech, on average, has approximately twice the work of the speeches of the other datasets. Thus, Table 5.6 presents the results of this dataset with the previously tested TIMIT sub-dataset.

With these results, we observed that data augmentation has a negative performance on the LDA and a positive one on the neural network. This indicates that the increase in the amount of data per user influences the LDA performance negatively and the neural network positively.

		with Data Augmentation	without Data Augmentation
LibrisSpeech	Neural Network (Multilayer Perceptron)	100%	98,63%
	Linear Discriminant Analysis	93,15%	93,84%
TIMIT (156)	Neural Network (Multilayer Perceptron)	100%	91,28%
	Linear Discriminant Analysis	100%	100%

Table 5.6: Libris dataset results

5.3 Conclusion

At the end of this experiment, we can conclude that both these approaches have excellent performance and scalability given the increase in the number of users, however when faced with a large number of users, the neural network is superior. In general, the amount of data (number of speeches) influences the final performance of the techniques. In LDA, its accuracy is low, and in the neural network, it increases. This is observable both in the experiment carried out on the dataset with the LibrisSpeech and TIMIT (156) dataset and in the use of the data augmentation technique.

This page is intentionally left blank.

Chapter 6

Conclusion

With a growth in the number of devices with a greater computational capacity, the need to innovate the human-machine interaction was necessary. Furthermore, with the current technological advances in speech processing and natural language processing, the possibility of interacting with devices has been created in the most natural way human beings have to communicate, the voice. Given its importance, some characteristics that deserve to be explored are identified, one of which is authentication.

As a result, the concepts of authentication, authorisation, and integration into a virtual voice assistant are explored. Having this, the different open-source assistants are analysed and studied. After identifying the assistant, a more detailed study is carried out on the different types of the Speaker Recognition System (SRS), its various components that characterise it and their respective techniques. Additionally, the different problems that need to be solved to build a successful Speaker Recognition System (SRS) are also identified.

Finally, after choosing the datasets to be used, the investigation carried out in implementing a Speaker Recognition System (SRS) is put into practice. The tests performed to the two selected modelling techniques (LDA and MLP) to the different data sets concluded that the quality and quantity of the dataset are essential for a good performance of the models. Through the data augmentation technique, we proved that a very small amount of data related to a user ends up harming the model's performance. This is confirmed through the Libris and TIMIT datasets (with 156 users). Their performance (without data augmentation) is relatively superior in the Libris dataset as this has a larger amount of data for each user. On the other hand, when there is a small number of users, considering that there is a considerable amount of data from each user (3 to 4 speeches), it is not necessary to increase the data. Therefore, it is possible to observe that the respective models' performance starts to worsen with the increase in the number of users and the reduction of data for each user. We also achieved that for a system with a reduced amount of users, both models reach a good performance, however, when we are present, a large amount of Multilayer Perceptron (MLP) is superior.

Despite presenting a system capable of recognising the user, this being a very complex system, it is necessary to continue studying the techniques and data processing and modelling to improve them. Some tasks that deserve to be highlighted are developing a technique capable of clearing signal noise for future work. As mentioned throughout the document, the system comprises the processing and modelling part. With this, it was important to focus only and exclusively on one of these components and develop their respective modelling and feature extraction techniques for future work.Voice activity detection and model to predict the user's gender.

This page is intentionally left blank.

References

[1] Inc. © Mycroft AI. Mycroft AI.

[2] Inc. © Mycroft AI. Mycroft Git.

[3] Kydyrbekova Aizat, Othman Mohamed, Mamyrbayev Orken, Akhmediyarova Ainur, and Bagashar Zhumazhanov. Identification and authentication of user voice using DNN features and i-vector. *Cogent Engineering*, 7(1), 2020.

[4] Open Assistant. Open Assistant.

[5] Ayehu. Human Learning Vs. Machine Learning – What's The Difference.

[6] Tom Bäckström. Gaussian mixture model (GMM), 2020.

[7] Tom Bäckström. Neural networks, 2020.

[8] Tom Bäckström. Vector quantization (VQ), 2020.

[9] Tom Bäckström. Voice activity detection (VAD), 2020.

[10] Smarajit Bose, Amita Pal, Anish Mukherjee, and Debasmita Das. Robust speaker identification using fusion of features and classifiers. *International Journal of Machine Learning and Computing*, 7(5):133–138, 2017.

[11] Supaporn Bunrit, Thuttaphol Inkian, Nittaya Kerdprasop, and Kittisak Kerdprasop. Text-independent speaker identification using deep learning model of convolution neural network. *International Journal of Machine Learning and Computing*, 9(2):143–148, 2019.

[12] Nebi Caka. What are the Spectral and Temporal Features in Speech signal ?, 2015.

[13] Oscar Contreras Carrasco. Gaussian Mixture Models Explained. 2019.

[14] Sandipan Chakroborty and Goutam Saha. Improved text-independent Speaker Identification using fused MFCC & IMFCC feature sets based on Gaussian filter. *World Academy of Science, Engineering and Technology*, 35(11):613–621, 2009.

[15] Amol Chaudhari and S. B. Dhonde. A review on speech enhancement techniques. *2015 International Conference on Pervasive Computing: Advance Communication Technology and Application for Society, ICPC 2015*, 00(c):6–8, 2015.

[16] François Chollet et al. Keras, 2015.

[17] Ieee International Conference and Signal Processing. PHYSIOLOGICALLY-MOTIVATED FEATURE EXTRACTION FOR SPEAKER IDENTIFICATION Jianglin Wang , Michael T . Johnson Speech and Signal Processing Laboratory Department of Electrical and Computer Engineering. pages 1709–1713, 2014.

[18] Julianna Delua. Supervised vs. Unsupervised Learning: What's the Difference?

[19] Ketan Doshi. Audio Deep Learning Made Simple (Part 3): Data Preparation and Augmentation. 2021.

[20] Keith W. Godin, Seyed Omid Sadjadi, and John H.L. Hansen. Impact of noise reduction and spectrum estimation on noise robust speaker identification. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 3656–3660, 2013.

[21] Gaurav Goel. Human Learning vs Machine Learning.

[22] Louis Grenard. Leon Git.

[23] Mayank Gupta. What, When and Why Feature Scaling for Machine Learning.

[24] Ujjwal Gupta. Stephanie.

[25] Ujjwal Gupta. Stephanie Git.

[26] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer. End-to-End Text-Dependent Speaker Verification. (Section 3):3–7.

[27] Matthew B. Hoy. Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants. *Medical Reference Services Quarterly*, 37(1):81–88, 2018.

[28] Ieee International, Workshop On, Machine Learning, and F O R Signal. SPEAKER IDENTIFICATION AND CLUSTERING USING CONVOLUTIONAL NEURAL NETWORKS Yanick Lukic , Carlo Vogt , Oliver D ¨ Zurich University of Applied Sciences , Winterthur , Switzerland. 2016.

[29] Irshivangini. Authentication vs. Authorization Defined: What's the Difference?, 2020.

[30] Amna Irum and Ahmad Salman. Speaker verification using deep neural networks: A review. *International Journal of Machine Learning and Computing*, 9(1):20–25, 2019.

[31] Shuangshuang Jiang, Hichem Frigui, and Aaron W. Calhoun. Speaker identification in medical simulation data using fisher vector representation. *Proceedings - 2015 IEEE 14th International Conference on Machine Learning and Applications, ICMLA 2015*, pages 197–201, 2016.

[32] Joel Jogy. How I Understood: What features to consider while training audio files? 2019.

[33] Andrei Dobre John McGonagle Geoff Pilling Andrei Dobre John McGonagle, Geoff Pilling. Gaussian Mixture Model.

[34] William M. Fisher Jonathan G. Fiscus David S. Pallett Nancy L. Dahlgren Victor Zue John S. Garofolo, Lori F. Lamel. TIMIT Acoustic-Phonetic Continuous Speech Corpus.

[35] Kalliope. Kalliope.

[36] Kalliope. Kalliope Git.

[37] Ahilan Kanagasundaram, David Dean, Sridha Sridharan, and Clinton Fookes. DNN based Speaker Recognition on Short Utterances. (1), 2016.

[38] Patrick Kenny, Vishwa Gupta, Themos Stafylakis, Jahangir Alam, and Pierre Ouellet. Deep Neural Networks for extracting Baum-Welch statistics for Speaker Recognition Speaker Odyssey 2014 Outline Introduction to DNNs in Speech & Speaker Recognition. *Proceedings of Odyssey 2014 - The Speaker and Language Recognition Workshop*, (June):1–18, 2014.

[39] Tai Hoon Kim, Wai Chi Fang, Muhammad Khurram Khan, Kirk P. Arnett, Heau Jo Kang, and Dominik Ślzak. Communications in Computer and Information Science: Preface. *Communications in Computer and Information Science*, 123 CCIS(January), 2010.

[40] De Lausanne. DEEP NEURAL NETWORK BASED POSTERIORS FOR TEXT-DEPENDENT SPEAKER VERIFICATION Ecole Polytechnique F ´.

[41] Phillip L De Leon and Senior Member. Speaker Model Clustering for Efficient Speaker Identification in Large Population Applications. 17(4):848–853, 2009.

[42] Lantian Li, Dong Wang, Askar Rozi, and Thomas Fang Zheng. Cross-lingual speaker verification with deep feature learning. *Proceedings - 9th Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2017*, 2018-Febru:1040–1044, 2018.

[43] Librosa. Librosa.

[44] Philip Loizou. NOIZEUS: A noisy speech corpus for evaluation of speech enhancement algorithm.

[45] Louis Grenard. Leon.

[46] Edward Ma. Data Augmentation for Audio, 2019.

[47] Zhanyu Ma, Hong Yu, Zheng Hua Tan, and Jun Guo. Text-Independent Speaker Identification Using the Histogram Transform Model. *IEEE Access*, 4:9733–9739, 2016.

[48] Spyros Matsoukas, Pavel Mat ˇ, Najim Dehak, Jeff Ma, S Cumani, O Glembek, H Hermansky, S H Mallidi, N Mesgarani, R Schwartz, M Soufifar, Z H Tan, S Thomas, B Zhang, and X Zhou. DEVELOPING A SPEAKER IDENTIFICATION SYSTEM FOR THE DARPA RATS PROJECT Old ˇ Brno University of Technology , Speech @ FIT , Brno , Czech Republic Raytheon BBN Technologies , Cambridge MA , USA Massachusetts Institute of Technology , Cambridge MA , USA The J. *Technology*, pages 6768–6772, 2013.

[49] Brian McFee, Colin Raffel, Dawen Liang, Daniel Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and Music Signal Analysis in Python. *Proceedings of the 14th Python in Science Conference*, (Scipy):18–24, 2015.

[50] Medevel. Open-source Voice Assistants Projects, 2018.

[51] Vikramjit Mitra, Mitchel McLaren, Horacio Franco, Martin Graciarena, and Nicolas Scheffer. Modulation features for noise robust speaker identification. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, (February 2016):3703–3707, 2013.

[52] Steve Penrod Mycroft AI's CTO. Mycroft Architecture: Current and future capabilities.

[53] Okta. Authentication vs. Authorization.

[54] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR corpus based on public domain audio books. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2015-Augus:5206–5210, 2015.

[55] Kashyap Patel and Rk Prasad. Speech Recognition and Verification Using MFCC & VQ. *International Journal of Emerging Science and ...*, 2013.

[56] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, , B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, , R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, , D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in {P}ython. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[57] Baijayanta Roy. All about Feature Scaling.

[58] Seyed Omid Sadjadi and John H.L. Hansen. Mean Hilbert envelope coefficients (MHEC) for robust speaker and language identification. *Speech Communication*, 72(January):138–148, 2015.

[59] Sagar Sharma. Activation Functions in Neural Networks, 2017.

[60] Maxim Sidorov, Alexander Schmitt, Sergey Zablotskiy, and Wolfgang Minker. Survey of automated speaker identification methods. *Proceedings - 9th International Conference on Intelligent Environments, IE 2013*, pages 236–239, 2013.

[61] Satyanand Singh and E.G. Rajan. Vector Quantization Approach for Speaker Recognition using MFCC and Inverted MFCC. *International Journal of Computer Applications*, 17(1):1–7, 2011.

[62] David Snyder, Pegah Ghahremani, Daniel Povey, Daniel Garcia-Romero, Yishay Carmiel, and Sanjeev Khudanpur. Deep neural network-based speaker embeddings for end-to-end speaker verification. *2016 IEEE Workshop on Spoken Language Technology, SLT 2016 - Proceedings*, pages 165–170, 2017.

[63] Devin Soni. Supervised vs. Unsupervised Learning.

[64] Alaa Tharwat, Tarek Gaber, Abdelhameed Ibrahim, and Aboul Ella Hassanien. Linear discriminant analysis: A detailed tutorial. *AI Communications*, 30(2):169–190, 2017.

[65] Neil Tyler. Voice recognition. *New Electronics*, 51(21):12–14, 2018.

[66] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez. Deep neural networks for small footprint text-dependent speaker verification. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 4052–4056, 2014.

[67] Raghav Vashisht. When to perform a Feature Scaling?

[68] Analytics Vidhya. Feature Scaling for Machine Learning: Understanding the Difference Between Normalization vs. Standardization.