Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

School of Engineering and Sciences

Identification of Pronunciation Errors in L2 English Speech by Spanish Speaking Natives for S-Impure Sounds.

A thesis presented by

José Aristh Valdiviezo Mora (Student)

Submitted to the
School of Engineering and Sciences
in partial fulfillment of the requirements for the degree of

Master of Science

in

Intelligent Systems

Monterrey, Nuevo León, Dec, 2019

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

School of Engineering and Sciences

The committee members, hereby, certify that have read the thesis presented by José Aristh Valdiviezo Mora (Student) and that it is fully adequate in scope and quality as a partial requirement for the degree of Master of Science in Intelligent Systems.

Dr. Ramón Felipe Brena Pinero
Tecnológico de Monterrey
Principal Advisor

Dr. Rogelio Soto Rodríguez
Tecnológico de Monterrey
Committee Member

Luis Alberto Ávila Rodríguez
Neoris
Committee Member

Román De León Flores
Talisis
Committee Member

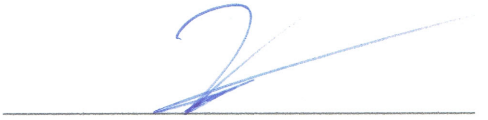
Dr. Rubén Morales
Associate Dean of Graduate Studies
School of Engineering and Sciences

Monterrey, Nuevo León, Dec, 2019

Declaration of Authorship

I, José Aristh Valdiviezo Mora (Student), declare that this thesis titled, "Identification of Pronunciation Errors in L2 English Speech by Spanish Speaking Natives for S-Impure Sounds." and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this dissertation is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.



José Aristh Valdiviezo Mora (Student)
Monterrey, Nuevo León, Dec, 2019

Dedication

This Thesis work is dedicated to those that have thrust in me, providing support, love, and encouragement to pursue and conclude this step in my professional development. For my Family, Friends, and my Love.

Thank you all for this!

Acknowledgements

First, I want to thank my Advisors: Dr. Brena, thank you for leading me to this point in my professional career; you helped me to learn invaluable lessons, thanks for your time and dedication to guide me in this process; Dr. Soto, I really appreciate your support and your time to be one committee member in this thesis, the Control Engineering course that I took from you as part of my professional degree was one of the best courses of all time; Master Román and Master Ávila, thank you both for being part of my committee, your reviews and comments were terrific, the quality of this document would not be as great as it is without your help; I really appreciate it.

This work is also a result of all the Teachers that I had during my time as an Undergraduate and Graduate Student. Your guidance and passion for knowledge will always remain engraved in my memories. I'll not mention your names here because I don't want to forget any of you, but I'm pretty sure I've left you a good impression as a student. Thank you all!

I also would like to express my sincere gratitude to my alma mater, Tecnológico de Monterrey, for allowing me to live my Professional and Master's degree experience. My sincere appreciation to CONACyT, for supporting me during my Master's studies.

This major milestone was also possible thanks to all my friends, that made my living in Monterrey an extraordinary experience, and my Family, that has always been with me, giving me strength, wisdom and guidance.

An exceptional thanks to Jacqueline, who shared with me the experience of writing this document, stayed by my side the long nights of work making corrections and giving me priceless advice, continued support, and unconditional love. I would like to especially thank you for encouraging me to complete this Thesis and being my life partner in this journey called life. This work is also yours.

A very special thanks to my Granny, Abu. You and the rest of my Family have made me the person I'm today.

Last but not least, I thank God for allowing me to stay here today.

Identification of Pronunciation Errors in L2 English Speech by Spanish Speaking Natives for S-Impure Sounds.

by

José Aristh Valdiviezo Mora (Student)

Abstract

In the field of Computer-Aided Pronunciation Training (CAPT) systems, there are several approaches to detect pronunciation errors. Among those, the cutting-edge in the past years has been Deep Neural Networks (DNN), but this approach is generally only feasible when a high quantity and quality of data are available. In this project, a big database was not available. For that reason a dataset of 1953 audio files was sampled and collected; however a database of this size is considered small and not entirely suitable for a DNN model. Therefore classical supervised learning techniques were revised, applied and tested in this work. The main goal was to identify the pronunciation errors of native Spanish speakers at pronouncing S-impure words. The database build from the 1953 tagged audios was binary classified by three independent judges to identify those recordings that have an S-impure error and later processed to extract key features, Mel Frequency Cepstral Coefficients (MFCC), Spectral-Flux (SF), Root Mean Square Energy (RMSE) and Zero-Crossing (ZR) to train with a K-Nearest Neighbors (KNN), Random Forest (RF) and Support Vector Machine (SVM) algorithms. The resulting model obtained with SVM using the grid-search technique for tuning Hyperparameters provided the best solution. The obtained results show that the model was able to detect errors with an accuracy of 85% which leads to a solid result given that the dataset had high noise levels.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

In this Chapter, a general introduction is presented. Chapter 02 brings a theoretical background and current state of the art. Chapter 03 presents the general method applied in this work, as well as the required steps to reproduce the results. In Chapter 04 a detail of the experiments is presented along with the list of words used to build the dataset and train the classificators. Chapter 05 is about the discussion of the results, future work, limitations, and improvements that can be explored in the following works.

The current research work is elaborated with the main purpose of detect pronunciation errors from the English Language Hispanic learners. In order to achieve that purpose, this work implemented a split and segmentation methodology, along with AI algorithms to identify mispronunciation patterns in a set of collected audios for the S-impure (Words that start on S and is following by one or two consonants) sounds, which are one of the most common pronunciation and region characteristic errors of Hispanic learners[1]. This classification and pronunciation error detection is intended to be implemented into a feedback system that allows the L2 English Learners to have feedback about the presence of an error at the moment of speak. This work is intended to be a baseline benchmark for the AVALINGUO research team, which is aimed to implement a Computer Assisted Language Learning (CALL) system based

on Virtual Reality and dynamic feedback to the learner about their pronunciation and fluency.

1.1 Motivation

Through time, languages have been a fundamental part of the development of modern society. Since the early stages of civilization, languages allowed humans to interact with each other, sharing ideas, thoughts, and emotions [2]. Later, the necessity of interaction between populations imposed a need to learn a second language to trade and exchange technology and ideas. Today in a globalized world, learning a second language has become an important requirement for professional and personal growth. The importance of proper oral communication in work environments and everyday activities is essential to avoid misunderstandings and improve personal self-esteem and confidence, as pointing in research, the pronunciation errors could be seen by Americans or native speakers as unpleasant [3] or even unattractive [4], as well could provide undesired prejudices about education and cause discrimination.[5]

In terms of importance, the widespread of the English language has been increasing over the years. Nowadays, learning English is considered a necessary skill for science, communication, and to be on the cutting edge of almost any science field [6]. In the same way, in the business field, English is considered Lingua Franca [7]. Regarding speaking, the English language is the third most spoken language in the world and accepted as the language of science [8, 9]. In the same line, it is interesting to note that the majority of the English speakers lived in countries where English is not the primary language, English is used internationally to access to higher education, better employability, and prestige. [10] Therefore the necessity of master English language for multiple purposes in the personal, academic and professional fields. However the learning of a new language brings important challenges to the newcomers; challenges related to grammar, vocabulary, pronunciation and intonation [10, 11, 1, 12]

To overcome those challenges a new research field was established, Computer Assisted

Language Learning (CALL) systems, and thanks to the wide availability of computational power at the disposal of the general population with the Smartphones and Computers, the widespread of Applications like Duolingo and similar apps provide to the student a way to improve his English skills. Unfortunately, there is a missing field that has not Hispanic, which is related to identifying pronunciation errors; that is why a specific field of CALL systems takes relevance. The Computer-Aided Pronunciation Training (CAPT) systems, is an essential part of any CALL system [13] and have the specific purpose of providing feedback to the learner about his pronunciation performance. A typical CAPT system or tool records the speech, applies algorithms, extracts audio features, and provides an output, diagnosing and detecting mispronunciations in it, as well provides information about the error, if it is present or not. [10]

According to [12] there is a resurgence in Computer-Assisted Pronunciation Training (CAPT) systems research, which has had a growth due to the increase of students of languages, many motivated by the economic increase and the high tide of opportunities to acquire foreign languages, especially English. There are several research projects and the development of efforts with the use of spoken language technology in education. One of the advantages of these kinds of systems is to complement the feedback that usually would come from an English teacher.[13, 10]

Historically the way a pronunciation error is analyzed is done with a probability-based score, which is a standard method nowadays, commonly referred to as Pronunciation Goodness (GOP). In practice, this probability is often used to quantify the difference between the source word against the pre-defined pronunciation target. [12] This approach has a drawback; that is because a probability approach and measure against a predefined pattern does not detect or diagnoses which pronunciation error is present in the audio. The way to address this classification challenge now has changed toward the use of Supervised Learning, Unsupervised Learning, and Deep-learning approaches. [12, 13, 10, 14]

The vast majority of the tools available for language learning focuses on providing a classical methodological approach, which involves repetition and grammar. Some of them include voice practice, but fail to identify word patterns and to provide feedback on mispronunciations. One of the tools that are specially intended to improve the pronunciation is Elsa Speaks. [15] However, its implementation and the actual benchmarks are unknown to the community, and an open-source work for assisting language learning is needed for continuous development in the field. In the same way, no related work using Artificial Intelligence techniques has been done in the Spanish community.

In the present work, we applied modern AI techniques and a methodological approach to identify pronunciation errors among native Spanish speakers learning English as a second language (L2).

1.2 Problem Statement and Context

When people try to learn a new language, a huge barrier that needs to be overcome is the ability to communicate effectively orally. In the context of a foreign learner, that does not have, in several cases the possibility to have a conversation with a native speaker or professional Teaches, this represents a problem for the learner as usually, they tend to associate foreign L2 language sounds with L1 mother language [16, 17] getting errors like disyllabism or phonemic intrusion of similar letter sound from L1 Language [1]. In the same case, if a learner feels ashamed of his language skills or is unable to perform correctly, effective communication becomes a barrier. [18, 19, 1].

Under this context and there is a necessity of learning a second language, there should be an improvement in the way a foreign language is taught. Nowadays, the use of mobile applications has been great acceptation as a compliment or primary source for learning the perks of a new language, which mainly focuses on grammar and passive pronunciation, which means the

user usually hears a modulated voice. When an actual CAPT approach is attempted, usually it does not include an actual pronunciation error detection; instead, a metric of how similar a word sounds compared with the predefined waveform is used. An issue with this approach is the lack of feedback to the user about what kind of error is present; if the performed error is not noticed and informed to the speaker on early stages, it could be a bad habit difficult to break. It is known that reliable feedback in early stages is fundamental to improve language performance[17]. For that reason, make the L2 English student aware of the error is important.

There are research works where is shown that the key pronunciation errors of native Spanish speakers at the moment of performing a phoneme are related with phonemes like [ʃ], [ʒ], [ɵ], [ʧ], [ɹ], [ʤ], as well, one of the most commons is the S sound.[11] Therefore, one of the characteristical points to identify a Spanish speaker is the S-Impure sounds. These kinds of errors cause some discomfort in the interaction between a native speaker and a learner and could represent a communication barrier.[1]. Unfortunately, no current CALL system addresses those kinds of errors to let the user know that he is making that kind of mistake.

Computer-Assisted Language Learning (CALL) systems have been in constant development; today there exist tools available for free that have the objective of helping a language learner to learn grammar and linguistics of a language. Among those applications one of the most commonly known today is Duolingo [20]. This application allows the user to improve his English skills and vocabulary but lacks in the field of intonation and pronunciation error detection.

As today's technologies do not offer a reliable error detection feedback system that allows a user to be aware of the pronunciation error he has, an area of opportunity was present here. For that reason, this work presents a CAPT method to classify S-impure words pronunciation errors. This baseline work is the first attempt to correctly point a particular phonetic error for Spanish natives learning L2 English.

1.3 Scope

This thesis work is focused on getting the S-impure mispronunciation words from Latino-Hispanic people who have English as L2 Language. The proposed method was tested against a database built on demand due the lack of a suitable database available on the web.

Also, this work only tested three classical Machine Learning techniques avoiding Deep Learning approaches due to the small size of the dataset even though this audio preprocessing is intended to be suitable to be consumed by other Machine Learning techniques and be able to be used as input for Deep Learning algorithms once the database size is increased.

1.4 Hypothesis

The feature extraction based on previous work developed by the AVALINGUO Research team [21] will be useful to match the reported accuracy of CAPT systems [10] when applied to the collected dataset composed with native Spanish speakers on L2 English S-impure words.

1.5 Hypothesis proof

The proposed hypothesis will be tested against the result obtained with the feature extraction from the audio files with three classical Machine Learning Algorithms (SVM, KNN, RF) with tunned parameters using Grid Search with Cross-Validation techniques.

1.6 Methodology

The methodology implemented in this work to reach the solution consist of the following steps:

1. Identify the General Problem
2. Research programming Basics
   - Python Programming Language
   - Machine Learning
   - Audio Open Source Libraries
   - State of the Art
3. Research the most common pronunciation errors in English
   - Select a phonetical error to classify.
4. Elaborate a list of words that contains the error.
5. Randomize the list of words that contain the phonetical error with other words.
6. Collect audios to generate a database.
7. Split the audios by word.
8. Shift the audio waveform to guarantee all the records start with the phoneme.
9. Normalize the length of the audios.
10. Extract the features.
11. Input the features to the Classificator.
12. Sample the classificator, train, and cross-validate.
13. Compare and get the best Classificator.
14. Validate the results against the reported values in literature.
15. Publish of results.

Chapter 2

Theoretical Framework

In this Chapter, a theoretical framework is going to be presented. The most important concepts are going to be reviewed to understand the general idea of the algorithms, the reason why the descriptors to extract the audio features were selected, and be able to understand related works in this field.

2.1 Sound

Describing sound is something we know exists, but sometimes we are not capable of fully define. In fact, there is a complete study that explains in layman words what sound is [22]. In this section, a concise definition of sound will be provided from the physics point of view.

Sound is defined as ”(a) Oscillation in pressure, stress, particle displacement, particle velocity, etc., propagated in a medium with internal forces (e.g., elastic or viscous), or the superposition of such propagated oscillation. (b) Auditory sensation evoked by the oscillation described in (a).” [23]. Given this definition, we can conclude that the sound is a Wave, has vibrations and that involves several physical properties as displacement, velocity, period and it is medium dependent.

Along with the sound, there are some properties that came along with it like the followings:[23]

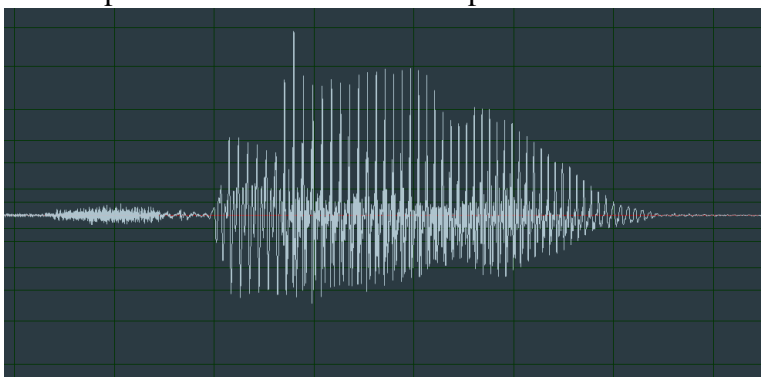
- Pitch.- That attribute of auditory sensation by which sounds are ordered on the scale used for melody in music.
- Tone.- (a) Sound wave capable of exciting an auditory sensation having pitch. (b) Sound sensation having pitch.
- Frequency.- The rate of change with time of the instantaneous phase of a sine function divided by 2π, with the dimensions of cycles per second or hertz (Hz).
- Scale.- Series of notes (symbols, sensations, or stimuli) arranged from low to high by a specified scheme of intervals, suitable for musical purposes.
- Wave. Disturbance such that the quantity serving as a measure of the disturbance varies with position and time in a manner that at pairs of neighboring positions the disturbance is similar except for a time difference.
- Amplitude. In ultrasonic testing, the vertical pulse height of a signal, usually base to peak, when indicated by an A-scan presentation.

2.1.1 Graphical Representation

Waveform

Graphically a wave of sound could be represented as a waveform, which is a graphical plot of the audio signal where the vertical axis represents its intensity and the horizontal axis represents time. A typical example of how a waveform looks like could be appreciated in figure 2.1

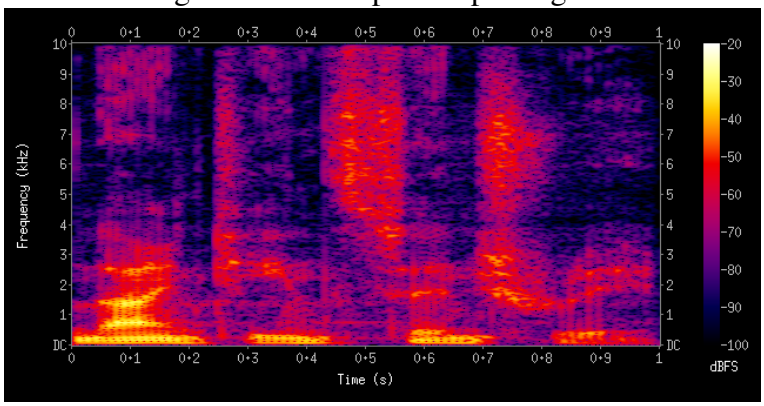
Figure 2.1: Example of a Waveform: sound representation of the word ”smile”



Spectrogram

Another, more complete way to represent a waveform, which is the most complete, is through a Spectrogram. The spectrogram shows the audio frequencies over time, along with its magnitude. This could be seen as a topographical map, where the color represents the height or intensity of the frequency associated. The Y-axis shows the frequency of the signal and X-axis the time. A clear representation of a typical Spectrogram is shown in figure 2.2

Figure 2.2: Example of Spectrogram



2.1.2 Audio Formats

In this section, an overview of the most common available audio formats is presented. This small description includes some comments about the formats available and the reason why

WAV was chosen as the storage and processing format for the audio files gathered in this project.

Basically, at the moment of deciding which audio format choose there are two selection criteria: Compression and Information preservation. In the Compression criteria, the formats as MP3 is one of the most common and widely available, it has the advantage of being small in size and useful for streaming or distribution, but on the other side, the level of compression of an MP3 format carry some loss of information, for this reason, these kinds of formats are only intended to be useful in the distribution part. In the other part there exist the uncompressed formats. These kinds of formats allow to directly access the information without extra computer process which makes it fast for processing but has the drawback of file size, which could tend to be large. Under the previous reasoning, there is a list of commonly used formats along with his characteristics, each one of the following formats has a tradeoff between Speed and Storage.

- MP3.- This audio format was designed by the Moving Pictures Expert Group (MPEG) in 1993. MP3 is a compressed and lossy audio format. This means that due to its nature, the information stored and processed on MP3 gets lost over time or over layers of processing. [24]
- AAC.- Audio format with multiple benefits over MP3, like added support for channels and better compression rates; but it still is a compressed and lossy audio format. This has the same drawbacks as MP3, and it is not a good design choice to process audio. [24]
- WMA.- Was a format developed by Microsoft as a response for MP3 with a proprietary standard. Usually, WMA had the same drawbacks as MP3 and AAC for processing, as it is a lossy audio file. However modern versions with codecs allow lossless information.
- AIFF.- An audio format developed by Apple. The Audio Interchange File Format (AIFF) is standard on Mac computers. This format was designed in mind for storage and editing as contains lossless format.
- WAV.- This format was developed by Microsoft. The Waveform Audio File Format is standard for windows and cross-platform compatible. This format has a big widespread

in library and open source support, and due its lossless qualities when using with PCM codec, wide availability and support were chosen in this project.

- PCM.- the Pulse Code Modulation (PCM) is a lossless format that contains all the waveform signal information to reproduce the information without data loss. This is the standard audio format for CDs and digital audio applications.
- FLAC.- Was developed by Xiph.Org Foundation. The Free Lossless Audio Codec (FLAC) is open source. It is considered a good mean for the storage of audio information as it allows lossless compression. Unfortunately due its compressing nature it is not the preferred choice to use in audio processing as extra computer power is needed to uncompressed and compress.
- ALAC.- The Apple Lossless Audio Codec (ALAC) is the Apple response to FLAC. It is a proprietary format only available for Apple devices. Due to its proprietary format, it is not suitable to use in open source projects.

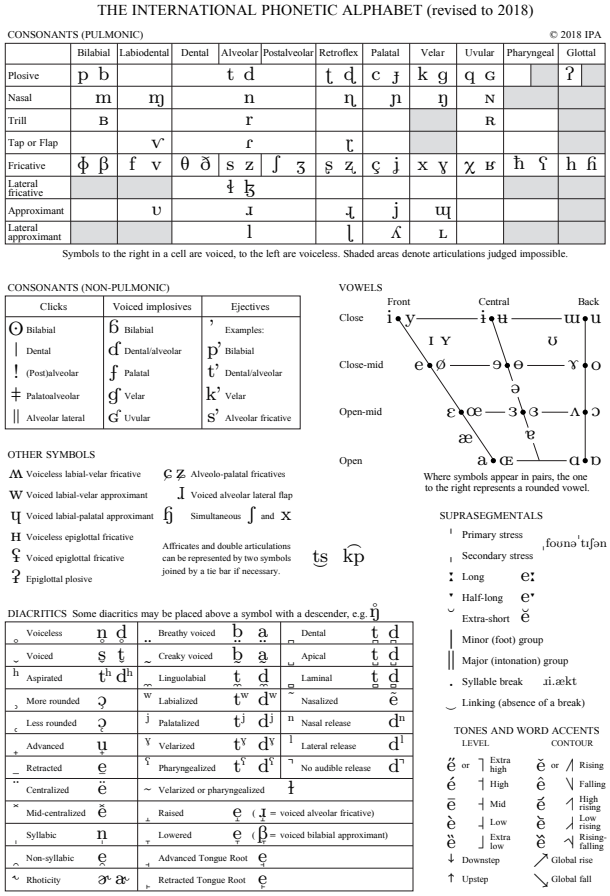
2.2 Phoneme and Pronunciation Error

2.2.1 Phoneme

A phoneme is defined as the smallest unit of speech distinguishing one word (or word element) from another [25]. In other words, a phoneme is an audible word represented by a letter, this kind of differentiation allows to distinguish between different words, i.e. “p” phoneme helps to differentiate “tap” from “tab”, “tag” or “tan”. In this case the word “tap” has three phonemes /t/, /a/, /p/.

A common way to distinguish between the different phonemes is through the International Phonetic Alphabet (IPA). This alphabet was developed in the 19th century to represent the pronunciation of languages. The general idea was to provide a unique symbol for each different sound in a language, in this way, having an alphabet of sounds.

Figure 2.3: International Phonetic Alphabet Guide.- Sample of avaliable combinations of sounds [26]



2.2.2 Pronunciation Error

Across different authors, pronunciation is a general term that covers a wide set of features, and it is an entirely subjective term in the hearing of the listener. As mentioned on [27] “pronunciation error is difficult to quantify. For that reason, more than a right or wrong pronunciation, there is a scale from unintelligible speech to native-sound speech.

Generally speaking, the pronunciation errors by non-native speakers can result in two causes:

- Phonemic.- Considered as the ”severe” kind of error, as a phoneme is deleted, substituted, or inserted.

- Prosodic.- The error can be categorized in terms of stress, rhythm, and intonation.

One common error on the pronunciation error field is the phoneme identification. Turns out that the shorter the phoneme is, the higher the variability in the judgment of the pronunciation quality. For this reason, identifying a pronunciation error at phoneme level is more complicated than the measurement of fluency.[27]

2.3 Audio Descriptors

In order to be able to classify audio, a series of key features must be computed from the raw audio files. These features allow the classification algorithm to have specific components related to the audio signal. These features are represented in a compact numerical representation to characterize the segment. Among the features that can be extracted from audio, there are the following: Volume, Pitch, Energy, Zero Crossing Rate (ZCR), Mel Coefficients (MFCCs), Spectral Flux (SF), Root Mean Squared Energy (RMSE). In particular, this work uses the latest four, as they have been reported to give high accuracy performance in the audio classification [21]

2.3.1 MFCCs - Mel Frequency Cepstral Coefficients

The Mel Frequency Cepstral Coefficients is one of the essential feature descriptors using nowadays in the field of audio recognition and preprocessing [28] This descriptor was introduced first in 1980 [29] and has a wide acceptation as the default scale to measure sound intensity.

The general idea of this Feature is to model human perception sensitivity. The way this is accomplished is using a series of filters where the high frequencies are processed in a logarithmic way and low frequencies linearly. In this way, this Feature Descriptor emulates the human ear and provides useful information to discern different phonemes. [29]

Going more in-depth, the MFCCs are based on the Mel Scale which has the "Mel" as the base unit. The Mel is defined as a measure of perceived pitch or frequency of a tone [30], and the key component is that it doesn't correspond linearly to the physical frequency of the tone. This scale was based on the research of Steven and Volkman whose arbitrarily chose the frequency of 1000 Hz and designed it as 1000 Mels then the listeners were asked to variate the frequency until the pitch perceived were twice as different. Based on these discoveries the authors were able to map this frequency. The conversion of Mel to frequency varies between authors, but those two equations are greatly accepted.

$$f_{mel} = 1127 \ln \left(1 + \frac{f_{Hz}}{700}\right)$$

2.3.2 ZCR - Zero Crossing Rate

Zero-Crossing Rate is an important parameter to identify the presence or absence of sounds. This feature measures the number of times the amplitude of audio signals passes through a value of zero. [31]

For speech analysis, the voice is normally done in extracting this information in speech signals. Therefore, the zero-crossing rate (ZCR) and energy are used; this method is the most common to be able to measure the frequency or period of a periodic signal. The ZCR could measure the frequency applied to the zero-crossing count has as an indicator of the frequency with which energy is concentrated in the frequency in the signal spectrum.

The loud speech is produced due to the excitation of the vocal tract by the periodic flow of air in the glottis and often shows a low zero-crossing count. And while in voiceless speech there is a construction of the vocal tract that is narrow enough to cause turbulent airflow that produces noise and shows a high zero crossing count.

A mathematical definition of zero-crossing rate is the following [31]:

$$Z_n = \sum_{m\,=\,\infty}^{\infty} |sng\left[x\left(m\right)\right] - sgn\left[x\left(m-1\right)\right]|\, w\left(n-m\right)$$

where

$$sgn\left[\,x(n)\,\right] = \begin{cases} 1, x(n) \geq 0 \\ -1, x(n) < 0 \end{cases}$$

and

$$w(n) = \begin{cases} \frac{1}{2\,N}\, for,\ 0 \ \leq \ n \ \leq \ N \ - \ 1 \\ 0 for, otherwise \end{cases}$$

2.3.3 RMSE - Root Mean Squared Energy

The Root Mean Square Energy (RMSE) is an important feature to know details about the intensity applied to a signal. It is known that the way to obtain the Root Mean Square of a signal is given by[32]:

$$E_{RMS} \ = \ \sqrt{\frac{1}{T}\int_0^T x\left(t\right)^2 dt}$$

When dealing with discrete sampled values, the RMS of a signal with discrete values: $[x_0^2 + x_1^2 + \cdots + x_{N-1}^2]$ is given by

$$E_{RMS} \ = \ \sqrt{\frac{x_0^2 + x_1^2 + \cdots + x_{N-1}^2}{N}}$$

if the Frequency domain is depicted as X(f), then as a result of Parseval's theorem, the RMS value results in:

$$E_{RMS} = \sqrt{\frac{\sum |X(f)|^2}{N}}$$

This feature gives information about the mean energy given by a signal. as well as the energy of the signal spread uniformly over the full length of the audio file.

2.3.4 SF - Spectral Flux

Spectral flux is related to the change of the power spectrum of a signal; it gives and evaluates how quickly this is changing within a range. This property is useful to discriminating speech from other sounds. The inclusion of Spectral Flux has proved an improved accuracy over the baseline classificator.[21, 33]For this reason, it was chosen.

According to [34]: "The spectral flux of a frame is the sum of the squared distances between the normalized magnitudes of successive frequency bins. It is a measure of how rapidly the spectrum changes in frequency."

and this can be represented mathematically as [33]:

$$SF_p(t) = \|X_m(t,w) - X_m(t-1,w)\|_1$$

where $\|\|_1$ represents the L^1-norm, and $X_m(t,w)$ the energy normalized mel-spectrum at frame t. In this project, the mean of resulting vector was used as an input feature.

2.4 Machine Learning

The amount of data available in recent years has been increased significantly [35]. The handling of data and the way to get insights from data has opened the door for new kinds of jobs. Not until recently, the world was aware of a Data Science Engineer, Data Engineer, Big Data

Specialist, or Machine Learning engineer. These new fields of professional careers emerged thanks to the vast amount of data available nowadays and with them a considerable number of opportunities to get knowledge and benefits from the available information.

The Machine Learning concept can be defined as: a general multipurpose mutable algorithm that generates rules based on examples. This means a new parading in programming, and some authors have called this revolution as the Software 2.0 [36]. We can say that behind the Machine Learning field of study is the great desire of learning and get insights from data. There are some key areas of human knowledge that involve clear steps, like sort numbers, manipulate data, create a report, or consume data from a database. However, complex tasks like e-mail spam detection, medical images examination, pronunciation error detection are not clear enough how to program using predefined steps. For a task like this learning from the examples is like a human would do it, and it is the way machine learning emulates humans to get insights of data.

Machine learning nowadays has been a hot field in research activity; the great success could be explained thanks to the improvement in computer power available as well as the vast amount of information, along with improvements in Machine Learning algorithms [35]. This continuous race to see who has the best performing classificatory or the most efficient way to learn knowledge from data has allowed considerable success. Machine Learning provides a new set of tools to deal with problems that before would require to program a considerable amount of cases to contain the human knowledge in form of code. Now with the use of ML Algorithms, a huge set of rules can be constructed, just taking examples as input. This tool has opened a vast amount of possibilities to address new kinds of problems that before Machine Learning it would have been impossible to achieve [37].

The field of Machine learning has evolved in a way that we can say that there exist two different kinds of machine learning algorithms, Classical Algorithms and Neural algorithms. Classical Algorithms are based on statistics and mathematical models because the core task

is making the inference from a sample and Neural algorithms represent a new paradigm in the way a computer emulates the brain. Those have the same Idea as classical algorithms, but the approach has been controversial due to the complex explanations behind the "Why" a computer learns using Neural Networks.

An important point to consider at the moment of choosing a Machine Learning algorithm is to review the kind of problem it is being addressed, the amount of data available to consume, the level of accuracy needed, the computational power required, and the robustness desired. Usually, when moderate data is available Classical Machine Learning Algorithms, like Support Vector Machines (SVM), Random Forest (RF), or K-Nearest Neighbours (KNN) presents a good performance. When there is huge amount of data available, the Neural Networks algorithms have presented usually better accuracy than classical approaches.

Among the kind of Machine Learning Algorithms presented, there are usually three fields associated with the way the algorithm behaves [35]

- Supervised Learning.- Area of Machine Learning that requires a set of labeled data. The general idea is to generate a set of rules from the labeled sampled data to classify or predict an outcome correctly.
- Unsupervised Learning.- In the case of unsupervised learning, the general idea is let to the algorithm to perform an automatic classification based on the available examples. This kind of ML Algorithm aims to form clusters or sets of data that share common properties between each cluster or group.
- Reinforcement Learning.- This kind of ML Algorithm aims to learn via trial and error. The idea is to prioritize a policy instead of an individual action. This idea is powerful

During the development of this project, a classification algorithm was needed, through which a supervised learning approach was considered.

2.5 Supervised Learning: Classification Algorithms

Due the nature of the current project, the chosen way to solve it was using a set of supervised learning algorithms. The Supervised learning algorithms could be easier to understand as basically represents an emulation of how a human would learn when examples of what is right or wrong are presented. Under this approach, a set of labeled data which has been previously classified by a human represents the set of instructions. Those instructions will be fed to the system to generate the desired rules of operation.

One of the advantages over the other kinds of learning algorithms is Supervised Learning provides the most accurate way to identify the way an algorithm is learning as there is a direct comparison of the error during training [38] . This could be observed by looking at the performance metrics to evaluate a model generated with Supervised Learning. The way to evaluate the systems just became a scan over the correctly classified examples to the misclassified examples.

In this work, three of the classical algorithms were used to learn from the labeled data. These algorithms are Support Vector Machines (SVM), Random Forest (RF) and K-Nearest neighbors (KNN). These three algorithms are highly related to Automatic Speech Recognition (ASR) and pronunciation error detection.

2.5.1 Support Vector Machine (SVM)

Support Vector Machines remains as a state-of-the-art classification method. Was first introduced in 1992 [39] and has been applied to several fields of knowledge, among those, there are speech analysis, fraud detection, image recognition and bioinformatics [40] due to its ability to deal with multidimensional data [41].

SVM belongs to a category composed of kernel methods, which can be defined as an algorithm that depends on dot products. The general idea behind SVM is to be able to separate

data in an hyperplane correctly. This means that high dimensional data that contains multiple features can be approached by SVM kernels.

In terms of audio applications, SVM has been applied in several studies [42, 43, 44] where high performance has been reported. These statements and related work validate the capabilities of SVM to work with complex datasets.

Among the parameters commonly used with SVM for tuning are the following [41, 40]:

- Penalty Parameter (C).- This parameter measures the closeness to the discriminant area for linear separation
- Gamma.- A parameter to control the influence of new features, directly connected with kernel.
- Kernel.- Mathematical transformations that deal with the way the decision boundary is built.

Among the general advantages of SVM for classification there are the followings [45]:

- Effective in high dimensional spaces.
- Effective in cases where dimensions are greater than the samples
- Memory efficient
- Versatile in the kernel function

2.5.2 Random Forest (RF)

Random Forest Algorithm was first introduced in 2001 [46] and has become a standard non-parametric classification and regression tool to generating prediction rules [47]. The general idea behind random forest is to combine several randomized decision trees and aggregate their individual predictions by averaging, in the same way, this supervised learning algorithm is based on the ”divide and conquer” principle, as applies sample fractions of the training data and generates a randomized tree predictor on each small piece to then aggregate and predict the final result. [48]

Random Forest Algorithm has proven its value in different areas like Data Science for predictions, chemoinformatics, ecology, 3D object recognition, pronunciation error detection, bioinformatics, and so on. [48, 49]

While Random Forest usually performs well without the need for tuning, a correct tuning could provide generous improvement in the performance of the algorithm. The parameters to consider are the followings:

- Number of Estimators.- The size of the forest.
- Minimum Samples Leaf.- The minimum number of samples to become a leaf.

2.5.3 K-Nearest Neighbors (KNN)

K-Nearest-Neighbours is considered a simple but effective classification algorithm. It has the advantage of being easy to understand and interpret the results [50] However, there are some key points that should be considered; one of them is lack of efficiency and dependency of a good choice of the K parameter.

KNN has been used in a broad amount of fields, like Automatic Speech Recognition [49], Text classification [50], network monitoring [51], spam detector system [52], among others. The general idea behind KNN is to have a case-based learning method, which keeps all the training data for classification. [50]. Under this approach, KNN has built on the basis of a cluster group algorithm that has the purpose of classifying the points that met the distance criteria with the labeled examples.

The common hyperparameters needed to tune KNN are the followings:

- Number of Neighbors.- The number of points considered to match.
- Algorithm.- The algorithm used to compute the nearest neighbors
- Leaf Size.- A parameter used on the Tree algorithm on KNN

- Weight.- Kind of criteria used to put weights to the points closer to the point to be classified.

2.6 CAPT Systems: State of the art

Computer-Aided Pronunciation Training (CAPT) systems, have improved over time, as well the field has kept in constant activity [53, 10]. Since the first introduction of the Computer-Aided systems for language teaching, it was presented in early stages as Computer-Aided Language Learning (CALL) systems in the 1960s [53], taking the first approach in the medical field for automatic display and detection for the deaf and hard of hearing in Children [54]. It was until 1972 [10] where the first modern CAPT system was implemented, and later on the 1990-2000 period improvements on the general technologies were presented using probabilistic methods [27] with remarkable works as [55] Where legacy Hidden Markov Model (HMM) Log-Likelihood ratio algorithm was proposed and later on [56] improvements over HMM Log-Likelihood were proposed, in the same line, on[57, 58] Goodness of Pronunciation was proposed and became the defacto probabilistic method used before the Supervised Learning Algorithms were developed [59].

As stated before, with the probabilistic methods based on GOP and HMM Log-Likelihood the general idea was to use a template based on predefined or goal pronunciation. This kind of method had been explored and had the drawback of not being able to correctly identify which pronunciation error was presented and for that reason, unable to provide accurate feedback to the user about how to improve a particular pronunciation error. [27]

Since the advances in AI algorithms with the introduction of Support Vector Machines, a great improvement was applied to the field [27, 43]. Related works (papers) had addressed the pronunciation error detection. In the same way other classical algorithms have been applied as Random Forest, K-Nearest Neighbors, and Multilayer Perceptron. These kinds of algorithms

have reached ranges of 83% to 85% of accuracy for similar tasks [49], and for modern CAPT systems, the accuracy of 86% is expected [10]. For this reason, the Supervised Algorithms had been considered state of the art. However, in recent years, with the improvement and continuous research in Deep Learning, a huge number of papers have been released lately where Deep Learning Architectures are trained and tested against complicated datasets . Architectures as Recurrent Neural Networks, Convolutional Neural Networks (CNN), and CNN with Transfer Learning approaches are among the principal variations available today [59, 42, 60, 61, 62] which allows passing the 90% accuracy mark. Still, the use of any form of Neural Network is prone to not to be able to surpass Classical Artificial Intelligence Algorithms if a curated dataset is not present or the Dataset is not balanced. [61, 63]

Nowadays, Supervised Learning Techniques as Support Vector Machine have proven to provide solid results when a huge dataset is not available, or Deep Learning is not suitable. Despite that inconvenience, the use of a technique called Data Augmentation has been applied to increase the dataset size when a big dataset is not present[64]. This has been explored in [62]; however the results obtained and the gains are not much noticeable becoming the training time and computational power needed a barrier. Different approach as Transfer Learning has been explored with great results, but these approaches still need a huge dataset. [42]

In conclusion, even the development of the field is moving toward Deep Learning techniques, a Baseline algorithm based on Classical Supervised Learning Techniques is still valid, providing solid results comparing against Neural Algorithms. For that reason, SVM has proven solid results reaching about 86% of accuracy, remarkable for small datasets and as a baseline algorithm. [49, 10]

Chapter 3

Method

3.1 Introduction

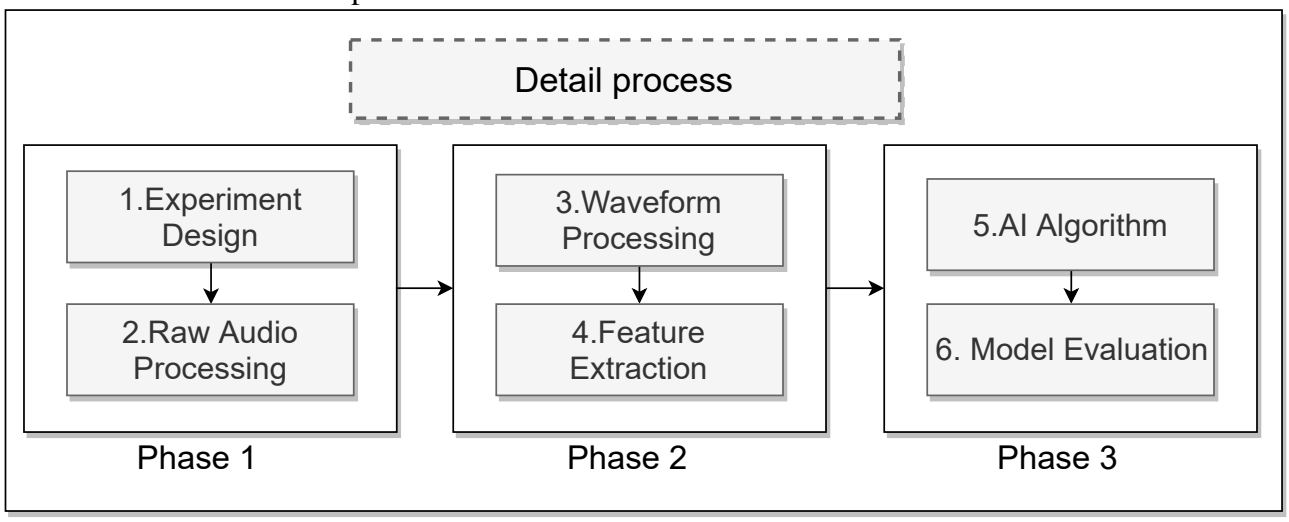
In this section, a general explanation of the whole process will be presented, since the data gathering process to the Algorithm design and implementation. This information should be enough to clarify the project structure and be able to replicate the process.

The general structure is depicted on 3.1. As showed there the general is composed of three general phases

1. Data Gathering.- This phase includes the Design of the Experiment to get raw audios from the volunteers. As well, the storage and label classification for the Raw Audio Files gathered
2. Feature Extraction.- In this phase the waveform processing came through. This phase includes all the modifications to the desired format, as well as the feature Extraction to let it ready for the AI algorithm to train.
3. Training and Modeling.- In the final processing step, the information generated in Phase 2 is consumed by the AI algorithms. At this point, the training, tunning and analysis of results is performed.

The general idea is to have a reproducible methodology that can be applied to other

Figure 3.1: General process used in this project. The system has basically 3 main phases where the information was processed.



similar pronunciation errors. This particular approach is focused on the pronunciation errors that occur at the beginning of the sentence. As it will be shown in Phase 2, some design considerations were applied to accomplish this and make easier for the AI algorithms to discern between the category classes.

3.1.1 Phase 1: Data Gathering and Preprocessing

The general idea of Phase 1 was to gather the data and deal with the storage of the data. This was a critical part of the project as the design of the experiment, and the correct storage of the information was essential to guarantee success in the following phases.

Experiment Design

The first phase of this work was focused on the previous research done in Chapter 2. The main idea behind this Phase was to design an experiment that will allow the AI Algorithm to get the desired features. In order to achieve this, a list of audios was elaborated, the key points to consider for the elaboration of the list were as follow:

- Word Selection: The list should contain enough words that contained the S-impure class

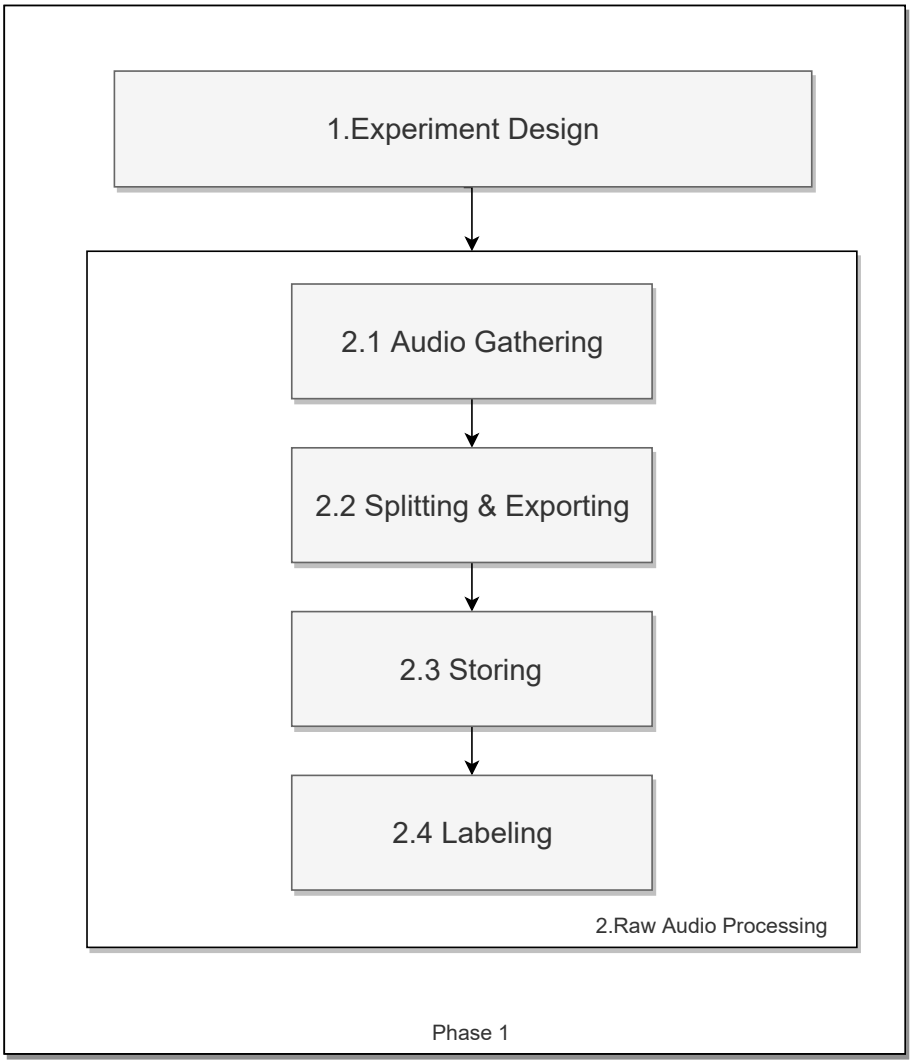


Figure 3.2: Phase 1. Experimental design and raw audio processing detail

along with neutral words.

- Length: The length of the words selected should be variable; this was guaranteed by the inclusion of different syllables.
- Random: A random order of the words was essential to avoid a repetitive behavior from the volunteers. As well the volunteers were not told about the purpose of this experiment.

The word selection consisted of the following design criteria:

- S-words: A total of 40 words were considered
- Random Words: A total of 40 words were considered.

- E words: A total of 20 words were considered.

The E words were considered as Dummy words. The main purpose was to ensure to induce the volunteers the pronunciation error needed to catch without them to knowing it.

In the same way, to take this project one step further, real audio data was considered and desired; this includes audio with background noise and non-studio quality for recordings. For that reason, no specific place to take the recordings was selected.

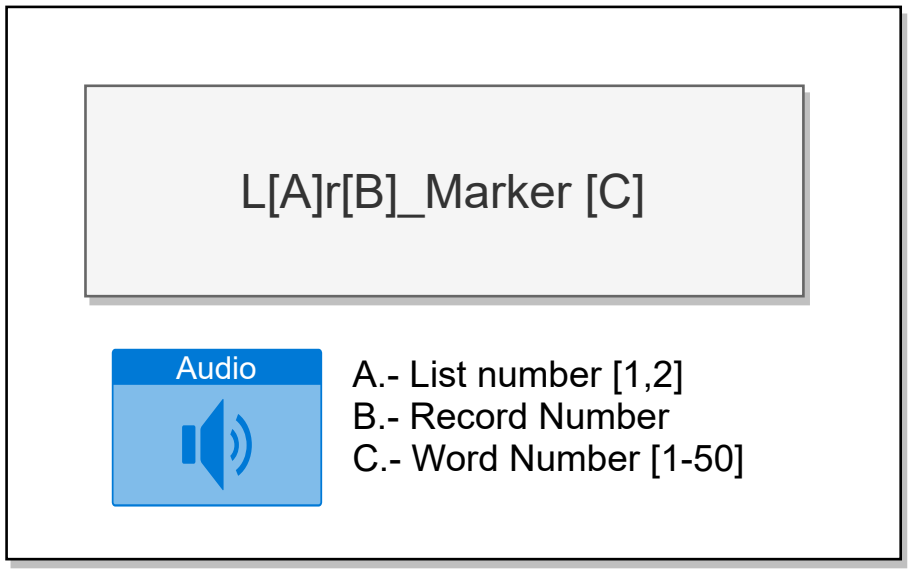
Another important point that was considered was to avoid causing tiredness of the volunteers. for that reason, a list of 100 words could take some time, and the effort of the volunteers in the last part could be a problem; for that reason a split of two lists was considered where 50% of each class of words was ensured to be included.

Raw Audio Processing

One key part of the success of the project was to correctly handle the audio acquisition, define the way to clean, split and store the files gathered to be able to classify them and refer to them precisely. For that reason, the following points where considered

- Audio Gathering.- In this part, the recording of the samples was performed.
- Conversion.- The conversion of the raw files from its source format to a lossless and uncompressed format.
- Splitting.- At the moment the audios are gathered, a splitting process is needed. In this part, the audios are constrained to gather only the waveform of the word, discarding the silences.
- Storing.- After the audios are split in a word by word basis, a correct way to refer to them is needed. The unique ID was considering using a Schema depicted in figure 3.3
- Labeling.- Once the files were stored three judges were selected to label the audios. This complete process is detailed in the next Chapter in Section 4.1.4

Figure 3.3: Unique Audio Record Identifier System



3.1.2 Phase 2: Waveform Processing and Feature Extraction

At this point, the Audios were ready to process by Python. The main goal desired was to obtain an output ready to train by an AI algorithm. During this process the audio length was normalized and left-aligned before performing the feature extraction. The graphical overview of the whole process is depicted in figure 3.4.

Silence Insertion

As part of the waveform processing, the length of the data was normalized to the largest data available in all the recordings. To do that a silence was inserted to tall the audios at the end of the recording.

Audio Shifting

The signal with data was shifted to the left. The objective of this Waveform transform was to enforce and assure that the pronunciation error will always be at the beginning of the waveform. This approach could also be useful for pronunciation errors present at the end of a word; where the audio will be shifted right.

Feature Extraction

Once the waveform was modified with the silence insertions and shifting, the feature vector is generated. The feature vector extraction is based on [21] work, considering MFCC, ZCR, RMSE, and SF as the main features.

Storage

During the process, all the data was stored in a Pandas DataFrame object. This data structure can still be useful to perform the rest of the operations and given as input to the Phase 3 of the process; however, a cleaner way is to extract the features and labels from the DataFrame in their raw Numpy Array form (.npz) format. The npz format has the advantages of smaller size and faster processing as it is based entirely on Numpy, which is faster than the Pandas Dataframe object.

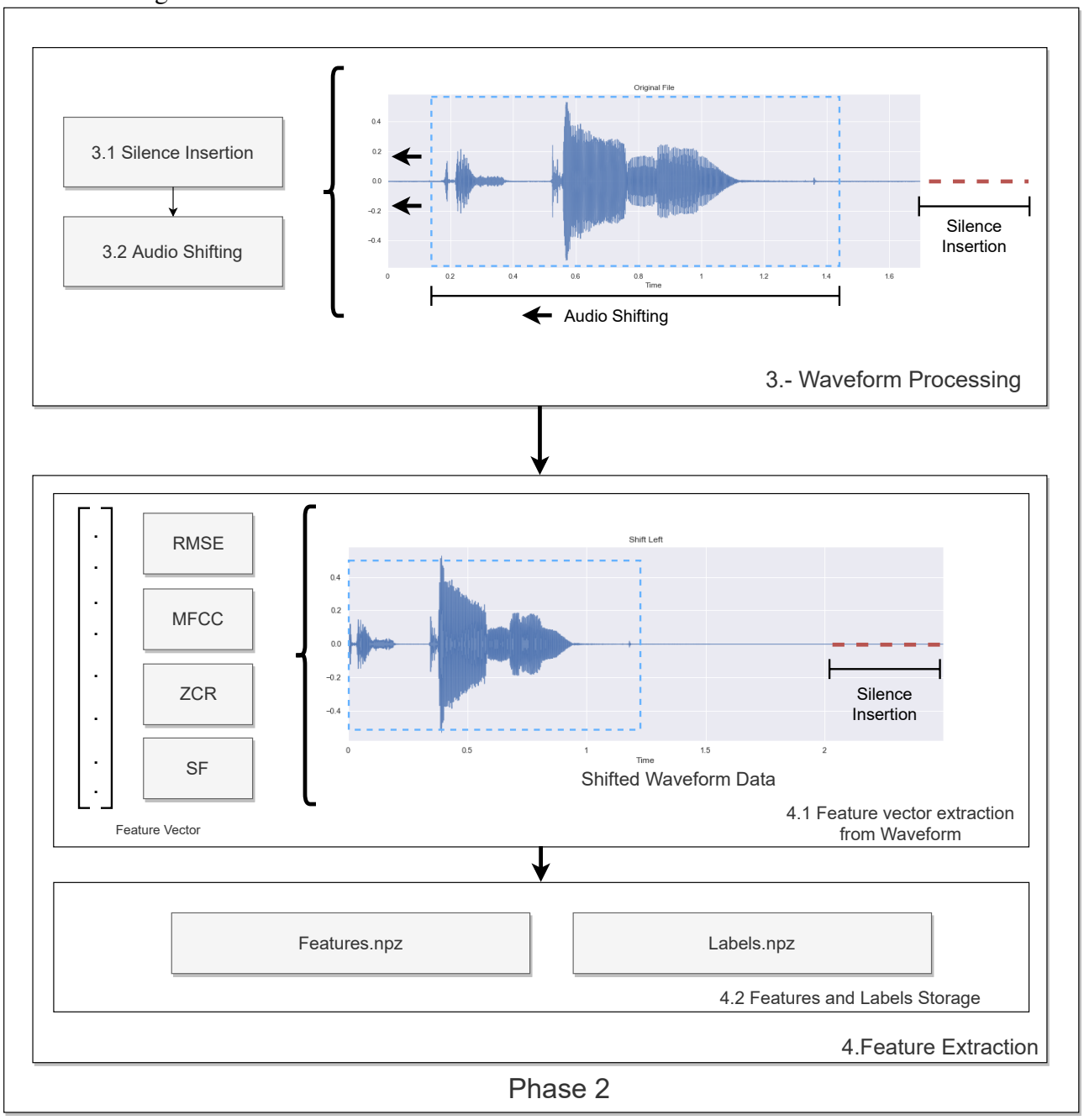
3.1.3 Phase 3: Algorithm Training and Model Evaluation

In this final phase, the training process was performed in three different classical Supervised Machine Learning Algorithms: Support Vector Machine (SVM), Random Forest (RF), and K-Nearest Neighbors (KNN). The general Graphical Overview of the complete phase is depicted in Figure 3.6. This includes points 5 and 6 from the general process in Fig. 3.1 as well a detail of the Model Evaluation part.

Input

The input to the system was the feature vector obtained on the Phase 2 stage. The features and labels are required to be in a high-performance data structure as npz format or Pandas Dataframe Data Structure to transform to Numpy Arrays.

Figure 3.4: Waveform processing, Feature Vector Extraction and High performance data structure storage.



AI Algorithm

The AI Algorithm could be visualized as a four-stage process. This selection is presented in Figure 3.5 and the detailed three stages are described below:

Stage 1: Split into Training and Testing Set

For this stage, the split process is performed. It is essential to balance the testing and training set to allow enough data on the Testing set while keeping essential information for training. For this project, 70% Training and 30% Testing sets were selected, as this is common in literature.

Stage 2: Scaling and Normalization

The normalization of the feature vectors is a standard process today in Machine Learning; in particular for the Algorithms used in this work that is not the exception. [65, 66, 41, 67] The normalization was achieved using StandardScaler from Scikit-Learn library.

Stage 3: AI Algorithm Selection and Hyperparameter selection

Given the proposed structure, in this Stage basically all the classical Supervised learning algorithms can be applied. For this project, three algorithms were considered: SVM, RF, and KNN.

SVM This classificator choice was driven by several works that include the use of them for this particular field of phoneme error detection for CAPT systems [63, 44, 43, 49, 68]. For Support Vector Machine, the Regularization Parameter (C), Gamma, and Kernel [41] were applied for Hyperparameter Grid-Search Tunning.

RF The use of Random Forest has been explored in this field as well [63, 49]. The Number of Estimators and Minimum Samples Leaf Hyperparameters were considered for Grid-Search Tunning. [69, 47, 70]

KNN In the case of K-Nearest Neighbors, it has been used for Phoneme classification [49, 68] and Automatic Speech Recognition (ASR) [71, 72]. Number of Neighbors, Leaf Size, Weight, and Algorithm, were the hyperparameters considered for Grid-Search Tunning. [73]

Stage 4: Training phase with Grid Search

The goal of this stage is to get the Output Model ready for Model Validation. For this process Grid search was selected. Grid Search is a high CPU consuming algorithm that aims to try all the possible combinations of the Hyperparameters and has shown important improvements in parameter tuning [74].

The grid search was performed using a Cross-Validation function from the Scikit-Learn library using a Stratified KFold of 5 for Cross-Validation parameter leaving 80% Training and 20% validation for each iteration. K=5 was chosen as a trade-off between speed and Variance reduction. [75]

Model Evaluation

As the final step, the Model Evaluation was performed using the following parameters:

- F1 Score
- Specificity
- Sensitivity
- Recall
- Precision
- Accuracy
- Learning Curve

Accuracy was considered as the key performance metric due to the nature of the problem, and a Learning Curve was obtained to evaluate the performance as the number of samples was introduced. The Learning Curve was a key process to evaluate the model and the experiments, as it will show if the dataset collected was enough to train the classifier.

Figure 3.5: AI algorithm detailed Stages

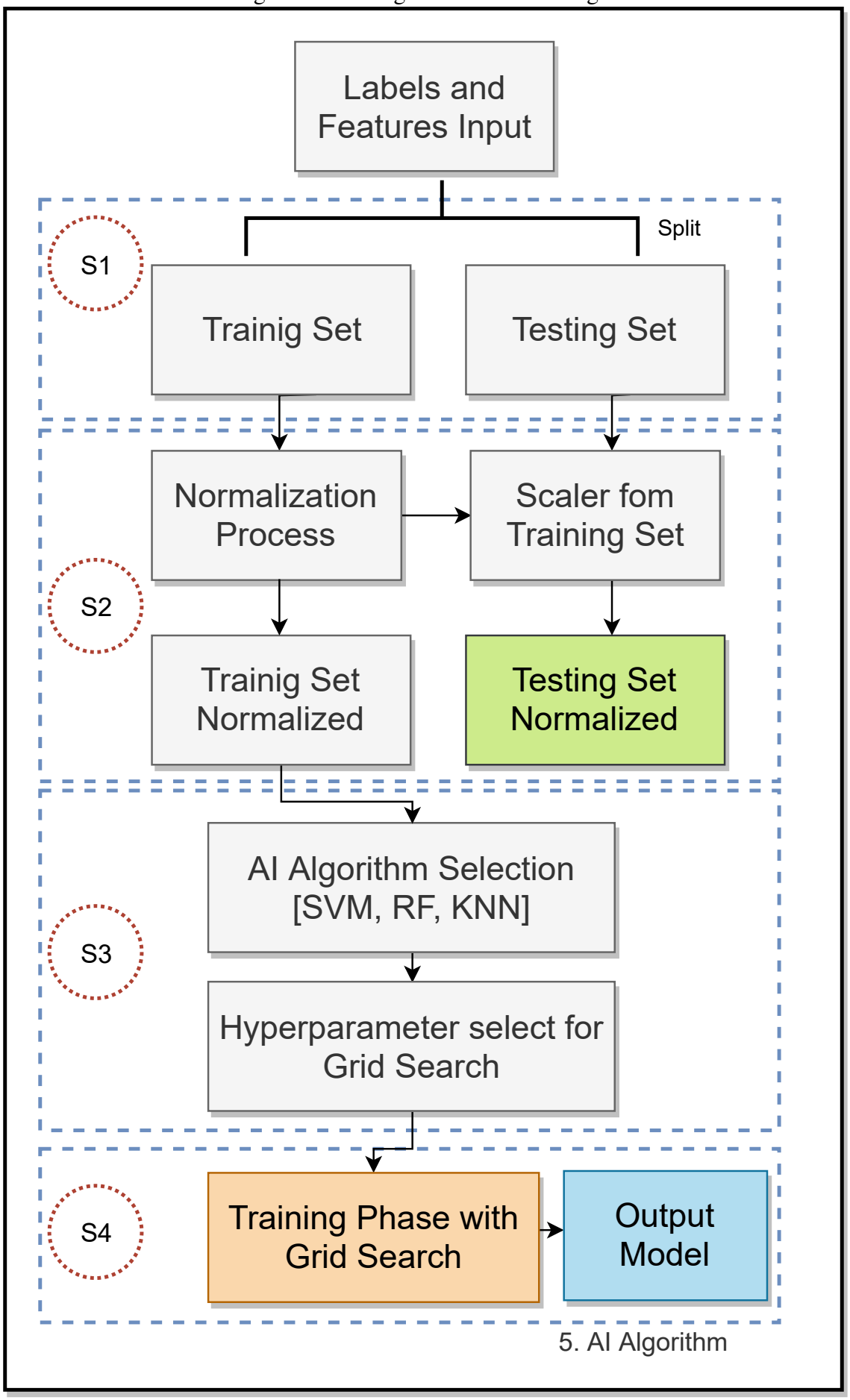
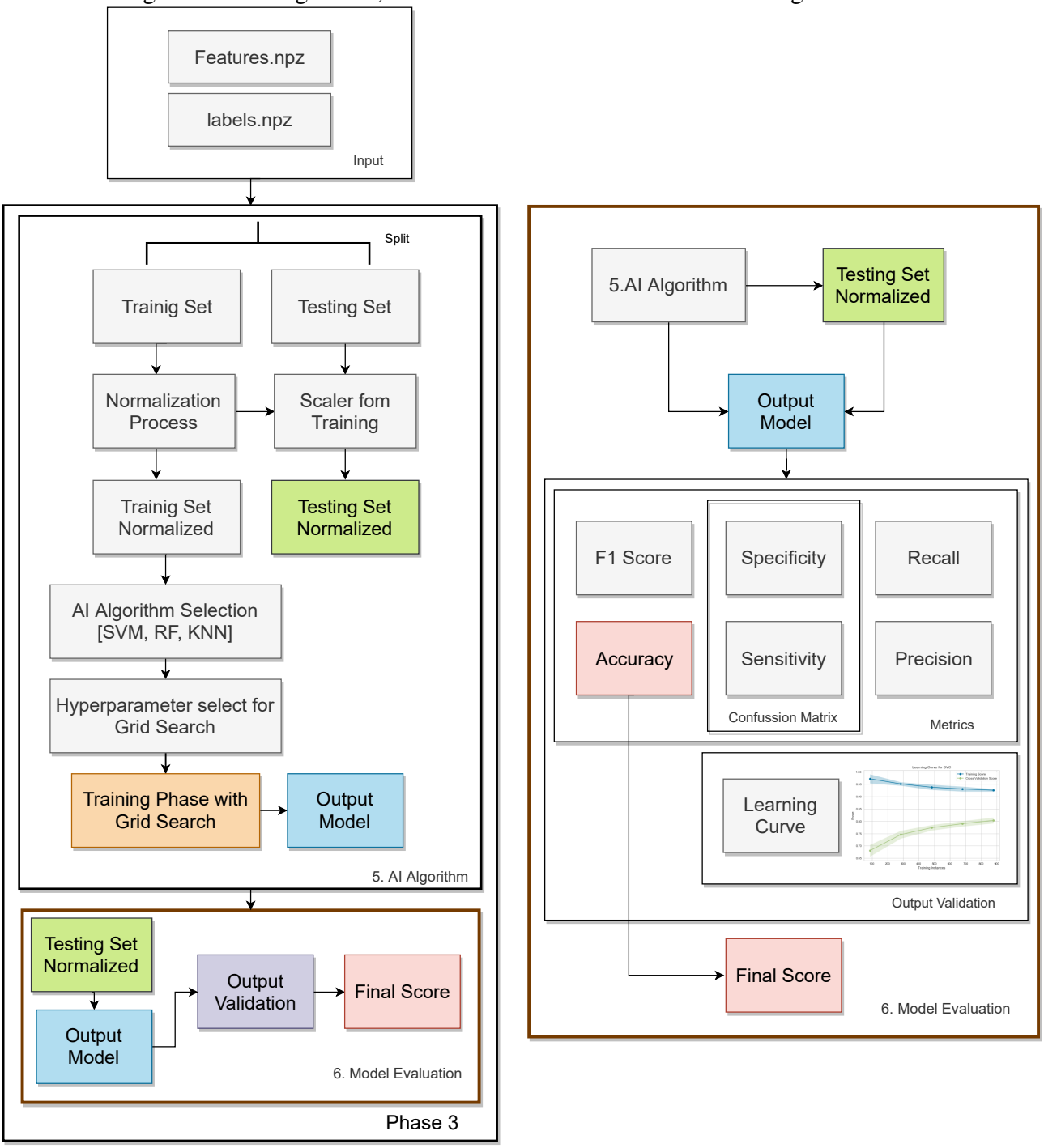


Figure 3.6: AI algorithm, Validation metrics and Final Score Diagram



Chapter 4

Detail of Experiments

4.1 Introduction

In the present chapter, a detailed description of the experiments performed using the method proposed on the previous chapter is shown; this includes a detail of the audios gathered, the algorithms tested as well as the reasoning related to the design choices of experiments along with the results obtained. In chapter 5 the discussions of the results will be presented.

4.1.1 Audio gathering: Database

One of the most significant problems when working with audio to train a classifier, especially when dealing with supervised algorithms, is the lack of data, specially labeled data. Regarding this project, was not the exception. Therefore, a data collection experiment was needed to design the labeled database of the S-impure words.

In order to select the words, a list of works where the error was expected to be found along with random S-words and random words. The words chosen for this experiment are presented in table 4.1

Table 4.1: List of words choosen for this experiment

Count	**S-impure**	**Random**	**E-Words**
1	Stitch	Word	Essential
2	Scale	Zero	Expression
3	Scaling	Return	Exclude
4	Scared	Begin	Exactly
5	School	Match	Expected
6	Screw	Would	Excel
7	Scramble	Like	Embarrass
8	Screams	Cellphone	Emergency
9	Smash	Computer	Escalate
10	Scope	Just	Escape
11	Skiing	Year	Essence
12	Sleep	Person	Estimate
13	Sliding	Now	Estheticians
14	Sloping	Think	Establish
15	Slow	Well	Estranged
16	Smile	Way	Escort
17	Snow	Because	Espionage
18	Spanish	Most	Esteem
19	Speak	About	Escrow
20	Speaking	Eyes	Espresso
21	Special	Company	
22	Specially	Child	
23	Stupid	Public	
24	Squeeze	Number	
25	Spell	One	
26	Spicy	Take	
27	Split	Visit	
28	State	Name	
29	Stuck	Build	
30	Skip	Protection	
31	Spam	Same	
32	Spirit	See	
33	Steam	Saw	
34	Start	Scientists	
35	Scalar	Scissors	
36	Small	Should	
37	Spouse	Sister	
38	Scam	Sounds	
39	Scar	Ship	
40	Strip	Seek	

Table 4.2: L1 and L2 lists generated with the random choice function

No	L1		L2	
1	Scale	Would	Name	Seek
2	State	Take	Smile	Espionage
3	Spell	Year	Expected	Squeeze
4	Spirit	Protection	Sister	Ship
5	Spanish	Visit	Excel	Company
6	Specially	Match	Because	Smash
7	Screw	Zero	Slow	About
8	Scared	Child	Split	Special
9	Scam	Scientists	Exactly	Estimate
10	Steam	Like	Small	Should
11	Sliding	Build	Embarrass	Escrow
12	Sloping	Well	Same	Scramble
13	Speak	Sounds	Now	Scissors
14	Speaking	Saw	Way	Skip
15	Screams	Public	Computer	Cellphone
16	Spicy	Esteem	Scalar	Stitch
17	Spouse	Escort	See	Start
18	School	Emergency	Strip	Just
19	Scaling	Estheticians	Person	Estranged
20	Scar	Essential	Skiing	One
21	Return	Expression	Stupid	Sleep
22	Most	Escape	Espresso	Number
23	Eyes	Exclude	Scope	Spam
24	Begin	Escalate	Stuck	Establish
25	Word	Essence	Snow	Think

The experiment was designed in this way to avoid that sample was biased. As well E words were also introduced to include representative data of similar sound.

In the following step, this list was split to make a short session with the person, and make it easier to record to the volunteers. To evenly distribute the words across the two lists, the complete list was registered as an array, and numpy's random.choice function was used to get the same amount of words of each group in the lists. Under this approach, the two lists L1 and L2 were generated as depicted in Table 4.2.

4.1.2 Data gathering

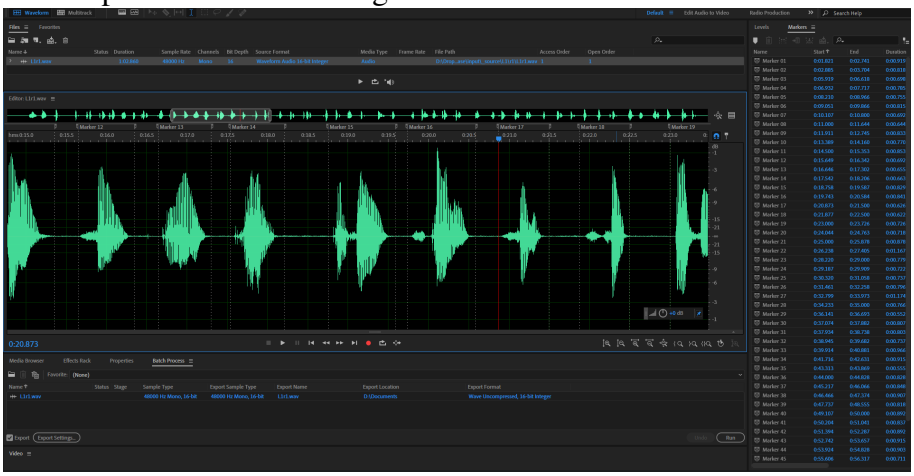
For this part of the project, the goal was to obtain about 100 audio files of both lists, to have nearly 2000 audio files with S-words. To achieve that, once the two lists of the words where shuffled and get ready to distribute, the data acquisition process begun, the targets were males and females from ages 15-40 years old that had taken some English classes before. The list of words was presented to them, and we collected the information via direct recording with a cellphone. On rare occasions the recording was capable of being recorded in place; virtually all the recordings were recorded in the home of the persons and shared via WhatsApp or electronic means. The only requirement was that the record was taken in a quiet place with the minimum noise possible.

4.1.3 Audio processing

Once the audios where gathered, they were converted to a suitable format. There is a wide availability of audio formats to choose from. In this project, choosing the correct audio format was needed, and some important criteria were required to take into consideration. For that reason a review of the available supported formats was done. As explained in Chapter 2, the WAV format was ideal for this case, as it is an uncompressed and lossless format. An additional advantage is the native support of WAV files in Python. The conversion of the raw files to WAV format was achieved using Pydub and Librosa libraries. In this step the Audios were converted to Mono Channel [21, 49] with Adobe Audition and standard Adobe Audition noise filtering was applied only when the noise levels were very high.

The split of the data was achieved using Adobe Audition Software, where the word was isolated at the beginning and at the end of the Waveform related to the word. An example of this is shown on Figure 4.1.Along with the splitting a pre-processing step was performed which include: Noise-canceling and Sound normalization to 6 dB only to those that don't have enough sound and splitting.

Figure 4.1: Split of the words using Mono Audio Waveform with Adobe Audition



Once the audios were correctly splitted they were exported. The output selection was set at 48KHz, Mono Audio, 16-bit uncompressed WAV format for all the files. The next step was to classify the data. For that purpose a hierarchical folder structure was selected, as presented on Figure 3.3. The idea was to have an audio tag that allow quick referencing for processing and labeling by the judges to identify if the file has or not a pronunciation error.

4.1.4 Classification

The labeling process was done by the individual judges, where the possible choices were the following:

- Label “e”.- The audio record has a s-impure error and has a good quality of sound.
- Label “s”.- The audio is correctly pronounced and has a good quality of sound.
- Label “n”.- The word has good quality, but the S sound is not present.
- Label “r”.- The word should be removed due poor audio or non-understandable sound.

Given these criteria, the results of the labels once the algorithm was implemented is presented on table 4.3.

Based on these criteria, the general algorithm of feature extraction and classification could be performed using one of the following voting systems to get the final label for each

Table 4.3: Results from the Judges

Class	J1	J2	J3
e	804	816	881
s	926	995	1031
n	57	55	41
r	166	87	0

Table 4.4: Result of the voting algorithms

Class	Type_remove	Type_vote	Type_vote_remove
s	845	994	936
e	732	817	796
n	41	55	55
r	335	87	166

audio:

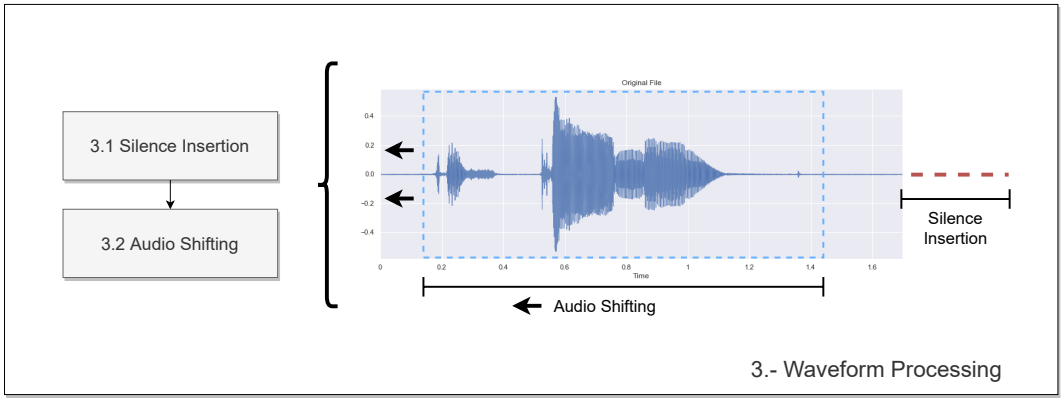
- Voting system.- The audios that have the majority of the information, i.e. two of the three judges qualify a label. If there is consense these labels get chosen; otherwise it will remove the audio.
- Voting system with removal.- Only when the three judges coincide, the audio is selected. Otherwise, the audio is removed.
- Voting system with exclussion.- This voting method will look for the "r" key, this means, if one of the judges thought that the audio was not useful, this automatically deletes the entry. Otherwise, the Voting System is implemented.

The classification results obtained are depicted on Table 4.4. Those labels were used to feed the supervised learning algorithms. A performance training of the three classificators was tested leading that the voting system with removal lead to the best results, for that reason the training with Grid-Search was performed using those labels.

4.1.5 Feature Extraction

Once the classification was completed, and the audios were stored in a hierarchical manner and WAV Format the entries were ready to process and get a feature extraction. To achieve this a Pandas Dataframe with key values as file path, Audio ID, number or record, class was

Figure 4.2: Detail of Waveform Processing



constructed scanning the folder.

Once all the files were stored in a master Pandas DataFrame, a processing loop was applied to all the audio files. This processing included an audio alignment using Librosa. The objective of this process was to keep all audio waveforms aligned to the left. To reinforce the location of the error desired to locate.

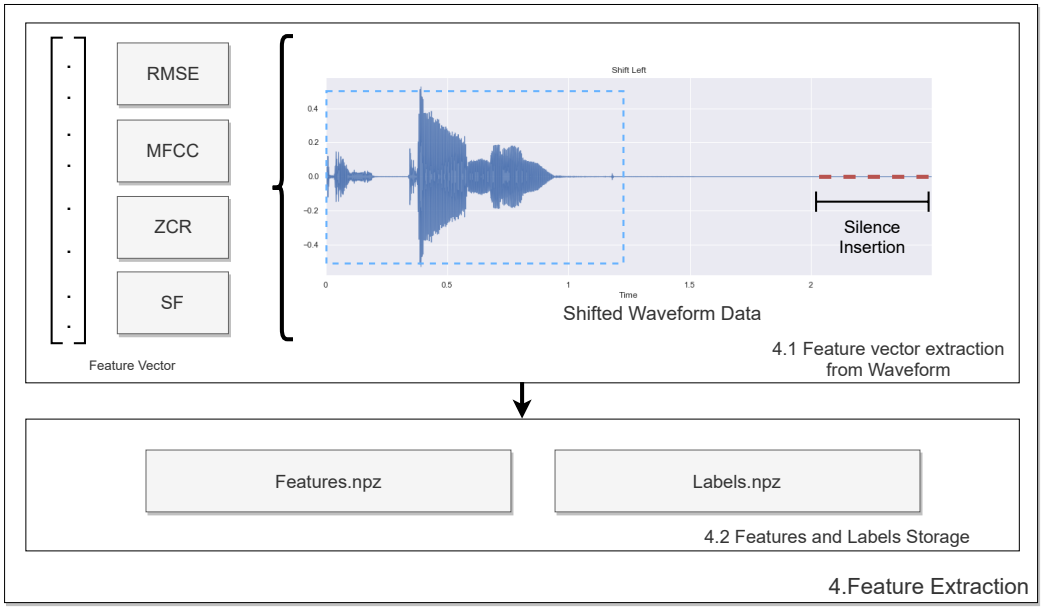
Once the alignment was completed, additional data processing was needed; in order to have all audios of the same length, librosa was used to determine the file with the greatest audio length and then the other audios were processed with silence insertion to fill the gaps. A general overview of the process can be observed in figure 4.2.

Finally the feature extraction was applied along with the storing of the data. The feature extraction phase included MFCC, RMSE, Spectral Flux and ZCR as features. The selection of these features was based on previous work by Avalinguo Team [21] and it is depicted on Figure 4.3.

4.1.6 Supervised Learning and Tunning

The key part of this work was to choose and tune the algorithms once the labels were defined. To achieve this result, Support Vector Machine (SVM) and Random Forest (RF) algorithms were used as proposed on previous works [21, 63, 44, 43, 49, 68] as well KNN algorithm

Figure 4.3: Feature Extraction process



has some interesting approaches with solid results [49] and widely used on Automatic Speech Recognition [72].

The training of the algorithms was performed on Sckit-Learn package and fed using the input data from the previous steps. In order to get the best possible parameters, a grid search was applied to the three algorithms varying the criteria of selection as discussed on the voting system section.

Regarding the process, all the algorithms were tested using Cross-Validation of 5, and inside the grid search a new final test and training data were generated with 70% of the data and the testing set was with 30% of the data. This approach was followed to always hide data for the final test despite the Cross-validation.

Support Vector Machine: SVM

The Support Vector Machine Algorithm was the algorithm that had the most expectations of the work due its previous uses in similar works. To make sure a good parameter tunning was achieved, the grid search was applied using three key parts of the SVM Algorithm:

- Regularization Parameter (C).- Main parameter to determine the degree of fit required

Table 4.5: Confusion Matrix of SVM

Pred\Truth	**Error**	**Good**
Error	194	33
Good	38	175

Table 4.6: Classification Report of SVM

Class	**Precision**	**Recall**	**F1 Score**	**Support**
S	0.8551	0.831	0.8429	213
E	0.8455	0.8678	0.8565	227

- Gamma.- Used to control the influence of new features.
- Kernel.- Fundamental to determine and find the decision boundary.

The search begun with the following parameters for each:

- “Voting”: [Voting witouth removal, Voting with removal, Voting with exclusion]
- ”C”: [10000, 8000, 7000, 5000, 25000, 10000, 1000, 500, 10]
- ”gamma”: [.00008, .00007, .00006, .00005, .00002, .0005, .0001, .001, .01]
- ”kernel”: [’poly’, ’sigmoid’, ’rbf’]

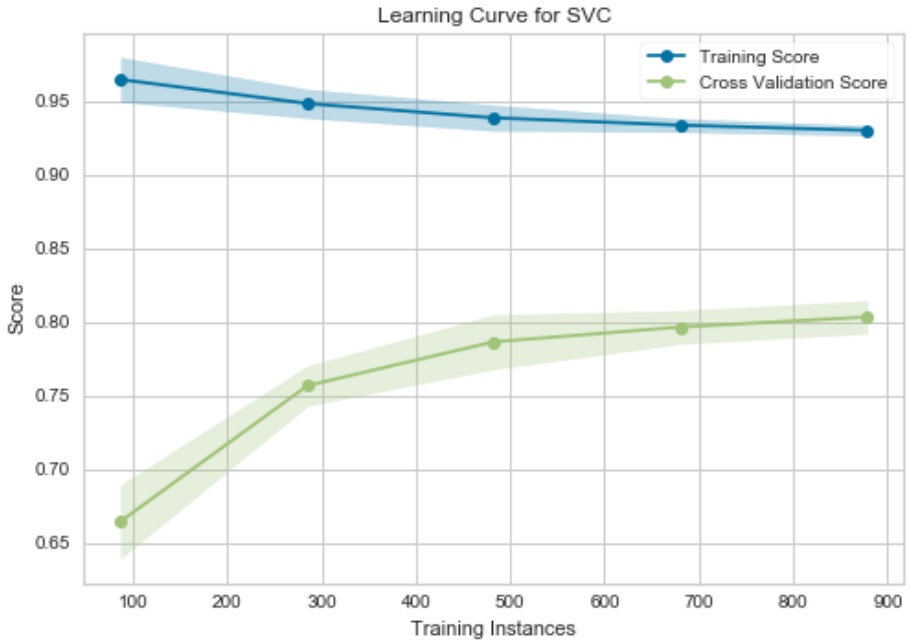
Based on those initial parameters the results lead to present the best accuracy with the following values:

- C=1.41
- Gamma = 0.052
- Kernel = rbf
- Voting = Voting with removal

These parameters were fed to the default SVM class object on Sckit-learn leads to the following confusion Matrix and Classification Report:

As showed in the confusion matrix in Table 4.5, the data presented was balanced. This distribution of the data clarifies if additional processing is needed to try to get the missing entries; however, in this case, no extra measures should be taken by the errors type 1 and 2, as the solution looked balanced.

Figure 4.4: Learning Curve for SVM



Finally a learning curve as processed via Cross-validation, to determine if further data was needed

The learning curve shows the classical funnel shape, this means that the algorithm has basically reached the limit on the training accuracy. For this reason, further data augmentation techniques or extra data collection was not considered. The resulting Learning curve is shown in Figure 4.4.

Random Forest: RF

Following the same process as SVM, a set of Random Forest Experiments were realized. In order to complete them the initial hyperparameters to tune with Grid-Search the below were considered:

- Number of Estimators.- Usually this parameter is set to a high number, for performance reasons this was considered.
- Minimun Samples Leaf.- This parameter determines how much of the samples are

Table 4.7:

Pred\Truth	**Error**	**Good**
Error	175	36
Good	48	181

Table 4.8: RF Classification Report

Class	**Precision**	**Recall**	**F1 Score**	**Support**
S	0.8341	0.7904	0.8117	229
E	0.7848	0.8294	0.8065	211

needed to generate to have a leaf node.

The search begun with the following parameters for each:

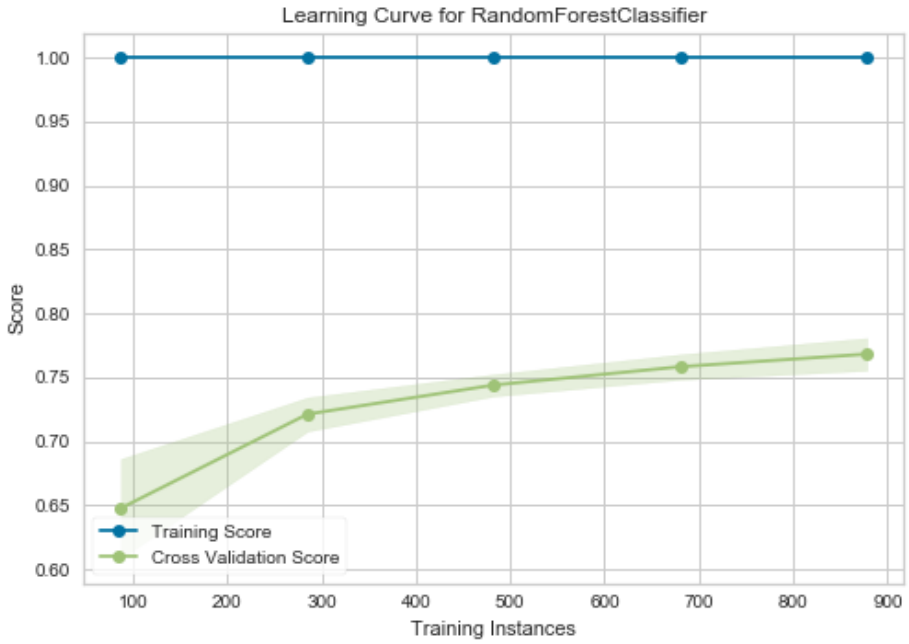
- “Voting”: [Voting witouth removal, Voting with removal, Voting with exclusion]
- ”Number of estimators”: [10,20,100,500,2000]
- ”Minimum Samples Leaf”: [5,10,20,30]

Based on those initial parameters the results lead to present the best accuracy with the following values:

- Voting = Voting with removal
- ”Number of estimators”: 2000
- ”Minimum Samples Leaf”: 2

The results presented show a balanced Confusion matrix, however the Random Forest Classifiers tends to classify better the S class of words.

Figure 4.5: Learning Curve for RF



The associated Learning curve is depicted on Figure 4.5. As it is shown, the best hyperparameters lead to a classification of individual examples per leaf, this behavior forms the curve shown. It is important to notice that despite of that the increase rate of the Cross Validation accuracy has not shown signs of significant increase passing 700 samples. This showed that extra data is not needed and that the algorithm is learning.

K-Nearest Neighbors: KNN

Finally, the Experiments with K-Nearest Neighbors were performed. For these experiments the following hyperparameters were considered:

- Number of Neighbors.- Used to get the number of neighbors to get the distances.
- Weight.- Used to determine the effect of the neighbors over the distance.
- Algorithm.- Algorithm employed to compute the nearest neighbors.
- Leaf Size.- Hyperparameter for the BallTree and KDTree algorithm.

Table 4.9: Confusion Matrix for KNN

Pred\Truth	**Error**	**Good**
Error	174	35
Good	42	189

Table 4.10: Classification report for KNN

Class	**Precision**	**Recall**	**F1 Score**	**Support**
S	0.8438	0.8182	0.8308	231
E	0.8056	0.8325	0.8188	209

The search begun with the following parameters for each:

- "Voting": [Voting witouth removal, Voting with removal, Voting with exclusion]
- "Number of Neighbors": [2,3,5,7,10]
- "Weight": ["uniform", "distance"]
- "Algorithm": ["ball_tree", "kd_tree"]
- "Leaf Size": [10,30, 50, 70, 100]

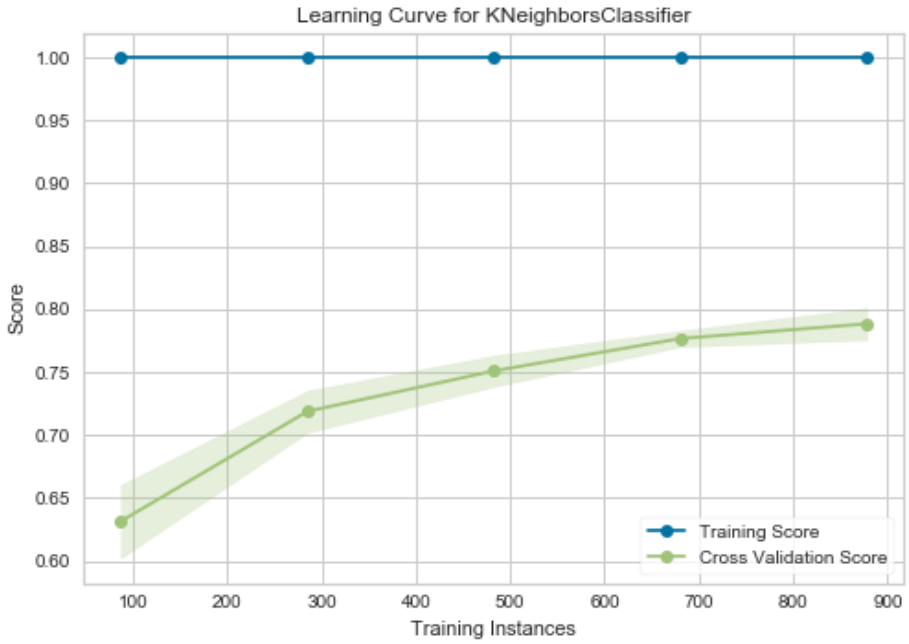
Based on those initial parameters the results lead to present the best accuracy with the following values:

- Voting = Voting with removal
- "Number of Neighbors": 5
- "Weight": distance
- "Algorithm": ball_tree
- "Leaf Size": 1

As can be shown on the Confusion Matrix in tab 4.9, the classification of the entries are balanced as well as SVM, this confirm that there is no particular predilection from the algorithm towards predict a particular class. Classification report showed and improved precision from class S over class E.

The related Learning curve for KNN is depicted on Figure 4.6. As it is shown, the hyperparameters lead to a perfect accuracy on training set. This kind of behavior is typical

Figure 4.6: Learning Curve for KNN



of this algorithm for training instances. Regarding the accuracy score for Cross Validation entries, can be shown that the slope is reducing passing the 700 samples as well as the SVM, and RF learning curves. This information confirms that extra data is not going to provide significant accuracy improvement.

Chapter 5

Conclusion and Future Work

5.0.1 Review

In this work, the use of SVM, RF and KNN algorithms was explored in a real data environment. The methodology to clean, organize, store and process audio files was implemented in the complete dataset providing an approach that can be used as a baseline for future related work.

As a result of the comparison of performance from the AI Algorithms applied in this work, the result of 85% reflects the expected result from the hypothesis, this kind of performance compiles with the current state of the Art algorithms in this field, especially the classical approaches. This result is considered solid, given the performance metrics obtained in different similar works when non-neural algorithms are applied.

About the method employed for audio classification, the three judges method applied, and the different exploration of the combinations provided clear insight, the best approach is to discard noisy data. This statement opens the door for further exploration where more focus will be needed in the audit of the audio data files.

While the result of 85% accuracy obtained using SVM is below the top-performing

results using Deep Neural Networks variants, it is worth to mention that the result complies with the current state of the art in the field using Classical Machine Learning Algorithms [49, 10]. Furthermore, the method applied handles well the noise as the data obtained contained a moderated level of noise as well they were not studio record quality. These associated characteristics of the dataset give a taste of the real case scenario that will be present on the real application.

Grid Search for Hyperparameter tunning proved to be an excellent complement to the training process. This method for parameter tuning improved the overall accuracy obtained when compared with a trial-error approach.

Support Vector Machine (SVM) Algorithm proves to be the best performing of the classical algorithms in this work. These results are consistent with the results reported in [49, 42] where the results were about 85%.

5.0.2 Contributions

The proposed method of pronunciation error detection method for S-impure words turned out to be efficient under noisy data and a small dataset, providing solid accuracy results compared to those shown in literature for classical Supervised Learning Algorithms.

5.0.3 Limitations

The current work has some limitations, especially in the scope. As this project addressed S-impure words and has proven solid results, the method applied still needs to be tested for other pronunciation errors.

The noise reduction and audio normalization techniques were applied manually. An automatic waveform analysis and noise removal methods need to be developed to improve Phase 1 process.

The judges applied in this work, while skilled in English, were no native or professional teachers. The help of a professional Teacher or Native Judges would be desirable to reduce the noise on the audio classification and have more accurate labeling; this could lead to better shapes in the Learning Curves.

Regarding the bias that could exist due to the nationality of the judges, is worth to mention that as only one of them lives in an English speaking country but all of them were Mexicans, for this reason it likely that could exist a bias in the way to classify the audios that have the S-impure pronunciation error.

The data collected during the experiment was slightly less than 2000 samples. This small dataset avoids the testing of the current Deep Learning State of the Art Algorithms mentioned in chapter 2.6.

5.0.4 Future Work

In order to improve the accuracy of the model, a better handle of the subjective evaluation of the judges could help, as well as the gathering of more data. In the same way data augmentation and a neural-net approach could be worth to review, or at least be considered if the size of the dataset is increased. Another possibility is to add or remove features and compare the outcomes, as mentioned on [21, 76] the variations the vector length of MFCCs features could impact the accuracy obtained as well the inclusion or removing of features. Exploring the MFCCs along other features as Linear Predictive Coding (LPC), Local Discriminant Bases (LDB), Perceptual Linear Prediction (PLP) [49, 76] would also be desired.

Regarding the implement of this project into a production environment and diminish the judges bias, a more complex voting system could be beneficial; for example a system where the pronunciation is evaluated in a not binary score, i.e. a decimal scale (1 to 10) where “10” is native pronunciation and “1” non-understandable speech. Including more judges of different

socio-demographic segments and evaluate the pronunciation of the user knowing more about his target speech location could provide more robust results; this is because there could be different kinds of valid pronunciations depending on the region and culture of a particular place.

In the same way, the same method discussed in Chapter 03 could be applied to the identification of pronunciation errors at the beginning of the word as well to the end. An interesting approach would be to identify this pronunciation error when the expected phoneme is in the middle of the word, as this will lead to having an unaligned dataset. For a situation like this recent works using Convolutional Neural Networks try to address this problem. [62, 42] The reasoning with that is justified using the analogy of the MFCCs as if they were pixels on images. The Neural Net should be capable to identify the MFCCs patterns of that particular error as long as it is the common feature in all the audios. However, audio alignment to the center using SVM could be worthwhile to review and compare against other approaches.

Talking about improvements in speed regarding the manual processing of the audio files, a possibility to be considered is to minimize the manual work of classification from the judges for each file and do it instead to clusters, applying an unsupervised learning clustering algorithm like proposed in [77]. This way the goal would be to tune the classificators to identify key elements that contain pronunciation errors; however, the needed to apply the judge's criteria to evaluate the correct pronunciation could not be avoided.

In summary, the future works related to this project are composed principally by: Increasing the dataset size, add/remove features in the audio feature extraction step, test with Neural Network approaches, use professional judges for labeling, include judges of different socio-demographical sectors, experiment with unsupervised learning, and apply this method for other kinds of errors.

Figure 5.1: Corrections

# Corrección	Descripción	Cómo se atendió
	Rogelio Soto:	
1	Comentar si no existe un bias debido al origen de los jueces.	En la página 52, en la subsección de limitaciones se comenta esta bias
2	Poner nombres más descriptivos en las fases del método. Corregir en las páginas 26 29 y 30 como en el capítulo 4, la aportación es la metodología, hay que poner en el capítulo 3 página 25 más detalle.	Se realizaron los cambios en las paginas mencionadas. Se detalló la fase en el título.
	Roman de León:	
3	Checar si hay una comparación justa con otros trabajos, tomando en cuenta que no se tiene una misma lista de palabras que otros autores.	Se compara con trabajo similares en la sección de conclusiones. Capitulo 5.
4	Entre los features porqué no se consideró evaluar algún otro feature. Para trabajo futuro.	En la página 30 se justifica porqué se usaron los features que se usaro, pero no se excluye (trabajo futuro) que haya otros features posibles que lo hagan mejorar, esto se menciona en la pagina 52
5	Correcciones en el documento en PDF enviado.	Cada una de ellas se corrigió en la página correspondiente.
	Luis Avila:	
6	Varios errores ortográficos.	Se usó Grammarly para checar ortografía y construcción de frases
7	Comentarios sobre implementar en un ambiente de producción	Atendido en pag 53
8	Orden de referencias	Corregido
	Ramón Brena:	
9	Agregar comentarios sobre como mejorar la clasificación de los jueces	Atendido en pag 53

Bibliography

[1] Juan Manuel Muñoz Sánchez and Manuel Gilberto Orozco Arevalo. *Clúster: "S-Impura" en la pronunciación del idioma inglés en los estudiantes de la Universidad Central del Ecuador, de la Facultad de Filosofía, Letras y Ciencias de la Educación, de la carrera Plurilingüe de séptimo y octavos niveles*. 2018. URL: http://200.12.169.19:8080/handle/25000/15551 (visited on 10/22/2019).

[2] Donald Winford and Suzanne Romaine. "Language in Society: An Introduction to Sociolinguistics". In: *Language* 73.4 (1997), p. 844. ISSN: 00978507. DOI: 10.2307/417331. URL: https://books.google.com.mx/books/about/Language{_}in{_}Society.html?id=1QZXbCGIhvMC{\&}redir{_}esc=y.

[3] Stephanie Lindemann. "Who speaks "broken English"? US undergraduates' perceptions of non-native English". In: *International Journal of Applied Linguistics* 15.2 (2005), pp. 187–212. ISSN: 0802-6106. DOI: 10.1111/j.1473-4192.2005.00087.x. URL: http://doi.wiley.com/10.1111/j.1473-4192.2005.00087.x.

[4] Kamilla. Knopf. *English pronunciation exercises Sprachlaborkurs nach didakt. Schwerpunkten d. Ausgangssprache Deutsch*. Fink, 1975. ISBN: 3770512537.

[5] Bertus van Rooy, Sanet van Rooyen, and Herman van Wyk. "An Assessment of High School Pupils' Attitudes Towards the Pronunciation of Black South African English".

In: *South African Journal of Linguistics* 18.sup38 (2000), pp. 187–213. ISSN: 1011-8063. DOI: 10.1080/10118063.2000.9724571.

[6] Mario S. Di Bitetti and Julián A. Ferreras. "Publish (in English) or perish: The effect on citation rate of using languages other than English in scientific publications". In: *Ambio* 46.1 (2017), pp. 121–127. ISSN: 0044-7447. DOI: 10.1007/s13280-016-0820-7. URL: http://link.springer.com/10.1007/s13280-016-0820-7.

[7] Mohammod Moninoor Roshid, Susan Webb, and Raqib Chowdhury. "English as a Business Lingua Franca: A Discursive Analysis of Business E-Mails". In: *International Journal of Business Communication* (2018), p. 232948841880804. ISSN: 2329-4884. DOI: 10.1177/2329488418808040. URL: http://journals.sagepub.com/doi/10.1177/2329488418808040.

[8] David G. Drubin and Douglas R. Kellogg. "English as the universal language of science: Opportunities and challenges". In: *Molecular Biology of the Cell* 23.8 (2012), p. 1399. ISSN: 10591524. DOI: 10.1091/mbc.E12-02-0108.

[9] Stephen Pihlaja Philip Seargeant, Ann Hewings. *The Routledge Handbook of English Language Studies*. Routledge, 2018. DOI: 10.4324/9781351001724.

[10] Chesta Agarwal and Pinaki Chakraborty. "A review of tools and techniques for computer aided pronunciation training (CAPT) in English". In: *Education and Information Technologies* (2019). ISSN: 15737608. DOI: 10.1007/s10639-019-09955-7.

[11] Carmen Alicia Salazar Parreño. *ESTRATEGIA DIDÁCTICA PARA LA PRONUNCIACIÓN DE LOS SONIDOS DEL INGLÉS AMERICANO — UNIVERSITARIA: Docencia, Investigación e Innovación*. 2013. URL: http://revistas.udenar.edu.co/index.php/duniversitaria/article/view/583 (visited on 11/07/2019).

[12] Nancy F. Chen and Haizhou Li. "Computer-assisted pronunciation training: From pronunciation scoring towards spoken language learning". In: *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA 2016*.

Institute of Electrical and Electronics Engineers Inc., 2017. ISBN: 9789881476821. DOI: 10.1109/APSIPA.2016.7820782.

[13] Richeng Duan et al. "Articulatory modeling for pronunciation error detection without non-native training data based on DNN transfer learning". In: *IEICE Transactions on Information and Systems* E100D.9 (2017), pp. 2174–2182. ISSN: 17451361. DOI: 10.1587/transinf.2017EDP7019.

[14] Abbas Pourhosein Gilakjani and Ramin Rahimy. "Using computer-assisted pronunciation teaching (CAPT) in English pronunciation instruction: A study on the impact and the Teacher's role". In: *Education and Information Technologies* (2019). ISSN: 15737608. DOI: 10.1007/s10639-019-10009-1.

[15] Elsa Speaks. *ELSA - Speak English fluently, easily, confidently*. 2019. URL: https://elsaspeak.com/ (visited on 11/14/2019).

[16] Maxine Eskenazi. "Using automatic speech processing for foreign language pronunciation tutoring: Some issues and a prototype". In: *Language Learning and Technology* 2.2 (1998), pp. 62–76. ISSN: 10943501. URL: https://eric.ed.gov/?id=EJ577631.

[17] Islam Ababneh. "English Pronunciation Errors Made by Saudi Students". In: *European Scientific Journal, ESJ* 14.2 (2018), p. 244. ISSN: 18577881. DOI: 10.19044/esj.2018.v14n2p244.

[18] Johanne Myles and Liying Cheng. "The social and cultural life of non-native English speaking international graduate students at a Canadian university". In: *Journal of English for Academic Purposes* 2.3 (2003), pp. 247–263. ISSN: 14751585. DOI: 10.1016/S1475-1585(03)00028-6.

[19] Renata F.I. Meuter et al. "Overcoming language barriers in healthcare: A protocol for investigating safe and effective communication when patients or clinicians use a second language". In: *BMC Health Services Research* 15.1 (2015), p. 371. ISSN: 14726963.

DOI: 10.1186/s12913-015-1024-8. URL: http://bmchealthservres.biomedcentral.com/articles/10.1186/s12913-015-1024-8.

[20] Duolingo. *Duolingo*. URL: https://www.duolingo.com (visited on 11/19/2019).

[21] Alan Preciado-Grijalva and Ramon F. Brena. "Speaker Fluency Level Classification Using Machine Learning Techniques". In: (2018). arXiv: 1808.10556. URL: http://arxiv.org/abs/1808.10556.

[22] Robert Pasnau. "What is Sound?" In: *The Philosophical Quarterly* 49.196 (1999), pp. 309–324. ISSN: 0031-8094. DOI: 10.1111/1467-9213.00144. URL: https://academic.oup.com/pq/article-lookup/doi/10.1111/1467-9213.00144.

[23] ASA Acoustic Society of America. *ANSI/ASA S1.1 & S3.20 Standard Acoustical & Bioacoustical Terminology Database - Welcome to ASA Standards*. 2019. URL: https://asastandards.org/asa-standard-term-database/ (visited on 11/04/2019).

[24] Karlheinz Brandenburg. "MP3 and AAC Explained". In: *Audio Engineering Society, 17th International Conference 2004 October 26–29* (1999), pp. 99–110.

[25] Britannica. *Encyclopedia Britannica — Britannica.com*. 2019. URL: https://www.britannica.com/ (visited on 11/04/2019).

[26] IPA. *International Phonetic Association Full IPA Chart*. 2018. URL: https://www.internationalphoneticassociation.org/news/201805/2018-ipa-charts-now-posted-online (visited on 11/21/2019).

[27] Silke Maren Witt. "Automatic error detection in pronunciation training: Where we are and where we need to go". In: *Proceedings of the International Symposium on Automatic Detection of Errors in Pronunciation Training (IS ADEPT)* (2012), pp. 1–8. URL: https://www.researchgate.net/publication/250306074{_}Automatic{_}Error{_}Detection{_}in{_}Pronunciation{_}

}Training{_}Where{_}we{_}are{_}and{_}where{_}we{_}need{_}to{_}go.

[28] Emre Cakir et al. "Convolutional Recurrent Neural Networks for Polyphonic Sound Event Detection". In: *IEEE/ACM Transactions on Audio Speech and Language Processing* 25.6 (2017), pp. 1291–1303. ISSN: 23299290. DOI: 10.1109/TASLP.2017.2690575. arXiv: 1702.06286.

[29] Steven B. Davis and Paul Mermelstein. *Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences*. 1980. DOI: 10.1109/TASSP.1980.1163420.

[30] John R Deller, John H L Hansen, and John G Proakis. "Signal Processing Background BT - Discrete Time Processing of Speech Signals, 1st edition". In: *Discrete Time Processing of Speech Signals, 1st edition*. Prentice Hall PTR, 1999. ISBN: 9780470544402. URL: http://ieeexplore.ieee.org/search/srchabstract.jsp?arnumber=5311871papers3://publication/doi/10.1109/9780470544402.part1.

[31] R. G. Bachu et al. "Voiced/unvoiced decision for speech signals based on zero-crossing rate and energy". In: *Advanced Techniques in Computing Sciences and Software Engineering*. 2010, pp. 279–282. ISBN: 9789048136599. DOI: 10.1007/978-90-481-3660-5-47. URL: https://link.springer.com/chapter/10.1007/978-90-481-3660-5{_}47.

[32] Mathuranathan Viswanathan and Mathuranathan Varsha. *Digital Modulations using Matlab: Build Simulation Models from Scratch*. 2017, p. 184. ISBN: 978-1521493885.

[33] Seyed Omid Sadjadi and John H.L. Hansen. "Unsupervised speech activity detection using voicing measures and perceptual spectral flux". In: *IEEE Signal Processing Letters* 20.3 (2013), pp. 197–200. ISSN: 10709908. DOI: 10.1109/LSP.2013.2237903.

[34] Andrew Ng et al. *Machine Learning Project Final Report VIDEO GAME GENRE CLASSIFICATION USING VIDEO GAME MUSIC*. 2016. URL: http://cs229.stanford.edu/proj2016/report/NiShiWugofski{_}FinalReport.pdf.

[35] Ethem Alpaydin. *Introduction to machine learning*. Ed. by Christopher Bishop et al. Massachusetts Institute of Technology, 2014, p. 640. ISBN: 9780262028189. DOI: 10.4018/978-1-7998-0414-7.ch003.

[36] Andrej Karpathy. *Software 2.0 – Andrej Karpathy – Medium*. 2017. URL: https://medium.com/@karpathy/software-2-0-a64152b37c35.

[37] Alexander Radovic et al. *Machine learning at the energy and intensity frontiers of particle physics*. 2018. DOI: 10.1038/s41586-018-0361-2.

[38] Virginia R. de Sa. "Learning Classification with Unlabeled Data". In: *Proc NIPS93 Neural Information Processing Systems* March 1997 (1993), pp. 112–119. URL: https://dl.acm.org/citation.cfm?id=2987204http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.53.9872.

[39] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. "Training algorithm for optimal margin classifiers". In: *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*. Publ by ACM, 1992, pp. 144–152. ISBN: 089791497X. DOI: 10.1145/130385.130401.

[40] William S. Noble. *What is a support vector machine?* 2006. DOI: 10.1038/nbt1206-1565.

[41] Asa Ben-Hur and Jason Weston. "A user's guide to support vector machines." In: *Methods in molecular biology (Clifton, N.J.)* 609 (2010), pp. 223–239. ISSN: 19406029. DOI: 10.1007/978-1-60327-241-4_13.

[42] Faria Nazir et al. "Mispronunciation detection using deep convolutional neural network features and transfer learning-based model for Arabic phonemes". In: *IEEE Access*

7 (2019), pp. 52589–52608. ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2912648.

[43] Su Youn Yoon, Mark Hasegawa-Johnson, and Richard Sproat. "Landmark-based automated pronunciation error detection". In: *Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH 2010*. 2010, pp. 614–617. URL: http://www.ets.org/research/policy{_}research{_}reports/publications/chapter/2011/jdjk.

[44] Yanlu Xie et al. "Landmark of Mandarin nasal codas and its application in pronunciation error detection". In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. Vol. 2016-May. Institute of Electrical and Electronics Engineers Inc., 2016, pp. 5370–5374. ISBN: 9781479999880. DOI: 10.1109/ICASSP.2016.7472703.

[45] Scikit-learn. *1.4. Support Vector Machines — scikit-learn 0.21.3 documentation*. URL: https://scikit-learn.org/stable/modules/svm.html (visited on 11/22/2019).

[46] Leo Breiman. "Random forests". In: *Machine Learning* 45.1 (2001), pp. 5–32. ISSN: 08856125. DOI: 10.1023/A:1010933404324.

[47] Philipp Probst, Marvin N. Wright, and Anne Laure Boulesteix. *Hyperparameters and tuning strategies for random forest*. 2019. DOI: 10.1002/widm.1301. arXiv: 1804.03515. URL: http://arxiv.org/abs/1804.03515http://dx.doi.org/10.1002/widm.1301.

[48] Gérard Biau and Erwan Scornet. "A random forest guided tour". In: *Test* 25.2 (2016), pp. 197–227. ISSN: 11330686. DOI: 10.1007/s11749-016-0481-7. arXiv: 1511.05741.

[49] Moner N. M. Arafa et al. "A Dataset for Speech Recognition to Support Arabic Phoneme Pronunciation". In: *International Journal of Image, Graphics and Signal Processing* 10.4 (2018), pp. 31–38. ISSN: 20749074. DOI: 10.5815/ijigsp.2018.04.04.

[50] Gongde Guo et al. “KNN model-based approach in classification”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 2888 (2003), pp. 986–996. ISSN: 03029743. DOI: 10.1007/978-3-540-39964-3_62.

[51] Yang Li and Li Guo. “An active learning based TCM-KNN algorithm for supervised network intrusion detection”. In: *Computers and Security* 26.7-8 (2007), pp. 459–467. ISSN: 01674048. DOI: 10.1016/j.cose.2007.10.002.

[52] Loredana Firte, Camelia Lemnaru, and Rodica Potolea. “Spam detection filter using KNN algorithm and resampling”. In: *Proceedings - 2010 IEEE 6th International Conference on Intelligent Computer Communication and Processing, ICCP10*. 2010, pp. 27–33. ISBN: 9781424482306. DOI: 10.1109/ICCP.2010.5606466.

[53] Zhenmei Shi and Yifei Luan. “Statistical Analysis and Introspection on Research Situation of Foreign Language Teaching in China under CALL Environment”. In: Association for Computing Machinery (ACM), 2019, pp. 43–46. ISBN: 9781450371698. DOI: 10.1145/3345094.3345117.

[54] J Mártony. “On the correction of the voice pitch level for severely hard of hearing subjects.” In: *American Annals of the Deaf* 113.2 (1968), pp. 195–202. ISSN: 0002726X. URL: http://www.ncbi.nlm.nih.gov/pubmed/5651476.

[55] Catia Cucchiarini et al. “Assessment of Dutch pronunciation by means of automatic speech recognition technology”. In: *Proc. of the fifth Int. Conference on Spoken Language Processing (ICSLP’98)*. 1998, pp. 1739–1742. URL: http://hdl.handle.net/2066/75007.

[56] Leonardo Neumeyer et al. “Automatic scoring of pronunciation quality”. In: *Speech Communication* 30.2 (2000), pp. 83–93. ISSN: 01676393. DOI: 10.1016/S0167-6393(99)00046-1.

[57] Silke Maren Witt. “Use of speech recognition in CALL”. In: *PhD Dissertation* (1999). URL: https://www.researchgate.net/publication/2613422{_

}Use{_}of{_}Speech{_}Recognition{_}in{_}Computer-assisted{_}Language{_}Learning{_}PhD{_}Dissertation.

[58] S. M. Witt and S. J. Young. "Phone-level pronunciation scoring and assessment for interactive language learning". In: *Speech Communication* 30.2 (2000), pp. 95–108. ISSN: 01676393. DOI: 10.1016/S0167-6393(99)00044-8.

[59] Wenping Hu et al. "Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers". In: *Speech Communication* 67 (2015), pp. 154–166. ISSN: 01676393. DOI: 10.1016/j.specom.2014.12.008. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167639314001010.

[60] Hung Yi Lee et al. "Personalizing Recurrent-Neural-Network-Based Language Model by Social Network". In: *IEEE/ACM Transactions on Audio Speech and Language Processing* 25.3 (2017), pp. 519–530. ISSN: 23299290. DOI: 10.1109/TASLP.2016.2635445.

[61] Aleksandr Diment et al. "Detection of Typical Pronunciation Errors in Non-native English Speech Using Convolutional Recurrent Neural Networks". In: Institute of Electrical and Electronics Engineers (IEEE), 2019, pp. 1–8. ISBN: 9781728119854. DOI: 10.1109/ijcnn.2019.8851963.

[62] Longfei Yang et al. "Improving pronunciation erroneous tendency detection with convolutional long short-term memory". In: *Proceedings of the 2017 International Conference on Asian Language Processing, IALP 2017*. Vol. 2018-Janua. Institute of Electrical and Electronics Engineers Inc., 2018, pp. 52–56. ISBN: 9781538619803. DOI: 10.1109/IALP.2017.8300544.

[63] Xuesong Yang, Anastassia Loukina, and Keelan Evanini. "Machine learning approaches to improving pronunciation error detection on an imbalanced corpus". In: *2014 IEEE*

Workshop on Spoken Language Technology, SLT 2014 - Proceedings. Institute of Electrical and Electronics Engineers Inc., 2014, pp. 300–305. ISBN: 9781479971299. DOI: 10.1109/SLT.2014.7078591.

[64] Justin Salamon and Juan Pablo Bello. "Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification". In: *IEEE Signal Processing Letters* 24.3 (2017), pp. 279–283. ISSN: 10709908. DOI: 10.1109/LSP.2017.2657381. arXiv: 1608.04363. URL: http://arxiv.org/abs/1608.04363http://dx.doi.org/10.1109/LSP.2017.2657381.

[65] Arnulf B A Graf, Alexander J. Smola, and Silvio Borer. "Classification in a normalized feature space using support vector machines". In: *IEEE Transactions on Neural Networks* 14.3 (2003), pp. 597–605. ISSN: 10459227. DOI: 10.1109/TNN.2003.811708.

[66] Adam Coates and Andrew Y. Ng. "Learning feature representations with K-means". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 7700 LECTU (2012), pp. 561–580. ISSN: 03029743. DOI: 10.1007/978-3-642-35289-8-30. URL: http://link.springer.com/10.1007/978-3-642-35289-8{_}30.

[67] Hanna Leena Halme, Antti Korvenoja, and Eero Salli. "ISLES (SISS) challenge 2015: Segmentation of stroke lesions using spatial normalization, random forest classification and contextual clustering". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 9556. Springer Verlag, 2016, pp. 211–221. ISBN: 9783319308579. DOI: 10.1007/978-3-319-30858-6_18.

[68] R. C. Naranjo and G. I. Alvarez. "A classifier model for detecting pronunciation errors regarding the Nasa Yuwe language's 32 vowels". In: *Ingenieria e Investigacion* 32.2 (2012), pp. 74–78. ISSN: 01205609. URL: http://www.scielo.org.co/

scielo.php?pid=S0120-56092012000200014{\&}script=sci{_}arttext{\&}tlng=en.

[69] Philipp Probst and Anne Laure Boulesteix. "To tune or not to tune the number of trees in random forest". In: *Journal of Machine Learning Research* 18 (2018), pp. 1–8. ISSN: 15337928. arXiv: 1705.05654. URL: http://arxiv.org/abs/1705.05654.

[70] Philipp Probst, Marvin N. Wright, and Anne Laure Boulesteix. *Hyperparameters and tuning strategies for random forest*. 2019. DOI: 10.1002/widm.1301. arXiv: 1804.03515. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1301.

[71] Muhammad Rizwan and David V. Anderson. "Using k-nearest neighbor and speaker ranking for phoneme prediction". In: *Proceedings - 2014 13th International Conference on Machine Learning and Applications, ICMLA 2014*. Institute of Electrical and Electronics Engineers Inc., 2014, pp. 383–387. ISBN: 9781479974153. DOI: 10.1109/ICMLA.2014.68.

[72] Afsaneh Asaei, Hervé Bourlard, and Benjamin Picart. *Investigation of KNN Classifier on Posterior Features Towards Application in Automatic Speech Recognition*. Tech. rep. 2010. URL: http://infoscience.epfl.ch/record/150577.

[73] Brent Komer, James Bergstra, and Chris Eliasmith. "Hyperopt-Sklearn: Automatic Hyperparameter Configuration for Scikit-Learn". In: *Proceedings of the 13th Python in Science Conference*. SciPy, 2014, pp. 32–37. DOI: 10.25080/majora-14bd3278-006.

[74] Damjan Krstajic et al. "Cross-validation pitfalls when selecting and assessing regression and classification models". In: *Journal of Cheminformatics* 6.1 (2014), p. 10. ISSN: 1758-2946. DOI: 10.1186/1758-2946-6-10. URL: https://jcheminf.biomedcentral.com/articles/10.1186/1758-2946-6-10.

[75] Ron Kohavi. "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection". In: *International Joint Conference of Artificial Intelligence* (1995). URL: https://dl.acm.org/citation.cfm?id=1643047.

[76] Tiwari Vibha. "MFCC and its applications in speaker recognition". In: *International Journal on Emerging Technologies* 1 (2010), pp. 19–22. URL: https://www.researchgate.net/publication/265809116{_}MFCC{_}and{_}its{_}applications{_}in{_}speaker{_}recognition.

[77] Yow Bang Wang and Lin Shan Lee. "Toward unsupervised discovery of pronunciation error patterns using universal phoneme posteriorgram for computer-assisted language learning". In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. 2013, pp. 8232–8236. ISBN: 9781479903566. DOI: 10.1109/ICASSP.2013.6639270.

Curriculum Vitae

Aristh Valdiviezo was born on a beautiful island in Mexico. He earned the Mechatronical Engineer (IMT) degree from the Instituto Tecnológico y de Estudios Superiores de Monterrey, Monterrey Campus, in May 2012. He was accepted in the graduate programs in Intelligent Systems (MIT) and graduated in December 2019.

This document was typed in using LaTeX 2_ε[a] by José Aristh Valdiviezo Mora (Student).

[a]The style file `phdThesisFormat.sty` used to set up this thesis was prepared by the Center of Intelligent Systems of the Instituto Tecnológico y de Estudios Superiores de Monterrey, Monterrey Campus