FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Whispered speech segmentation based on Deep Learning

Gonçalo Duarte Nunes

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

Supervisor: Aníbal João de Sousa Ferreira

July 17, 2023

Abstract

Speech has shaped human history, facilitating social connections, societal organization, and transformative change. It is the foundation of human socialization, allowing expression of thoughts, ideas, and emotions. Whispered speech is used in contexts where silence or privacy is desired. Individuals with vocal impairments depend on whispered speech for communication. However, relying on whispered speech presents challenges such as weakened projection, reduced intelligibility, and the loss of vocal signature. These challenges affect interpersonal communication and interactions with voice-oriented technologies. Non-invasive assistive technologies are necessary to improve vocal communication for those relying on whispered speech. Whispered-to-normal speech conversion systems show promise by reconstructing the periodic component missing in whispered speech. This restoration enhances vocal projection, intelligibility, and the desired voiced sound signature. Accurate voicing decisions are crucial for the success of whispered-to-normal speech conversion systems. Preserving the inherent unvoiced nature of certain phones is essential for intelligibility and linguistic accuracy during speech reconstruction. Therefore, developing a voicing decision subsystem that accurately distinguishes between candidate and not candidate to voicing phones is of utmost importance. To address this challenge, the present study leverages state-of-the-art deep learning models, including TCN, CNN, LSTM, GRU, and Transformer. A comparative analysis was conducted using two feature subsets: a baseline subset consisting of 49 MFCCs features, and a 49-features subset selected through feature engineering. The results of the analysis demonstrate that the TCN model, when combined with the selected 49-features subset, exhibits superior performance across various evaluation metrics. Specifically, the TCN model outperforms other models in terms of Accuracy (98.72%), Precision (98.71%), Recall (98.74%), Specificity (98.71%), F1 Score (98.72%), and AUC-ROC (99.91%). This best performing model was further assessed, substantiating its effectiveness and online usability. The use of the selected features subset instead of the baseline features subset enabled absolute gains of performance across all models and metrics. For instance, the TCN model exhibits gains of 3.25% in Accuracy, 2.94% in Precision, 3.60% in Recall, 2.90% in Specificity, 3.27% in F1 Score, and 0.72% in AUC-ROC. Similar enhancements are verified in all other models. These findings underscore the potential of deep learning approaches in enhancing the performance of whispered-to-normal speech conversion systems, providing a promising avenue for improving the communication abilities and overall quality of life for individuals with impaired phonation ability.

Keywords: Voicing decision, candidate to voicing, whispered speech, deep learning, whispered-to-normal, speech conversion.

Resumo

A fala tem desempenhado um papel crucial na história da humanidade, contribuindo para a formação de laços sociais, a organização da sociedade e a instigação de mudanças transformativas. Constitui a base da socialização humana, permitindo a expressão de pensamentos, ideias e emoções. A fala sussurrada é empregue em contextos nos quais o silêncio ou a privacidade são desejados. Contudo, existem indivíduos com problemas de saúde que afetam as cordas vocais, sendo obrigados a recorrer à fala sussurrada como único modo de comunicação vocal. Esta dependência origina vários problemas, como a atenuação da projeção vocal, a redução da inteligibilidade e a perda da assinatura sonora individual. Estes entraves comprometem a comunicação interpessoal e as interações com tecnologias orientadas para a voz. Neste sentido, a fim de melhorar a comunicação vocal destes pacientes, tornou-se indispensável o desenvolvimento de tecnologias assistivas não invasivas. Os sistemas de conversão de fala sussurrada em fala normal surgem como uma solução promissora, possibilitando a reconstrução da componente periódica ausente na fala sussurrada, melhorando a projeção vocal, a inteligibilidade e a assinatura vocal desejada. O êxito dos sistemas de conversão de fala sussurrada em fala normal está dependente do sucesso das decisões de vozeamento, que são fulcrais para a preservação da natureza não-vozeada de certos fonemas. Apenas deste modo é possível assegurar a inteligibilidade e a precisão linguística durante a reconstrução da fala sussurrada. Portanto, é fundamental o desenvolvimento de um subsistema capaz de efetuar decisões de vozeamento, que permita distinguir de forma precisa entre os fonemas que são candidatos e os que não são candidatos ao vozeamento, em tempo real. Para superar este desafio, este estudo recorre a modelos de Deep Learning de última geração, incluindo TCN, CNN, LSTM, GRU e Transformer. Foi realizada uma análise comparativa utilizando dois subconjuntos de características: um subconjunto base composto por 49 MFCCs e um subconjunto de 49 características selecionadas. Os resultados da análise indicam que o modelo TCN, quando combinado com o subconjunto de 49 características selecionadas, apresenta um desempenho superior em várias métricas de avaliação, nomeadmente em termos de Accuracy (98,72%), Precision (98,71%), Recall (98,74%), Specificity (98,71%), F1 Score (98,72%) e AUC-ROC (99,91%). Este modelo de alto desempenho foi objeto de uma avaliação mais detalhada, a fim de validar a sua eficácia e capacidade de operação em tempo real. A utilização do subconjunto de características selecionadas, em vez do subconjunto base de características, resultou em melhorias de desempenho em todos os modelos e métricas. Por exemplo, o modelo TCN apresentou melhorias de Accuracy (3.25%), Precision (2.94%), Recall (3.60%), Specificity (2.90%), F1 Score (3.27%) e AUC-ROC (0.72%). Foram observadas melhorias semelhantes em todos os outros modelos. Estas descobertas realçam o potencial das abordagens de Deep Learning na obtenção de decisões de vozeamento, permitindo melhorar o desempenho dos sistemas de conversão de fala sussurrada em fala normal. Assim, a abordagem proposta fornece uma base promissora para a melhoria das capacidades de comunicação e a qualidade de vida dos pacientes com a capacidade de fonação comprometida.

Palavras-chave: Decisão de vozeamento, candidatos a vozeamento, classificação de fones, fala sussurrada, aprendizagem computacional, conversão de fala, sussurada para normal.

Agradecimentos

Ao meu orientador, Professor Aníbal João de Sousa Ferreira, agradeço todo o tempo e dedicação dispensados.

Ao meu supervisor no INESC TEC, Engenheiro João Miguel Pinto Pereira da Silva, agradeço todo o seu empenho em proporcionar um ambiente de aprendizagem dinâmico e eficiente. A sua paixão contagiante pelo tema da dissertação, ambição e profissionalismo foram cruciais para o sucesso deste trabalho.

À minha mãe, Cristina, e ao meu pai, Laurentino, agradeço o apoio emocional prestado bem como a transmissão de valores culturais, morais e sociais, fundamentais para o meu desenvolvimento pessoal.

À minha companheira e amiga, Joana Manuel, agradeço a sua presença em todos os momentos, que me permitiu atingir a estabilidade necessária para superar adversidades e atingir objetivos com determinação.

À minha família, agradeço o afeto e a confiança demonstrados desde o início da minha vida.

Aos meus amigos, agradeço todas as conversas enriquecedoras e momentos de lazer, que aligeiraram os piores momentos e intensificaram os melhores.

Gonçalo Duarte Nunes

"The best way to predict the future is to create it."

Peter Drucker

Contents

List of Figures

List of Tables

Abbreviations and Symbols

ANOVA Analysis of Variance

APLS Average Phone Length in Samples

ASR Automatic Speech Recognition

AUC-ROC Area Under the Receiver Operating Characteristic Curve

AWLF Average Word Length in Frames

AWLP Average Word Length in Phones

Adam Adaptive Moment Estimation

BFS Baseline Feature Subset

CAPE-V Consensus Auditory-Perceptual Evaluation of Voice

CCS Contiguous Context Size

CELP Code-Excited Linear Prediction

CENS Chroma Energy Normalized Statistics

CIT Classifier's Inference Time

CNN Convolutional Neural Network

CPU Central Processing Unit

CQT Constant-Q Transform

CTV Candidate to Voicing

CUDA Compute Unified Device Architecture

CVCV Consonant-Vowel-Consonant-Vowel

CV Consonant-Vowel

DCT Discrete Cosine Transform

DL Deep Learning

DNN Deep Neural Network

DTW Dynamic Time Warping

EDA Exploratory Data Analysis

EP European Portuguese

FFT Fast Fourier Transform

FLOPS Floating-point Operations per Second

FNN Feed Forward Neural Network

FS Frame Size

GMM Gaussian Mixture Model

GPU Graphics Processing Unit

GRU Gated Recurrent Unit

H1 Hypothesis 1

H2 Hypothesis 2

H3 Hypothesis 3

HAS Human Auditory System

HSPS Human Speech Production System

HS Hop Size

ID Identification

IFET Individual Feature Extraction Time

IPA International Phonetic Association

K-FCV K-Fold Cross Validation

KNN K-Nearest Neighbors

LPC Linear Predictive Coding

LSTM Long Short-Term Memory

MAPT Maximum Allowable Processing Time

MATLAB Matrix Laboratory

MFCCs Mel-Frequency Cepstral Coefficients

ML Machine Learning

MOCHA Multilingual, Open-source Corpus of Heterogeneous Acoustic data

MOS Mean Opinion Score

NCTV Not Candidate to Voicing

NLP Natural Language Processing

NS Normal Speech

OCS Overlapping Context Size

OS Operating System

PAL Phonetic Annotation Label

PCA Principal Component Analysis

PCC Pearson Correlation Coefficient

PSD Power Spectral Density

RAM Random Access Memory

RFI Random Forest Importance

RMS Root-Mean-Square

RNN Recurrent Neural Network

ReLU Rectified Linear Unit

ResNet Residual Neural Network

SAMPA Speech Assessment Methods Phonetic Alphabet

SCC Spearman Correlation Coefficient

SFS Selected Feature Subset

SF Sampling Frequency

SM Speech Mode

SSD Solid State Drive

STFT Short-Time Fourier Transform

STRAIGHT Speech Transformation and Representation using Adaptive Interpolation of Weighted Spectrum

SVM Support Vector Machine

SWAV Segment's Waveform Audio File Format File

TCN Temporal Convolutional Network

TEPT Total Estimated Processing Time

TFET Total Feature Extraction Time

TIMIT Texas Instruments/Massachusetts Institute of Technology Acoustic-Phonetic Continuous Speech Corpus

TTS Train-Test Split

UV Unvoiced

VD Voicing Decision

V Voiced

WAV Waveform Audio File Format

WS Whispered Speech

ZCR Zero Crossing Rate

cuDNN Compute Unified Device Architecture Deep Neural Network library

wTIMIT Whispered Texas Instruments/Massachusetts Institute of Technology Acoustic-Phonetic Continuous Speech Corpus

Chapter 1

Introduction

This Chapter offers an introduction of the research topic, presenting its overview and motivation (**1.1**), objectives (**1.2**), research question and hypotheses (**1.3**), and document structure (**1.4**).

1.1 Overview and motivation

Speech has played an invaluable role in shaping the course of human history. By serving as a medium for communication, it has been instrumental in fostering social connections, facilitating societal organization, and instigating transformative change. Undeniably, speech communication stands as the backbone of human socialization [5, 6, 7].

Whispered Speech (WS) is an Unvoiced (UV) speech mode produced by a turbulent flow of air that is expelled by the lungs and is forced through the supra-laryngeal structures, acting as an excitation signal to the vocal tract. Since there is no vibration of the vocal folds in the larynx — a mechanism also known as phonation — involved in its production, it lacks the periodic component of voice excitation that is present in Normal Speech (NS) [8, 9].

This mode of speech is used intentionally in human vocal communication, particularly in certain environments where silence is recommended or privacy is desired [10, 11]. However, there is a group of health conditions that temporarily or permanently affect the vocal folds, impairing or disabling the phonation ability. Patients affected by this health condition rely solely on this UV speech mode [12, 13, 14, 15, 16]. Being characterized by a weak vocal projection, reduced intelligibility and a loss of the individual Voiced (V) sound signature, involuntary WS is detrimental to their vocal communication ability.

Since human-to-human vocal communication is an essential mechanism of socialization, their mental health and well-being may deteriorate [17, 18, 19, 20]. The limitations imposed by involuntary WS affect human-machine interaction as well, compromising the effectiveness of the increasingly adopted voice-oriented technologies based on Automatic Speech Recognition (ASR), limiting user experience and reducing accessibility [21, 22, 23]. Therefore, there is a pressing need to develop non-invasive assistive technologies that improve these patients' vocal communication ability [11, 24, 25].

Whispered-to-normal conversion systems allow the conversion of WS into NS, enhancing its vocal projection and intelligibility, while providing the V sound signature desired by its users [11, 24, 25, 26, 27, 28]. To synthetically voice an originally UV speech, the system reconstructs the periodic component of V speech that is missing.

There are some phones in the European Portuguese (EP) language that are originally UV in NS, being correctly produced by individuals with impaired or disabled phonation ability. There is no need for the whispered-to-normal conversion systems to voice those phones. In fact, an incorrect decision to voice a phone that should not be V may affect the intelligibility and the linguistic content of the reconstructed speech. Thus, the success of the speech reconstruction is highly dependent on the effectiveness of each Voicing Decision (VD). This fact highlights the importance of the development of a VD subsystem to integrate the broader whispered-to-normal speech conversion system. This subsystem segments the speech signal based on the classification between two major classes of phones — Candidate to Voicing (CTV) and Not Candidate to Voicing (NCTV) — so that the synthetic voicing mechanism is accurately triggered.

For that purpose, the utilization of state-of-the-art VD systems is crucial. The encouraging results reported in recent literature regarding the effectiveness of VD systems based on Deep Learning (DL) motivated their integration into this dissertation [29, 30, 31, 32, 33].

1.2 Objectives

The goal of this research work is the development of a VD subsystem able to perform effective and efficient online frame-based classification of EP WS between CTV and NCTV. The specific objectives of this research are detailed as follows:

1. **Phonetically annotated WS/NS dataset description**: Describe the acquisition and characteristics of the available phonetically annotated dataset;

2. **Dataset preprocessing**: Prepare the dataset for feature engineering;

3. **Feature engineering**: Extract features from the preprocessed dataset; Process them to be amenable for subsequent analysis; Define a Baseline Feature Subset (BFS) and a Selected Feature Subset (SFS);

4. **Selection and design of DL-based model architectures**: Choose and design DL model architectures, defining structures, layers, and parameters;

5. **Evaluation metrics definition**: Define performance and computational metrics to quantitatively evaluate the model/features subset pairs;

6. **Assessment and comparison of all model/features subset pairs**: Evaluate and compare performance and computational efficiency of model/features subsets, by performing Train-Test Split (TTS) evaluations; Compare the performances obtained across features subsets; Identify the best performing pair;

7. **Assessment of the best performing model/features subset pair**: Further assess the best performing pair, by executing: K-Fold Cross Validation (K-FCV) evaluation; performance analysis across articulation manner classes; VD segmentation exemplification; verification of the compliance with a defined Maximum Allowable Processing Time (MAPT) that guarantees online operation.

1.3 Research question and hypotheses

The research question and hypotheses articulated in this Section serve as a systematic framework for investigating the effectiveness and efficiency of a DL-based model in executing online frame-based VD within the context of EP WS:

- **Research question:** *"What is the effectiveness and efficiency of a carefully chosen Deep Learning (DL)-based model which performs online frame-based VDs in European Portuguese (EP) Whispered Speech (WS), utilizing a Selected Feature Subset (SFS) as input?"*
  - **Hypothesis 1 (H1):** *A carefully chosen DL-based model **effectively** performs online frame-based VDs in EP WS*;
  - **Hypothesis 2 (H2):** *A carefully chosen DL-based model performs online frame-based VDs in EP WS **efficiently**, taking less than the MAPT to process and decide on the input features*;
  - **Hypothesis 3 (H3):** *A carefully chosen SFS, when used as input, **improves the effectiveness and efficiency** of a DL-based model performing online frame-based VDs in EP WS, compared to a BFS.*

1.4 Document structure

The dissertation document is structured as follows:

1 Introduction: This chapter introduces the dissertation, presenting its:

1.1 Overview and motivation;

1.2 Objectives;

1.3 Research question and hypotheses;

1.4 Document structure;

2 Background: This chapter provides background information on various topics related to the research. It includes sections on:

2.1 Human Speech Production System (HSPS): Discusses the HSPS, including the speech production mechanism (2.1.1), EP phonetics (2.1.2), and WS and NS modes (2.1.3);

2.2 Human Auditory System (HAS): Provides an overview of the HAS, covering the peripheral region (2.2.1) and psychoacoustics (2.2.2);

2.3 Speech signal processing: Discusses speech signal processing techniques, including the source-filter model (2.3.1) and Linear Predictive Coding (LPC) (2.3.2);

2.4 Deep Learning (DL): Introduces DL and its relevance to the research. Covers topics such as DL-based models (2.4.1), learning frameworks (2.4.2), evaluation techniques (2.4.3), performance metrics (2.4.4), and computational metrics (2.4.5);

2.5 Feature engineering: Provides an overview on feature engineering, covering feature extraction (2.5.1) and feature selection (2.5.2);

2.6 Voicing Decision (VD) in whispered-to-normal speech conversion systems: Focuses on the VD in whispered-to-normal speech conversion systems, which is the key aspect of the research;

3 Voicing decision approaches — a review: This chapter presents a review of VD approaches. It includes sections on:

3.1 Paper selection: Discusses the criteria and process for selecting relevant papers for the review;

3.2 Review of the selected papers: Provides a detailed review of the selected papers, categorizing them into rule-based approaches (3.2.1), Machine Learning (ML)-based approaches (3.2.2), and hybrid approaches (3.2.3);

4 Methodology: This chapter presents the methodology employed in the research. It includes sections on:

4.1 Hardware and software description: Provides a description of the hardware (4.1.1) and software (4.1.2) used in the research;

4.2 Phonetically annotated WS/NS dataset acquisition: Explains the phonetically annotated WS/NS dataset acquisition process, including participant selection, recording, screening, and training (4.2.1), corpus (4.2.2), and phonetic annotation (4.2.3);

4.3 Dataset preprocessing: Describes the preprocessing steps applied to the dataset, such as downsampling of audio files (4.3.1), phone annotation-based segmentation (4.3.2), dataset selection and cleaning (4.3.3), CTV/NCTV segments labelling (4.3.4), and audio segments normalization (4.3.5);

4.4 Feature engineering: Explains the feature engineering process, including feature extraction (4.4.1), feature normalization (4.4.2), dataset explosion from segments to frames (4.4.3), context size definition (4.4.5), context-sized sequences dataset generation (4.4.6), BFS definition (4.4.7), and SFS definition (4.4.8);

4.5 Selection and design of DL-based model architectures: Covers the selection and design process of DL-based model architectures;

4.6 Evaluation metrics definition: Describes the definition of metrics to quantitatively assess the models, namely performance metrics (4.6.1) and computational metrics (4.6.2);

4.7 Assessment and comparison of all model/features subset pairs: Presents the methods used to assess and compare all model/features subset pairs, namely TTS evaluation (4.7.1), performance comparison across feature subsets (4.7.2), and selection of the best performing model/features subset (4.7.3);

4.8 Assessment of the best performing model/features subset pair: Presents the methods used to assess the best performing model/features subset pair, namely performance assessment across articulation manner classes (4.8.1), K-FCV evaluation (4.8.2), exemplification of VD segmentation (4.8.3), and verification of the compliance with the MAPT (4.8.4);

5 Results and Discussion: This chapter includes the presentation and discussion of the results. It includes sections on:

5.1 Phonetically annotated Whispered Speech (WS)/Normal Speech (NS) dataset acquisition: Presents and discusses the results related to the acquisition of the phonetically annotated WS/NS dataset;

5.2 Dataset preprocessing: Presents and discusses the results of the dataset preprocessing steps;

5.3 Feature engineering: Presents and discusses the results obtained from the feature engineering process, more specifically from: feature extraction (5.3.1), feature normalization (5.3.2), dataset explosion from segments to frames (5.3.3), class distribution balancing (5.3.4), context size definition (5.3.5), context-sized sequences dataset generation (5.3.6), BFS definition (5.3.7), and SFS definition (5.3.8);

5.4 Selection and design of Deep Learning-based model architectures: Presents and discusses the resulting architectures of the selection and design process: Convolutional Neural Network (CNN) (5.4.1), Separable CNN (5.4.2), Residual Neural Network (ResNet) (5.4.3), Long Short-Term Memory (LSTM) (5.4.4), Gated Recurrent Unit (GRU) (5.4.5), Temporal Convolutional Network (TCN) (5.4.6), and Transformer (5.4.7);

5.5 Assessment and comparison of all model/feature subset pairs: Presents and discusses the results of the assessment and comparison of all model/feature subset pairs, focusing on TTS evaluation using the BFS (5.5.1) and the SFS (5.5.2), performance comparison across features subsets (5.5.3), and selection of the best performing model/features subset pair (5.5.4);

5.6 Assessment of the best performing model/features subset pair: Presents and discusses the results of the assessment of the best performing model/feature subset pair, focusing on performance assessment across articulation manner classes (5.6.1),

K-FCV (5.6.2), exemplification of VD segmentation (5.6.3), and compliance with the MAPT (5.6.4);

6 Conclusions: This chapter provides the conclusions drawn from the research, as follows:

6.1 Summary of key findings: Presents a summary of the key findings derived from Chapters *Methodology* and *Results*;

6.2 Research question and hypotheses: Answers the research question, by addressing and validating the hypotheses H1 (6.2.1), H2 (6.2.2) and H3 (6.2.3);

6.3 Contributions, innovations and implications: States the main contributions (6.3.1), innovations (6.3.2) and implications (6.3.3) of the research work;

6.4 Limitations and future work: Identifies the limitations of the research and proposes future work to overcome them, focusing on time and computational resources (6.4.1), data (6.4.2), feature engineering (6.4.3), selection and design of DL-based model architecures (6.4.4) and assessment of model/features subset pairs (6.4.5).

1.5 Chapter summary

In the Chapter *"Introduction"* (1), an overview and motivation (1.1) for the research were presented. The objectives (1.2) of the study were outlined, followed by the research question and hypotheses (1.3). The Chapter concludes with a brief description of the document structure (1.4).

The next Chapter *"Background"* (2), will provide the necessary background information for the research. It will cover various topics, including the HSPS (2.1), the HAS (2.2), speech signal processing techniques (2.3), DL (2.4), feature engineering (2.5), and VD in whispered-to-normal speech conversion systems (2.6).

Chapter 2

Background

This Chapter *"Background"* provides foundational knowledge on various topics related to the research. It covers the HSPS (2.1), the HAS (2.2), speech signal analysis and modelling techniques (2.3), and an introduction to DL (2.4). Additionally, it focus on the VD in whispered-to-normal speech conversion systems (2.6).

2.1 Human speech production system

The HSPS is responsible for translating thoughts into speech. This process encompasses the following phases: selection of words, organization of grammatical forms, and articulation of the resulting sounds using the vocal apparatus. The last phase will be addressed in this Section, by covering the speech production mechanism (2.1.1), EP phonetics (2.1.2) and the WS and NS modes (2.1.3).

2.1.1 Speech production mechanism

The main human body organs that enable the human speech production mechanism are the lungs, the larynx, the pharynx, the nose, and the mouth. The energy source of this mechanism is the force responsible for the expulsion of air from the lungs. This flux of air is modulated in various ways, originating an acoustic wave that is propagated through a set of several cavities — the vocal tract — and radiated by the mouth and nostrils. The vocal tract is an acoustic tube limited by the larynx and the lips. In the larynx, there are two tissue folds — the vocal folds. The transversal section area of this tube is not uniform, varying with the movement of the articulators — lips, jaws, tongue, and velum. The velum is responsible for the coupling of the vocal tract with the nasal tract — another acoustic tube — that is limited by the velum and the nostrils.

The main modes of speech sound production are [34]:

- **Phonation:** Consists in the vibratory action of the vocal folds, causing periodic interruption of the air flux from trachea to pharynx;
- **Turbulence:** Caused by vocal tract constriction, generating a turbulent air flux.

2.1.2 European Portuguese phonetics

This Subsection provides an overview of EP phonetics, encompassing voicing (2.1.2.1), vowels (2.1.2.2) and consonants (2.1.2.3).

2.1.2.1 Voicing

The produced speech sound is considered V if phonation occurs. Otherwise, it is considered UV.

2.1.2.2 Vowels

Vowels are produced when phonation occurs without turbulence. Thus, they are always V sounds. There are two types of vowel sounds:

- **Monophthongs:** Vowel sounds pronounced with a single, unchanging articulation of the vocal tract. In other words, the tongue and lips remain fixed in one position while the vowel sound is being produced. Table 2.1 presents all the 14 EP vowel monophtongs (9 oral and 5 nasalized);
- **Diphthongs:** Vowel sounds which involve a gradual movement of the tongue and/or lips from one position to another within the same syllable. Table 2.2 presents all the 14 EP vowel monophtongs (10 oral and 4 nasalized).

Figure 2.1 presents the EP vowel space, describing vowel production in terms of two dimensions:

- **Vowel height:** Represented in the vertical axis, refers to the position of the tongue in the vertical plane of the mouth. It is determined by how much space there is between the tongue and the roof of the mouth. Divides the vowel space into four main vowel height categories: close, close-mid, open-mid, and open;
- **Vowel backness:** Represented in the horizontal axis, refers to the position of the tongue in the horizontal plane of the mouth. It is determined by how close the tongue is to the back of the mouth. Divides the vowel space into three main vowel backness categories: front, central, and back.

2.1.2.3 Consonants

Phonation may occur concurrently with turbulence. Consonants are produced whenever turbulence occurs. The consonant sounds of EP may be classified based on their articulation manner, accordingly to the International Phonetic Association (IPA) alphabet [1]:

- **Plosives:** Plosives, also known as stops, are consonant sounds that are produced by completely blocking the flow of air through the vocal tract and then releasing it suddenly. Examples of plosive sounds in the IPA alphabet include [p], [t], and [k];

Table 2.1: European Portuguese vowel monophtongs [1].

Oral vowel monophtongs				*Nasalized vowel monophtongs*			
i	vi	*vi*	‘saw’ (1 sg)	ĩ	vĩ	*vim*	‘came’ (1 sg)
e	ve	*vê*	‘see’ (3 sg)	ẽ	ˈẽtru	*entro*	‘enter’ (1 sg)
ɛ	sɛ	*sé*	‘cathedral’	-	-	-	-
a	va	*vá*	‘go’ (3 sg)	-	-	-	-
ɔ	sɔ	*só*	‘alone’	-	-	-	-
o	so	*sou*	‘I am’	õ	sõ	*som*	‘sound’
u	ˈmudu	*mudo*	‘mute’ (m)	ũ	ˈmũdu	*mundo*	‘world’
ɐ	pɐˈgar	*pagar*	‘to pay’	ɐ̃	ˈɐ̃tru	*antro*	‘den’
ɯ	pɯˈfar	*pegar*	‘to grip’	-	-	-	-

Table 2.2: European Portuguese vowel diphthongs [1].

Oral vowel diphtongs				*Nazalized vowel diphtongs*			
ɛi	ɐˈnɛiʃ	*anéis*	‘rings’ (n)	-	-	-	-
ai	sai	*sai*	‘go out’ (3 sg)	ɐ̃i	sɐ̃i	*cem*	‘hundred’
ɐi	sɐi	*sei*	‘know’ (1 sg)	-	-	-	-
ɔi	mɔi	*mói*	‘grind’ (3 sg)	-	-	-	-
oi	ˈmoitɐ	*moita*	‘thicket’	õi	ɐˈnõiʃ	*anões*	‘dwarves’ (m)
ui	ɐˈnuiʃ	*anuis*	‘agree’ (2 sg)	ũi	ˈmũitɐ	*muita*	‘much, many’ (f)
iu	viu	*viu*	‘saw’ (3 sg)	-	-	-	-
eu	meu	*meu*	‘mine’ (poss m)	-	-	-	-
ɛu	vɛu	*véu*	‘veil’	-	-	-	-
au	mau	*mau*	‘bad’ (m sg)	ɐ̃u	mɐ̃u	*mão*	‘hand’ (n)

- **Nasals:** Nasal consonant sounds are produced by allowing air to flow through the nasal cavity while speaking. Examples of nasal sounds in the IPA alphabet include [m], [n], and [ŋ];

- **Trills:** Consonant sounds that are produced by rapid vibration of the tongue or other speech organs. These sounds are characterized by a trilled or vibrating sound. Examples of trill consonants in the IPA alphabet include [r] and [ʀ];

- **Taps or Flaps:** Consonant sounds that are produced by a brief, single-contact closure of the vocal tract. These sounds are characterized by a brief, percussive sound. Examples of tap or flap consonants in the IPA alphabet include [ɾ];

- **Fricatives:** Consonant sounds that are produced by narrowing the vocal tract and forcing air through a small opening, which creates a hissing or buzzing sound. Examples of fricative

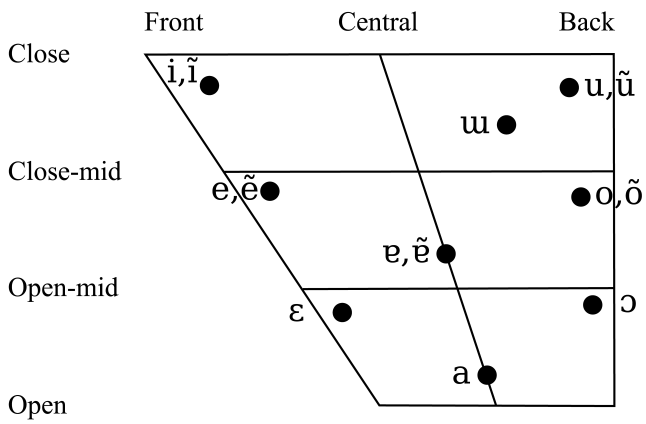

Figure 2.1: European Portuguese vowel space [1].

sounds in the IPA alphabet include [s], [f], and [ʃ];

- **Lateral Fricatives:** Consonant sounds that are produced by narrowing the vocal tract and forcing air through a small opening while allowing air to flow over the sides of the tongue. These sounds are characterized by a lateral airflow and a fricative quality. Examples of lateral fricative sounds in the IPA alphabet include [ɬ] and [ɮ];

- **Lateral Approximants:** Lateral approximant consonant sounds are produced by allowing air to flow over the sides of the tongue while speaking. These sounds are characterized by a lateral airflow through the mouth. Examples in the IPA alphabet include [l].

Table 2.3 summarizes all the consonants of the EP according to the IPA, highlighting the pairs of consonants that are articulated in the same way, differing only in voicing. An incorrect VD for one of these consonants may affect the intelligibility and the linguistic content of the reconstructed speech.

2.1.3 Whispered and normal speech modes

The NS mode is characterized by the presence of V phones in the signal. This voicing is caused by the vibration of the vocals folds — a mechanism also known as phonation — that confers a periodic component of voice excitation to speech. Thus, V speech segments can be detected by the presence of a harmonic structure. In normal speech, the vowels are always V, and the consonants may be partially or totally V. Figure 2.2 depicts the normalized waveform and spectrogram of the EP word *"pica"* uttered in NS, where the aforementioned characteristics can be observed.

In contrast, WS does not involve phonation, being solely produced by a turbulent flow of air that is expelled by the lungs and is forced through the supra-laryngeal structures, acting as an excitation signal to the vocal tract. This lack of harmonicity confers it a noisy nature. The vocal tract articulation is still able to produce formant frequencies necessary to distinguish vowels and the air

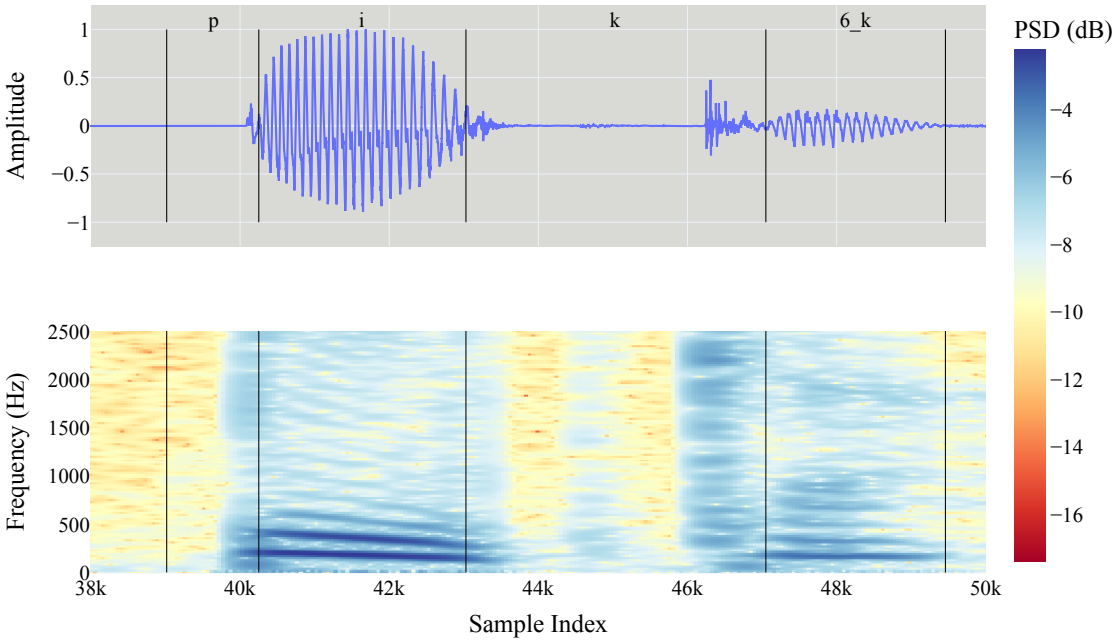

Figure 2.2: Normalized waveform and spectrogram of the word *"pica"* uttered in Normal Speech.

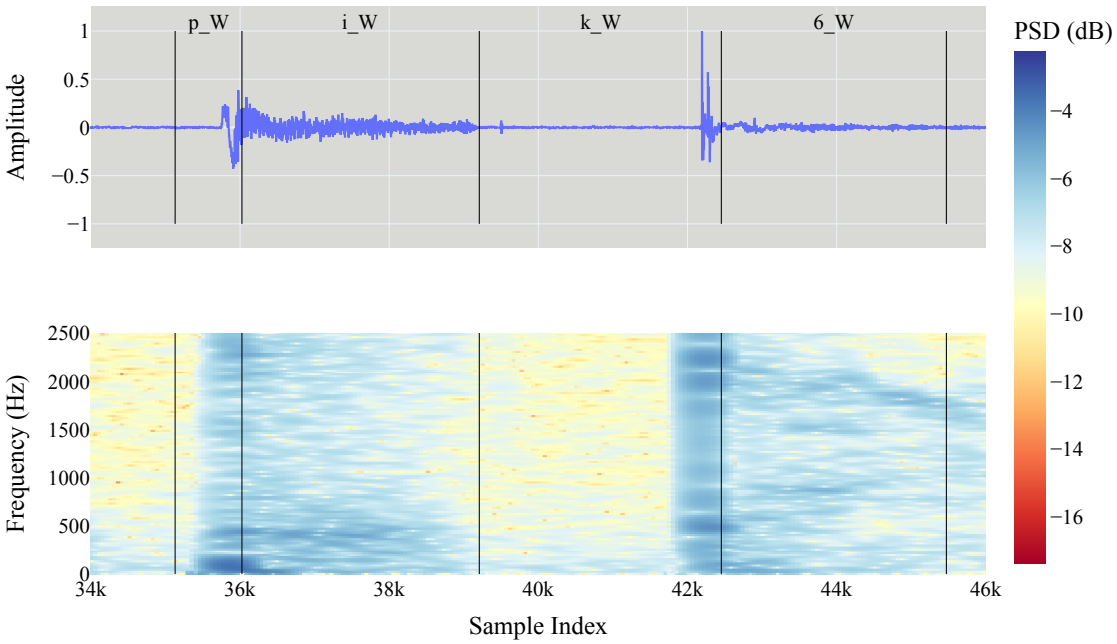

Figure 2.3: Normalized waveform and spectrogram of the word *"pica"* uttered in Whispered Speech.

Table 2.3: European Portuguese consonants [1].

	Articulation manner												
	Plosive			*Nasal*			*Tap/Flap*	*Fricative*				*Lateral Approx.*	
Voiced	b	d	g	m	n	ŋ	ɾ	v	z	ʒ	ʁ	l	ʎ
Unvoiced	p	t	k	-	-	-	-	f	s	ʃ	-	-	-

p	ˈpatu	pato	‘duck’ (m)	t	ˈtatu	tacto	‘tact’	k	katu	cacto	‘cactus’
b	ˈbatu	bato	‘I strike’	d	ˈdatu	dato	‘I date’	g	ˈɡatu	gato	‘cat’ (m)
m	ˈmatu	mato	‘I kill’	n	ˈnatu	nato	‘innate’ (m)	ŋ	ˈpiŋɐ	pinha	‘pine cone’
f	ˈfatu	fato	‘costume’	s	ˈkasu	caço	‘I hunt’	ʃ	ˈʃatu	chato	‘flat’ (m)
v	ˈviŋɐ	vinha	‘vine’	z	ˈkazu	caso	‘I marry’	ʒ	ˈʒatu	facto	‘jet’
				ɾ	ˈpiɾɐ	pira	‘pyre’	ʁ	ˈʁatu	rato	‘mouse’ (m)
				l	ˈliŋɐ	linha	’line’	ʎ	ˈpiʎa	pilha	‘battery’

flux constrictions needed for producing consonants. Thus, WS is still able to generate the desired linguistic content, but with weak vocal projection, reduced intelligibility and a loss of the individual V sound signature. Its Power Spectral Density (PSD) is flatter, with less pronounced formants, which occur at slightly higher frequencies [35]. Figure 2.3 depicts the normalized waveform and spectrogram of the EP word *"pica"* uttered in WS, where the aforementioned characteristics can be observed.

2.2 Human auditory system

The HAS enables the perception of sound, by processing the acoustic information that reaches the auricles. This processing comprises several phases, namely: capture, conditioning, mechano-electrochemical transduction, neural/synaptic conduction, and interpretation [34].

In this Section, an analysis of the HAS was performed, underlining its peripheral region (2.2.1) and psychoacoustics (2.2.2).

2.2.1 Peripheral Region

The peripheral region of the HAS allows the conversion from acoustic energy transported by the oscillation of air particles to neural information. Then, this information is communicated to the central regions of the HAS, located in the brain. This region of the HAS may be divided in three main subregions: the outer ear (2.2.1.1), the middle ear (2.2.1.2), and the inner ear (2.2.1.3) [34].

2.2.1.1 Outer ear

The outer ear includes the auricle and the ear canal, responsible for the capture and conduction of the acoustic waves until the tympanic membrane, that oscillates. The tympanic membrane

separates the outer ear from the middle ear.

2.2.1.2 Middle ear

The tympanic membrane transmits the mechanical energy through three ossicles (malleus, incus, and stapes), located in the middle ear, to another membrane — the oval window. The oval window is responsible for communicating the oscillations to an aqueous medium. The lever action performed by the ossicles, and the area relation between the tympanic membrane and the oval window allow the impedance matching between the external medium, composed by air, and the internal aqueous medium. The middle ear is filled with air, in order to establish a point of equilibrium of the tympanum and the eustachian tube, which is connected to the exterior.

2.2.1.3 Inner ear

It is mainly composed by the cochlea and auditory nerves. The cochlea is a spiral-shaped bone structure, with three parallel channels-vestibular duct, tympanic duct, and cochlear duct. Those channels are filled with a liquid, and separated by elastic membranes. The vestibular duct starts at the oval window, communicating with the tympanic duct at the other extremity of the spiral. The tympanic duct ends in a flexible membrane oriented to the middle ear, the round window. The inner ear is separated by the middle ear by the oval and round windows. Variations of pressure, introduced inside the cochlea by the stapes, are compensated by an opposite displacement of the round window.

The spectral analysis or decomposition of audio signals occurs in the cochlea, where the conversion from mechanical energy to nervous impulses takes place. This conversion is performed by thousands of hair cells, distributed along the basilar membrane. This flexible membrane separates the tympanic duct from the cochlear duct. Its physical properties vary along its length: it is thinner and more rigid in the extremity close to the oval window, and thicker and more flexible in the opposite extremity. Therefore, it has mechanical resonance characteristics along its length, which allows it to act as a spectrum analyzer.

In general terms, the first phase of the HAS's sound analysis consists on the following principle: the pressure variations caused by an audio signal and communicated to the cochlea through the oval window, escape to the round window, choosing the point of the basilar membrane with lower impedance. Because of the variations of the basilar membrane's mechanical characteristics, the point of lower impedance depends on the frequency of the sound wave. This spatial arrangement of sound reception is referred to as tonotopic tuning (or tonotopy).

2.2.2 Psychoacoustics

Psychoacoustics, a scientific discipline focused on the perceptual analysis of acoustic signals, aims to establish quantitative models that bridge the gap between objective physical properties of sounds and the human auditory experience [35]. The perception of sound by individuals can be highly subjective and does not directly correspond to objectively measurable characteristics. This

motivated the conception of perceptual characteristics of sound — loudness, pitch, and timbre —, which correspond to their objective and measurable counterparts — sound pressure, frequency, and spectral structure. Fletcher-Munson equal loudness curves (2.2.2.1) and perceptual scales (2.2.2.2) are two key topics in the realm of psychoacoustics.

2.2.2.1 Fletcher-Munson equal loudness curves

The loudness is commonly characterized by the Fletcher-Munson equal loudness curves, which represent the variations in perceptual loudness with sound pressure level and frequency.

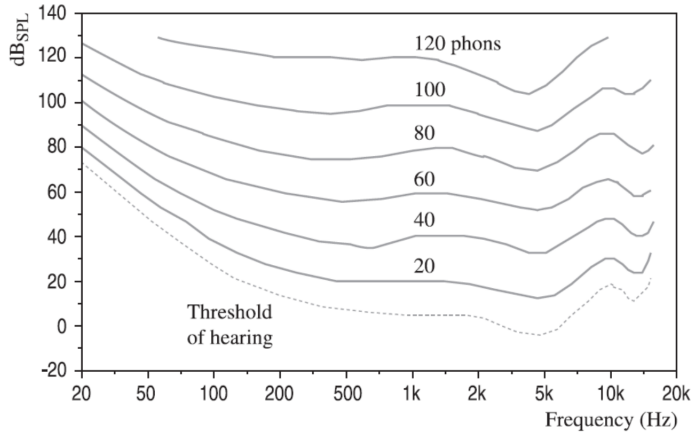

Figure 2.4: Fletcher-Munson equal loudness curves [2].

2.2.2.2 Perceptual scales

The non-linearity observed on the perception of frequency led to the development of perceptual pitch scales, which attempt to map objective frequency values to their perceptual counterpart. The more commonly used perceptual scales of pitch are:

- **Mel Scale:** The mel scale is a perceptual scale of pitches judged by listeners to be equal in distance. The reference point between this scale and the objective frequency scale is defined by assigning a perceptual pitch of 1000 *mel* to a 1000 Hz tone, 40 dB above the listener's threshold. Above 500 Hz, increasingly large intervals of objective frequency are judged by the listeners to produce equal pitch increments [36]. Expression 2.1 can be used to convert objective frequency into Mel values:

$$Mel = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{2.1}$$

- **Bark Scale:** The Bark scale is based on the critical bands of the HAS, which correspond to ranges of frequency that activate a single area of the basilar membrane. They are measured perceptually by the smallest frequency difference for which two sine tones are heard as distinct or as a single sine tone. Expression 2.2 can be used to approximately convert objective

frequency into Bark values [37]:

$$Bark = 6\sinh^{-1}\left(\frac{f}{600}\right) \tag{2.2}$$

2.3 Speech signal analysis and modeling

This Section explores the analysis and modeling of speech signals, encompassing the source-filter model (2.3.1) and LPC (2.3.2).

2.3.1 Source-filter model

The source-filter model is a widely-accepted model that describes the HSPS. It was proposed by Gunnar Fant in his book *"Acoustic Theory of Speech Production"* [38]. It suggests that the production of speech sounds can be modeled by two independent components: the source and the filter.

According to the source-filter model, the human vocal tract acts as a filter that shapes the sound produced by the vocal folds, or the source. The vocal folds generate a basic source signal during phonation, the glottal pulse. The glottal pulse is modified by the resonance frequencies of the vocal tract, which act as a filter that shapes the sound. Those resonance frequencies are determined by the size and shape of the vocal tract, which can be modified by changing the position of the speech articulators. In particular, the vocal tract changes the spectral envelope of the glottal pulses. Local peaks in the spectral envelope correspond to formant frequencies, which are useful for phone classification.

Overall, the source-filter model provides a valuable framework for understanding the complex interactions between the vocal folds and the vocal tract in the production of speech. It clarifies how different sounds can be produced using different combinations of energy sources and filters.

2.3.2 Linear Predictive Coding

LPC analysis is based on the idea that a speech sample can be approximated by a linear combination of previous samples [39, 34]. The idea is illustrated by Equation (2.3), in which $r(i)$ represents the estimate of $x(i)$, and n the order of the model, which determines the number of previous samples used in the estimation. The prediction coefficients, $y_1 \dots y_n$, are appropriate for estimating every sample if Equation (2.3) is true for all values of i.

$$x(i) \approx r(i) = \sum_{j=1}^{n} y_j x(i-j) \tag{2.3}$$

It is possible to compute a set of prediction coefficients (linear combination weights), by minimizing the sum of the squared difference between the current samples and the predicted samples (2.4), during a finite time interval.

$$sum_i[x(i) - r(i)]^2 \tag{2.4}$$

With the computed coefficients, it is possible to predict future samples of the signal. Therefore, LPC can be used in several applications, such as signal interpolation, signal restoration and noise reduction. By leveraging the relatively low number of coefficients that characterize the original signal, LPC can be applied for signal compression. To achieve compression, only the coefficients and the first n samples are stored or transmitted, and the remaining signal is approximated from these values by the recursive application of Equation (2.3). Since LPC coefficients are compact representations of the original signal, they can also be exploited to establish comparisons between different signals [40].

2.4 Deep Learning

DL models have emerged as powerful tools for representation learning in speech processing, harnessing their potential to decipher intricate patterns hidden within large volumes of data. These models are designed with an inherent ability to learn complex representations from data, diminishing the reliance on human-crafted features. This key advantage has fueled a significant shift within the speech processing community, who are increasingly adopting DL techniques for a variety of applications.

In order to explore the domain of DL, several aspects were examined, namely DL-based models (2.4.1), learning frameworks (2.4.2), and evaluation techniques (2.4.3). Additionally, widely adopted performance (2.4.4) and computational metrics (2.4.5) were described.

2.4.1 Deep Learning-based models

In the field of speech processing, various DL models have emerged as popular choices for representation learning [3]. These models leverage the power of DL techniques to extract meaningful representations from speech data. State-of-the-art DL models are briefly described next, namely CNN (2.4.1.1), Recurrent Neural Network (RNN) (2.4.1.2), and TCN (2.4.1.3).

2.4.1.1 Convolutional Neural Network

CNNs are a class of DL models known for their proficiency in dealing with image data. This proficiency is largely due to the unique architecture of CNNs, which includes specialized layers like convolutional layers and pooling layers. Convolutional layers work by sliding learnable filters over the input data to create feature maps, while pooling layers reduce the spatial dimensions of the data, resulting in a model that is not only computationally efficient, but also resilient to small shifts or distortions in the input. Stacking these layers results in a network capable of abstracting increasingly complex features from input data, which is critical for tasks such as image classification or object detection. Variants of CNNs include:

- **Separable CNN:** Separable CNNs are a modification to standard CNNs that aim to decrease computational demand while preserving performance. The primary change is in the convolutional layers, where the convolution operation is divided into depthwise and pointwise steps. By separating the convolution into these two parts, Separable CNNs are able to drastically reduce the number of mathematical operations required, making these models faster and lighter. This makes them especially useful in resource-constrained environments or for processing large-scale image or video data;

- **ResNet:** Residual Networks (ResNets) are a type of CNN that introduced the novel idea of skip connections. In ResNets, input from early layers can skip over some intermediate layers and then be added to the output of later layers. This forms a so-called residual block, which can help mitigate the vanishing gradient problem that often occurs when training very deep networks. By enabling the training of extremely deep networks, ResNets can achieve remarkable performance in tasks like image classification and object detection.

2.4.1.2 Recurrent Neural Network

Recurrent Neural Networks (RNNs) are a type of neural network designed for processing sequential data. They possess an internal loop that allows information to be passed from one step in the sequence to the next, providing the network with a form of memory. This unique feature allows RNNs to process sequences of varying lengths and to capture temporal dependencies in data, which is crucial for tasks like natural language processing or time-series prediction. Variants of RNNs include:

- **LSTM:** LSTMs are a type of RNN designed to address the issue of long-term dependencies in sequence data. They introduce a memory cell and a system of gates to control the flow of information in and out of this cell. This mechanism enables LSTMs to remember or forget information over long sequences, effectively dealing with the problem of vanishing gradients that affect standard RNNs. LSTMs are therefore ideally suited for tasks involving sequential data with long-range dependencies, such as machine translation or speech recognition;

- **GRU:** GRUs are another variation of RNNs. They also utilize gating mechanisms to control the flow of information, but with a simpler structure that involves fewer parameters. Despite this simplicity, GRUs often achieve performance on par with LSTMs, and are used in similar domains involving sequence data.

2.4.1.3 Temporal Convolutional Network

TCNs are a type of neural network that extends the applicability of CNNs to sequential data. They preserve the strengths of CNNs, like the ability to handle translation invariance, while also ensuring that the temporal order of data is respected. This is achieved by using dilated causal convolutions, a technique that allows the network to have a wider receptive field without an increase

in computational complexity. This makes TCNs particularly powerful for tasks that involve long sequences and long-term dependencies.

2.4.1.4 Transformer

The Transformer model has been a game changer in the field of Natural Language Processing (NLP). It introduces a self-attention mechanism that allows it to weigh the importance of different parts of the input data relative to each other, enabling the model to better understand context and nuances in language. Additionally, unlike RNNs and CNNs, Transformers process all parts of the input data in parallel, which leads to significant improvements in computational efficiency. As a result, Transformers have become the model of choice for many large-scale NLP tasks that require the capture of complex patterns and long-range dependencies, such as machine translation or text summarization.

2.4.2 Learning frameworks

This Subsection explores various learning frameworks that can be employed to train DL-based models on speech data, namely supervised learning (2.4.2.1), unsupervised learning (2.4.2.2), semi-supervised learning (2.4.2.3), transfer learning (2.4.2.4), and reinforcement learning (2.4.2.5).

Table 2.4 provides an overview of the key features and applications of the explored learning frameworks.

2.4.2.1 Supervised learning

In supervised learning, feature representations are learned from datasets by considering label information [3].

2.4.2.2 Unsupervised learning

Unsupervised learning enables the analysis of unlabeled input data, aiming to learn the underlying structure or distribution of data [3].

2.4.2.3 Semi-supervised learning

Semi-supervised learning resorts to large amounts of unlabeled data, together with labelled data. Often, the goal of this technique is surpassing the lack of sufficient labelled training data [3].

2.4.2.4 Transfer learning

Transfer learning is the usage of any knowledge resources (i.e., data, models, and labels) to improve model learning and generalization for the target task [3].

Table 2.4: Comparison of different learning frameworks [3].

Learning framework	**Key features**	**Applications**
Supervised Learning	Learn explicitly; Data with labels; Direct feedback is given; Predict outcome/future; No exploration.	Classification; Regression.
Unsupervised Learning	Learn patterns and structure; Data without labels; No direct feedback; No prediction; No exploration.	Clustering; Association.
Semi-Supervised Learning	Blend on both supervised and unsupervised; Data with and without labels; Direct feedback is given; Predict outcome/future; No exploration.	Classification; Clustering.
Transfer Learning	Transfer knowledge from one supervised task to other; Labelled data for different task; Direct feedback is given; Predict outcome/future; No exploration.	Classification; Regression.
Reinforcement Learning	Reward-based learning; Policy making with feedback; Predict outcome/future; Adaptable to changes through exploration.	Classification; Control.

2.4.2.5 Reinforcement learning

Reinforcement learning follows the principle of behavioral psychology: an agent learns to take actions in an environment and tries to maximize the accumulated reward over its lifetime. The agent and its environment are often modelled as a state, that contains all related information about the current situation. The agent can perform actions. The goal of reinforcement learning is obtaining a mapping between states and actions, called policy. The policy chooses actions in given states that maximize the cumulative expected reward [3].

2.4.3 Evaluation techniques

Evaluation techniques are fundamental for assessing the effectiveness and efficiency of ML models. They provide an empirical measure of how the model is likely to perform on unseen data, highlighting its real-world applicability. These techniques involve partitioning the available data into distinct sets for training, validation, and testing, employing different strategies to ensure a robust evaluation. Each technique offers a unique approach to data splitting and utilization, aiming to provide a comprehensive view of model performance and computational efficiency. Prevalent evaluation techniques in ML include TTS (2.4.3.1) and K-FCV (2.4.3.2).

2.4.3.1 Train-Test Split

The TTS is a fundamental technique in statistical learning to evaluate the performance of predictive models. The detailed procedure is as follows:

1. **Data splitting:** Divide the entire dataset into two mutually exclusive sets. These are typically named the training set and the testing set. A common ratio is 70 : 30 or 80 : 20, where the larger portion is used for training and the smaller portion for testing;

2. **Model training:** Using the training set, the predictive model learns the relationship between the feature variables (also called predictors, independent variables, inputs) and the target variable (also called outcome, dependent variable, output);

3. **Model testing:** Apply the trained model to the test set. The model uses the feature variables in the test set to predict corresponding target variables;

4. **Performance evaluation:** Compare the predicted target variables to the actual target variables in the test set. The discrepancy between these values gives a measure of model performance. Several performance metrics can be obtained during this step.

2.4.3.2 K-Fold Cross Validation

K-FCV is a robust and widely-used evaluation technique that provides more comprehensive performance metrics, allowing to assess the model's generalizability. Here are the steps involved in this process:

1. **Data partitioning:** Split the entire dataset into K equally sized subsets or folds. The choice of K is usually 5 or 10, but it can be any integer value less than the total number of data points;

2. **Model training and validation:** Perform K separate learning experiments. In each experiment, choose one fold as the validation set, and the remaining $K-1$ folds together form the training set. Train the model on the training set and validate the model on the validation set;

3. **Performance evaluation:** After K experiments, K different performance measures are obtained. The final performance measure is the average of these measures. This gives a more comprehensive view of the model's performance across different subsets of the data, allowing to better assess its generalizability.

2.4.4 Performance metrics

Performance metrics serve as fundamental instruments for assessing and comparing the effectiveness of ML models. Each metric imparts crucial information about varied aspects of a model's predictive capability. Key performance metrics in ML include Accuracy (2.4.4.1), Precision (2.4.4.2), Recall (2.4.4.3), Specificity (2.4.4.4), F1 Score (2.4.4.5) and Area Under the Receiver Operating Characteristic Curve (AUC-ROC) (2.4.4.6).

2.4.4.1 Accuracy

Accuracy, as given by Equation (2.5), is a measure of the overall correct predictions out of all predictions made by the model.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{True\ Positives + True\ Negatives + False\ Positives + False\ Negatives} \tag{2.5}$$

2.4.4.2 Precision

Precision, shown in Equation (2.6), quantifies the proportion of true positive predictions out of all positive predictions. It shows how precise the model is in predicting positive instances.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{2.6}$$

2.4.4.3 Recall

Recall, also known as sensitivity, defined in Equation (2.7), is the proportion of actual positive instances that were correctly identified.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{2.7}$$

2.4.4.4 Specificity

Specificity, as described by Equation (2.8), is the proportion of actual negative instances that were correctly identified.

$$Specificity = \frac{True\ Negatives}{True\ Negatives + False\ Positives} \tag{2.8}$$

2.4.4.5 F1 Score

The F1 Score, shown in Equation (2.9), is the harmonic mean of Precision and Recall, providing a balance between these two metrics.

$$F1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{2.9}$$

2.4.4.6 Area Under the Receiver Operating Characteristic Curve

AUC-ROC provides an aggregate measure of model performance across all possible classification thresholds. This measure does not have a simple formula like the other metrics, but is calculated by plotting the Recall (True Positive Rate) against the False Positive Rate (1 - Specificity) at various threshold settings, then numerically integrating to find the area under the curve.

2.4.5 Computational metrics

Computational metrics provide valuable insights into the practical aspects of implementing and operating an ML model. Understanding these metrics is crucial to ensure operational efficiency and compatibility with the available computational resources and performance requirements. Several computational metrics were explored, namely the Number of trainable parameters (2.4.5.1), Number of training epochs (2.4.5.2), Training time (2.4.5.3), Average training time per epoch (2.4.5.4), Best epoch (2.4.5.5), and Inference time (2.4.5.6).

2.4.5.1 Number of trainable parameters

The number of trainable parameters in a model represents the amount of learning capacity the model has. Having more parameters makes the model more flexible in fitting a wide range of functions, but can also lead to overfitting if not properly regulated.

2.4.5.2 Number of training epochs

The number of training epochs represents the number of times the learning algorithm will work through the entire training dataset. One epoch means that each sample in the training dataset has had an opportunity to update the internal model parameters.

2.4.5.3 Training time

Training time is the total amount of time that the model spends in the training phase. This is dependent on numerous factors including the size of the dataset, the complexity of the model, and the hardware capabilities.

2.4.5.4 Average training time per epoch

Average training time per epoch can be calculated as the total training time divided by the number of epochs, as shown in Equation (2.10):

$$Average\ Training\ Time\ Per\ Epoch = \frac{Total\ Training\ Time}{Number\ of\ Epochs} \tag{2.10}$$

2.4.5.5 Best epoch

The best epoch is the epoch number at which the model performed the best on the validation set during training. This is usually where the model achieves the best balance between learning the training data and generalizing to unseen data.

2.4.5.6 Inference time

Inference time is the time that the model takes to make a decision after it has been trained. This is a critical measure in many applications where decisions need to be made in real time.

2.5 Feature engineering

Feature engineering is a key process in ML that involves creating new input variables or modifying existing ones to enhance the performance of the models. It encompasses the extraction, transformation, and selection of features, and is instrumental in improving a model's effectiveness. In this context, a feature refers to an individual, measurable property or characteristic of the phenomenon being observed. Feature extraction (2.5.1) and feature selection (2.5.2) are two fundamentals aspects of the feature engineering process, which will be addressed in this Subsection.

2.5.1 Feature extraction

Feature extraction is a crucial step during the process of feature engineering. This technique involves extracting significant characteristics from the raw audio signals, enabling a deeper understanding and analysis of the data. A comprehensive set of commonly utilized feature extraction techniques implemented in *Librosa* was explored [41]. Each technique focuses on capturing specific aspects of the audio signals, providing a diverse range of information for further analysis.

2.5.1.1 Zero Crossing Rate

Description: Zero Crossing Rate (ZCR) is a feature used in audio signal processing to quantify the rate at which a signal changes its sign. It provides information about the temporal characteristics and the amount of waveform fluctuations in the signal [42].

Calculation: The ZCR is calculated by counting the number of times the signal crosses the zero axis within a given time frame or signal segment. Mathematically, it can be expressed as:

$$\text{ZCR} = \frac{1}{N}\sum_{n=1}^{N} |sgn(x[n]) - sgn(x[n-1])| \tag{2.11}$$

where:

- $x[n]$: input signal;
- N: total number of samples.

2.5.1.2 Root-Mean-Square Energy

Description: Root-Mean-Square (RMS) Energy is a feature used in audio signal processing to quantify the overall energy or amplitude of a signal. It provides information about the signal's power distribution and is commonly used to measure loudness or intensity variations over time. The RMS Energy is a useful feature for various speech and audio analysis tasks, such as speech recognition, speaker identification, and audio classification [42].

Calculation: The RMS Energy is computed by taking the square root of the average of the squared amplitudes of the signal samples over a given time frame or signal segment. Mathematically, it can be expressed as:

$$\text{RMS} = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1} |x[n]|^2} \tag{2.12}$$

where:

- $x[n]$: input signal;
- N: total number of samples.

2.5.1.3 Short-time Fourier Transform

Description: The Short-Time Fourier Transform (STFT) is a mathematical technique used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time. It is a fundamental tool in the field of signal processing [42, 41, 43].

Calculation: The STFT is calculated by segmenting the signal into smaller, overlapping windows and then applying the Fourier Transform to each of these windows. This process can be formally represented as:

$$\text{STFT}\{x[n]\}(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n} \tag{2.13}$$

where:

- $x[n]$: input signal;
- $w[n-m]$: window function centered around m;
- ω: frequency variable.

2.5.1.4 Mel Spectrogram

Description: A Mel Spectrogram is a spectrogram where the frequencies are converted to the Mel scale. It is a common way to represent a speech signal in the domain of speech processing. The Mel scale approximates the human ear's response to different frequencies, making Mel spectrograms more perceptually meaningful [42, 41, 36, 44].

Calculation: The Mel spectrogram is computed in several steps. First, the STFT is computed to obtain the spectrogram of the signal x. Then, the PSD of the spectrogram is calculated. Next, the Mel filter bank, which is a set of triangular filters designed to mimic the HAS, is applied to the PSD spectrogram. The logarithm of the energy in each Mel filter is then taken. This process can be summarized as:

$$\text{Mel Spectrogram} = log(\text{Mel Filter Bank}(\text{Power Spectrum}(x))) \tag{2.14}$$

2.5.1.5 Short-Time Fourier Transform Chromagram

Description: The STFT Chromagram is a representation that captures the evolving tonal characteristics of an audio signal over time. It provides a concise summary of the harmonic content by quantifying the energy distribution across pitch classes. By examining the relative presence of each pitch class, the STFT Chromagram offers valuable insights into the tonal composition of the signal. [42, 41, 43].

Calculation: The STFT Chromagram is computed by first obtaining the STFT of the signal x. Then, the magnitudes of the Fourier coefficients are mapped to their respective pitch classes in the chromatic scale (12 equally tempered pitches per octave in Western music), usually C, C#, D, D#, E, F, F#, G, G#, A, A#, B. This mapping is achieved by considering the frequency of each Fourier coefficient and attributing it to the nearest pitch class, as follows:

$$\text{STFT Chromagram}_i = \sum_j |\text{STFT}(x)_j| \quad \text{for all } j \text{ such that } f_j \text{ maps to pitch class } i \tag{2.15}$$

where:

- STFT Chromagram$_i$: energy of the i-th pitch class;
- STFT$(x)_j$: j-th Fourier coefficient;
- f_j: frequency of the j-th Fourier coefficient.

2.5.1.6 Constant-Q Transform Chromagram

Description: A Constant-Q Transform (CQT) Chromagram is a compact representation of an audio signal that illustrates the evolution of energy for the 12 pitch classes of the chromatic scale over time. Unlike the STFT Chromagram, it employs the CQT instead of the STFT. The CQT utilizes a logarithmically spaced frequency axis, which aligns more closely with the human perception of pitch. This representation provides valuable insights into the frequency content and pitch characteristics of the audio signal [42, 41, 45].

Calculation: The CQT Chromagram is computed by first applying the CQT to the signal x.The magnitudes of the CQT coefficients are then mapped to their respective pitch classes in the chromatic scale, much like the STFT Chromagram.

$$\text{CQT Chromagram}_i = \sum_j |\text{CQT}(x)_j| \quad \text{for all } j \text{ such that } f_j \text{ maps to pitch class } i \tag{2.16}$$

where:

- CQT Chromagram$_i$: energy of the i-th pitch class;
- CQT$(x)_j$: j-th CQT coefficient;
- f_j: frequency of the j-th CQT coefficient.

2.5.1.7 Chroma Energy Normalized Statistics

Description: Chroma Energy Normalized Statistics (CENS) is a feature representation technique used in music information retrieval. Derived from the Chromagram, it provides a robust summary of the 12 pitch classes' energy over time. By applying normalization and statistical measures, CENS minimizes the impact of dynamic variations. [42].

Calculation: CENS features are computed through a series of transformations applied to a Chromagram of an audio signal. First, the Chromagram is computed (usually using either the STFT or the CQT). Next, a temporal smoothing operation is performed, typically by computing a moving average over the chroma vectors. Then, the chroma vectors are normalized to have an L1-norm of 1. Finally, the normalized chroma vectors are downsampled, and each bin of the resulting vectors is quantized into a small number of levels. The following Equation (2.17) summarizes the process:

$$\text{CENS} = \text{Quantization}(\text{Downsampling}(\text{L1 Normalization}(\text{Smoothing}(\text{Chromagram})))) \quad (2.17)$$

2.5.1.8 Spectral Centroid

Description: The Spectral Centroid is a measure that characterizes the center of energy distribution across the frequency range in a signal [42, 41, 46].

Calculation: The Spectral Centroid is calculated by weighting each frequency bin in the spectrum by its magnitude or power and computing the weighted average of these frequencies.

$$\text{Spectral Centroid} = \frac{\sum_{k=0}^{N-1} f_k \cdot x_k}{\sum_{k=0}^{N-1} x_k} \quad (2.18)$$

where:

- N: total number of frequency bins in the spectrum;
- x_k: spectrum value of the frequency bin at index k;
- f_k: frequency value at index k, in Hz.

2.5.1.9 Spectral Bandwidth

Description: The Spectral Bandwidth is a measure used in signal processing to quantify the spread or width of a spectrum. It provides information about the frequency range covered by the signal's power spectrum [42, 41, 46].

Calculation: The spectral bandwidth is typically computed as the second central moment of the spectrum, weighted by the squared magnitude of the frequencies, as follows:

$$\text{Spectral Bandwidth} = \sqrt{\frac{\sum_{k=0}^{N-1} (f_k - \text{Spectral Centroid})^2 \cdot x_k}{\sum_{k=0}^{N-1} x_k}} \quad (2.19)$$

where:

- N: total number of frequency bins in the spectrum;
- x_k: spectrum value of the frequency bin at index k;

- f_k: frequency value at index k in Hz.

2.5.1.10 Spectral Contrast

Description: Spectral Contrast is a feature used in signal processing to measure the difference in magnitudes between peaks and valleys in a frequency spectrum. [42, 41, 47].

Calculation: The calculation involves performing the Fast Fourier Transform (FFT) to obtain spectral components, which are then divided into sub-bands based on octaves. Spectral Contrast is calculated for each sub-band. The raw Spectral Contrast feature measures the intensity of spectral peaks, valleys, and their differences in each sub-band. To ensure stability of the feature, the strength of spectral peaks and valleys is estimated using the average value in a small neighborhood around the maximum and minimum values, rather than the precise maximum and minimum values themselves. This small neighborhood is described by a parameter called the neighborhood factor α [47].

$$\text{Peak}_k = \log\left(\frac{1}{\alpha N}\sum_{i=1}^{\alpha N} x'_{k,i}\right) \tag{2.20}$$

$$\text{Valley}_k = \log\left(\frac{1}{\alpha N}\sum_{i=1}^{\alpha N} x'_{k,N-i+1}\right) \tag{2.21}$$

$$\text{SpectralContrast}_k = \text{Peak}_k - \text{Valley}_k \tag{2.22}$$

where:

- $x'_{k,i}$: sorted FFT vector element at index i in the k-th sub-band;
- α: parameter controlling the fraction of sorted vector elements to consider for peak estimation;
- N: total number of elements in the k-th sub-band;
- $x'_{k,N-i+1}$: sorted FFT vector element at index $N-i+1$ (in reverse order) in the k-th sub-band;
- Peak_k: peak strength in the k-th sub-band;
- Valley_k: valley strength in the k-th sub-band.

2.5.1.11 Spectral Flatness

Description: Spectral Flatness is a measure used in signal processing which indicates the balance between the energy in the harmonic and non-harmonic components of the spectrum [42].

Calculation: The spectral flatness is calculated by comparing the geometric mean to the arithmetic mean of the magnitudes of the frequency spectrum, as follows:

$$\text{Spectral Flatness} = \frac{\exp\left(\frac{1}{N}\sum_{k=0}^{N-1}\ln(x_k)\right)}{\frac{1}{N}\sum_{k=0}^{N-1}x_k} \tag{2.23}$$

where:

- N: total number of frequency bins in the magnitude spectrum;
- x_k: magnitude value at frequency bin k.

2.5.1.12 Spectral Rolloff

Description: Spectral Rolloff is a feature used in signal processing to measure the frequency below which a specified percentage of the total spectral energy is concentrated. It provides information about the spectral shape of the signal [42, 41, 46].

Calculation: The Spectral Rolloff is typically defined as the frequency below which a certain percentage of the total spectral energy lies, as follows:

$$\text{Rolloff} = f_i \text{ such that } \sum_{k=0}^{i}|x_k| = \kappa\sum_{k=0}^{N-1}|x_k| \tag{2.24}$$

where:

- f_i: frequency value at index i, in Hz;
- x_k: spectral value at bin k;
- N: total number of frequency bins in the magnitude spectrum;
- κ: specified energy threshold, usually 95% or 85%.

2.5.1.13 Tonnetz Features

Description: Tonnetz features are a set of musical features used to analyze and represent harmonic relationships between musical chords or notes. They are based on the concept of the Tonnetz, a geometric representation of musical pitch classes [42, 41, 48].

Calculation: Tonnetz features are typically computed using the Tonnetz representation, which arranges the pitch classes in a lattice-like structure. The relationships between pitch classes in the Tonnetz can be quantified using various metrics, such as euclidean distances or angular distances. Tonnetz features can be derived from these metrics, capturing different aspects of harmonic relationships. The Tonnetz features in *Librosa* are calculated as described in [48].

2.5.1.14 Mel-frequency cepstral coefficients

Description: Mel-Frequency Cepstral Coefficients (MFCCs) are widely used features in speech and audio signal processing. They are designed to capture the characteristics of the HAS by modeling the perceptual properties of speech sounds. MFCCs provide a compact representation of the spectral envelope of a signal, making them useful for various speech-related tasks [42, 41, 36, 44].

Calculation: The calculation of MFCCs involves several steps. First, the audio signal is divided into short frames. Then, the power spectrum of each frame is computed using techniques like the FFT. The resulting spectrum is then transformed using a Mel Filterbank, which groups frequencies according to the Mel scale, mimicking the non-linear frequency resolution of human hearing. Next, the logarithm of the filterbank energies is taken, and the Discrete Cosine Transform (DCT) is applied to decorrelate the coefficients. Finally, a subset of the resulting DCT coefficients (MFCCs) is retained, typically discarding the higher-frequency coefficients that contain less perceptually relevant information. The calculation of MFCCs can be summarized by the Equation (2.25).

$$\text{MFCCs} = \text{DCT}\left(\log\left(\text{Mel Filter Bank}\left(\text{Power Spectrum}\left(\text{Frame}\right)\right)\right)\right) \tag{2.25}$$

2.5.1.15 Mel-frequency Cepstral Coefficients Delta

Description: MFCCs Delta are a time-based derivative of MFCCs. They are commonly used as supplementary features to capture the temporal dynamics or rate of change of the MFCCs. MFCCs Delta features provide information about the speech signal's spectral variations over time, enhancing the discriminative power of MFCCs for speech-related tasks [42, 41, 36, 44].

Calculation: The calculation of MFCCs Delta involves estimating the rate of change of MFCCs over time. This is typically done by applying a sliding window (e.g., 5 frames) to a sequence of MFCCs frames. A weighted linear regression is then performed on the MFCCs frames within the window to estimate the slope. The resulting slope values represent the MFCCs Delta coefficients.

2.5.1.16 Mel-frequency Cepstral Coefficients Delta Delta

Description: MFCCs Delta Delta are the second-order derivatives of MFCCs. They capture the rate of change of temporal dynamics of the MFCCs over time, providing additional temporal information beyond MFCCs and MFCCs Delta. MFCCs Delta Delta features are commonly used in speech processing tasks where the dynamics of the spectral features play a significant role [42, 41, 36, 44].

Calculation: The calculation of MFCCs Delta Delta involves estimating the rate of change of the MFCCs Delta coefficients over time. Similar to computing MFCCs Delta, a sliding window is applied to the sequence of MFCCs Delta frames. A weighted linear regression is performed on

the MFCCs Delta frames within the window to estimate the second-order slope, representing the MFCCs Delta Delta coefficients.

2.5.1.17 Polynomial Features

Description: Polynomial Features are designed to capture non-linear interactions in the frequency domain and can be useful for signal processing tasks [42, 41].

Calculation: The calculation of Polyfeatures involves computing the coefficients of fitting an nth-order polynomial to the columns of a spectrogram, using the least-squares method. These coefficients represent the non-linear relationships between frequencies and powers and can capture complex spectral patterns.

2.5.2 Feature selection

Feature selection plays a vital role in data analysis and ML by identifying a subset of relevant features from a larger set. The primary objective is to choose the most informative and discriminative features that significantly contribute to improving model performance, reducing computational complexity, and enhancing interpretability. To achieve this goal, researchers employ a variety of techniques and metrics in the field of feature selection, namely Pearson Correlation Coefficient (PCC) (2.5.2.1), Spearman Correlation Coefficient (SCC) (2.5.2.2), Analysis of Variance (ANOVA) F-value (2.5.2.3), and Random Forest Importance (RFI) (2.5.2.4). These approaches enable the systematic evaluation of feature relevance and discriminatory power, facilitating the identification of a subset of features that exhibit superior predictive capabilities and provide meaningful insights.

2.5.2.1 Pearson correlation coefficient

The PCC is a statistical measure used to quantify the strength and direction of the linear relationship between two variables. The PCC for 2 variables with raw scores X and Y is given by Equation (2.26):

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} \tag{2.26}$$

where:

- cov: covariance;
- σ_X: standard deviation of X;
- σ_Y: standard deviation of Y.

The resulting PCC r_{xy} ranges from -1 to 1, where a value close to 1 indicates a strong positive linear correlation, a value close to -1 indicates a strong negative linear correlation, and a value close to 0 suggests no linear correlation between the feature and the target variable.

2.5.2.2 Spearman correlation coefficient

The SCC measures the rank-based association between two variables, emphasizing their relative positions in a dataset rather than their absolute values. This metric evaluates the strength and direction of a monotonic relationship, where an increase in one variable's values corresponds to an increase (or decrease) in the other's rankings consistently.

The SCC is essentially the PCC applied to rank variables. For 2 variables with raw scores X, Y, these scores are transformed into ranks, denoted as $\mathrm{R}(X), \mathrm{R}(Y)$. The SCC, represented as r_s, is then calculated as:

$$r_s = \rho_{\mathrm{R}(X),\mathrm{R}(Y)} = \frac{\mathrm{cov}(\mathrm{R}(X), \mathrm{R}(Y))}{\sigma_{\mathrm{R}(X)} \sigma_{\mathrm{R}(Y)}} \tag{2.27}$$

where:

- $\rho_{\mathrm{R}(X),\mathrm{R}(Y)}$: PCC applied to the rank variables;
- $\mathrm{cov}(\mathrm{R}(X), \mathrm{R}(Y))$: covariance of the rank variables;
- $\sigma_{\mathrm{R}(X)}$ and $\sigma_{\mathrm{R}(Y)}$: standard deviations of the rank variables.

The SCC lies between -1 and 1, with 1 suggesting a perfect monotonic increasing relationship, -1 indicating a perfect monotonic decreasing relationship, and values near 0 implying minimal monotonic association.

2.5.2.3 ANOVA F-value

The ANOVA F-value is a statistical measure used to assess the significance of class differences for a specific feature in the analysis of multiple classes. It quantifies the portion of variance explained by different classes relative to the variance within those classes. The F-value serves as a quantitative measure of the potential importance or relevance of a feature in differentiating the classes.

For a given feature with values X_{ij} where i denotes the class and j denotes the individual observation within the class, the ANOVA F-value, denoted as F, is computed as:

$$F = \frac{\text{Between-group variance}}{\text{Within-group variance}} = \frac{\frac{1}{k-1}\sum_{i=1}^{k} n_i(\bar{X}_i - \bar{X})^2}{\frac{1}{N-k}\sum_{i=1}^{k}\sum_{j=1}^{n_i}(X_{ij} - \bar{X}_i)^2} \tag{2.28}$$

where:

- $\bar{X}_i$: mean of class i;
- $\bar{X}$: overall mean;
- n_i: number of observations in class i;
- N: total number of observations;

- k: number of classes.

A higher F-value signifies a stronger relationship between the feature and the classes, indicating a greater potential for the feature to be informative in the context of the analysis.

2.5.2.4 Random Forest Importance

The impurity-based RFI technique is utilized to determine the importance of each feature in the feature selection process.

To compute the feature importance scores:

- A Random Forest Classifier is trained on the available data to compute the feature importance scores;
- The Random Forest model constructs an ensemble of decision trees during training;
- Each decision tree is trained on a bootstrap sample of the data with a random subset of features for each split;
- The decision trees evaluate the importance of each feature based on impurity measures, such as Gini impurity or entropy, when making splits;
- The impurity-based importance scores reflect the impact of each feature on the overall impurity reduction in the decision trees;
- Importance scores for each feature are computed by averaging the impurity-based importance across all decision trees in the ensemble;
- Features that lead to a larger decrease in impurity are assigned higher importance scores.

By examining these feature importance scores, researchers can identify the features that have the most significant impact on the overall predictive performance of the Random Forest model. This information aids in the feature selection process by guiding the selection of the most informative and influential features for subsequent analyses and modeling.

2.6 Voicing decision in whispered-to-normal speech conversion systems

The focus of this research is to develop a DL-based classifier subsystem that allows the segmentation of WS based on two phone classes — CTV and NCTV. This process is commonly known in the literature as VD. It ensures that the broader whispered-to-normal speech conversion system only implants a replacement for the missing periodic signal component in regions of the WS that would be V in NS. The UV regions of WS should remain untouched after the conversion. For the purposes of this research:

1. All the vowels fall in the phone class CTV, since they are always V in NS;
2. The consonants that are V and UV in NS fall in the phone classes CTV and NCTV, respectively.

By enhancing VD in whispered-to-normal speech conversion systems, this research aims to improve the overall quality and naturalness of the converted speech.

2.7 Chapter summary

This Chapter *"Background"* provided foundational knowledge on various topics related to the research. It covered the HSPS (2.1), the HAS (2.2), speech signal analysis and modelling techniques (2.3), an introduction to DL (2.4), and feature engineering (2.5). Additionally, it focused on the VD in whispered-to-normal speech conversion systems (2.6).

In the next Chapter *"Voicing decision approaches — a review"* (3), a comprehensive review of VD approaches will be presented, encompassing a discussion on the criteria and process for paper selection (3.1) and a detailed review of the selected papers (3.2). The selected papers are categorized into rule-based (3.2.1), ML-based (3.2.2), and hybrid (3.2.3) approaches.

Chapter 3

Voicing decision approaches — a review

This chapter presents the selection (3.1) and review (3.2) of state-of-the-art articles which propose different approaches to VD.

3.1 Paper selection

A search on academic literature indexers was conducted, namely on *Scopus* and *Google Scholar*. The key terms utilized during the search encompassed *"voice decision"* or *"voicing decision"*, *"candidate to voice"* or *"candidate to voicing"*, and *"unvoiced detection"* or *"voiceless detection"*. These terms were always combined with *"whispered speech"* or *"whisper speech"*. Several papers that describe VD systems applied to whispered-to-normal speech conversion were retrieved. The papers that present approaches for VD that are decoupled from the rest of the whispered-to-normal conversion system were selected.

3.2 Review of the selected papers

The selected papers were categorized as rule-based (3.2.1), ML-based (3.2.2) or hybrid (3.2.3) approaches. They are reviewed next. Table A.1 provides a concise summary of the retrieved information, encompassing the title, reference, year, description, classifier, features, training data, evaluation, advantages, and disadvantages of the scientific papers.

3.2.1 Rule-based approaches

Rule-based approaches for VD rely on predefined thresholds applied to speech signal features, which can limit their ability to generalize. The effectiveness of these approaches heavily relies on domain expertise and often leads to the development of low complexity systems. Papers which describe rule-based approaches for the problem under study were reviewed.

3.2.1.1 Rule-based classifier using spectral centroid thresholding

Summary: In the paper *"Glottal flow synthesis for whisper-to-speech conversion"* [49], it is described a rule-based VD approach based on the spectrum center of gravity of the speech signal, which corresponds to the spectral centroid of the power spectrum.

Broadband sounds and low energy regions are considered evidence for the presence of consonants and vowels, respectively. The most prominent formants produced during V speech tend to lie below 4 kHz. Therefore, whisper vocalic sounds are expected to exhibit a center of gravity below this threshold. Conversely, UV consonants fill the high-frequency spectrum. Thus, it is expected that they exhibit a center of gravity much higher than the mentioned threshold.

The authors defined the V/UV frequency as a threshold to distinguish between vowels and consonants. The VD value is 0 when the center of gravity is below this threshold. An UV frequency is also defined as the geometric mean of the values of spectrum center of gravity above the V/UV frequency. The VD value is 1 when the spectrum center of gravity is above this threshold.

Variations of center of gravity from vocalic sound to consonants are sometimes slower, resulting in slow VD transitions. To surpass this problem, it is proposed that the mapping between the threshold frequencies and the VD is done non-linearly, based on the sigmoid function.

Objective evaluation was performed, allowing to obtain the V, UV and total error values of 7.6%, 10.1% and 17.7%, respectively.

Critical analysis: The proposed approach exhibits low complexity, making it potentially suitable for real-time applications. However, its rule-based nature may compromise the overall effectiveness and generalization capability of the subsystem. In comparison to other methods for VD, the error rates obtained with this approach are relatively high. Yet, this paper lacks a more comprehensive quantitative evaluation of the VD subsystem, including performance and computational metrics such as Recall, Precision, and Inference Time, which are essential for a thorough assessment of its effectiveness and efficiency.

3.2.1.2 Rule-based classifier using temporal and frequency-band energy variations thresholding (1)

Summary: In the paper *"Implantation of voicing on whispered speech using frequency-domain parametric modelling of source and filter information"* [50], a rule-based approach for obtaining a VD is described.

The WS is segmented in silent, plosive, sibilant, fricative and V regions. Silence detection is implemented by monitoring the dynamics of the absolute energy of the signal using short-time analysis. Plosive detection is implemented by combining a criterion based on the phase provided by short-time analysis and a criterion testing the gradient of the signal energy across time. Sibilants are detected by evaluating the ratio of the signal energy above and below $2,800\ Hz$. Fricatives are detected when the signal is not silence, is not classified as a sibilant, and when the ratio between

the energy concentration in the range $2,000\ Hz - 4,500\ Hz$, and the energy concentration above $4,500\ Hz$, does not exceed a predefined threshold.

Critical analysis: The proposed approach is distinguished by its commendable focus on achieving low computational complexity, resulting in efficient computational performance. In the experimental evaluation, the approach is effectively demonstrated using a single word as an example. However, to provide a more comprehensive understanding, it would be valuable to include a comparative analysis with other state-of-the-art approaches. Incorporating such a comparison would enhance the study by shedding light on the relative strengths and weaknesses of the proposed method within the broader landscape of related research. It would contribute to a more comprehensive evaluation of the approach and facilitate a deeper understanding of its potential impact.

3.2.1.3 Rule-based classifier using temporal and frequency-band energy variations thresholding (2)

Summary: In the paper *"Reconstruction of normal sounding speech for laryngectomy patients through a modified Code-Excited Linear Prediction (CELP) codec"* [51], an approach is described for frame-level VD.

Fricatives are detected by comparing the power of whispered frames in bandwidths above and below $3\ kHz$. Then, a set of band pass filters compares signal energy ratios in small bands of high and low frequency to identify plosives and vowels. Energy concentration in $1-3\ kHz$ range, in comparison with $6-7.5\ kHz$, is considered a possible indicator of a vowel sound. Furthermore, other information, such as detecting the energy burst after a small silence, is considered as evidence of a plosive. Plosives are confirmed by comparing signal energy ratios in small bands of low and high frequency, as well as considering the small silence (low energy) in previous segment to confirm the decision.

Critical analysis: Despite the authors stating that the VD approach proposed in their study led to an improvement in the speech reconstruction system in terms of Mean Opinion Score (MOS), they have not presented any performance or computational metrics to enable a thorough evaluation of the effectiveness and efficiency of the VD subsystem. Additionally, they have not compared their approach with other state-of-the-art methods. It is worth noting that the proposed approach has low complexity. However, because it is rule-based, there may be concerns about the robustness and generalization capability of the classifier.

3.2.2 Machine learning-based approaches

ML approaches leverage ML models to perform the VD task. Their success often depends on the quantity and quality of data available for training purposes. Usually, they result in higher complexity systems, with higher generalization capabilities, relatively to rule-based approaches. Papers which describe ML approaches for VD were reviewed.

3.2.2.1 BLSTM classifier trained with MFCCs, velocity and acceleration features

Summary: In the paper *"Whispered speech to neutral speech conversion using bidirectional LSTMs"* [29], a bidirectional LSTM model is employed to predict the VD.

The model is trained using the following features: MFCCs, delta and delta delta computed from the smooth spectrum of WS. The excitation parameter is obtained from Speech Transformation and Representation using Adaptive Interpolation of Weighted Spectrum (STRAIGHT) analysis of neutral speech. The training database consists in parallel data of WS and NS: 60 sentences taken from the Multilingual, Open-source Corpus of Heterogeneous Acoustic data (MOCHA) database (an extension of Texas Instruments/Massachusetts Institute of Technology Acoustic-Phonetic Continuous Speech Corpus (TIMIT)) were spoken by 3 male and 3 female speakers in both normal and whispered modes.

From an objective evaluation, it is concluded that the bidirectional LSTM based VD error value is inferior to the obtained using the baseline Deep Neural Network (DNN) based scheme, with a value of about 8%.

Critical analysis: This approach showcases a promisingly low error rate in objective evaluation, indicating its potential effectiveness. However, the lack of other quantitative evaluation metrics hinders a comprehensive assessment of the performance and computational efficiency of the VDs. The utilization of a DNN model in this approach is advantageous due to its inherent generalization capabilities. Nevertheless, it is crucial to recognize the inherent complexity associated with a LSTM-based system and the necessity for abundant training data to achieve satisfactory results.

Furthermore, it is worth noting that the bidirectional nature of the network assumes access to future audio data, which renders it unsuitable for online usage scenarios where such data is unavailable. This limitation should be taken into consideration when considering the practical applicability of the proposed approach.

3.2.2.2 DNN classifier, trained with MFCCs features computed from data driven colored noises dictionary

Summary: In the paper *"A robust voiced/unvoiced phoneme classification from whispered speech using the "color" of whispered phonemes and deep neural network"* [30], a method to perform frame level VD on WS was described.

It was hypothesized that a WS spectrum could be represented as a linear combination of a set of colored noise spectra. Then, a five-dimensional feature is computed by employing non-negative matrix factorization with a fixed basis dictionary, constructed using spectra of five colored noises. A DNN is used as a classifier, resorting to the proposed feature. For training purposes, an in-house annotated WS database was used, consisting of about 450 phonetically balanced sentences red from the MOCHA-TIMIT database.

Objective evaluation was performed. The proposed 5D feature is compared to two baseline features: MFCCs and features computed from a data driven dictionary. The following values were

obtained for V, UV and average accuracies: 73.63%, 78.51% and 76.06% using the MFCCs-DNN scheme; 73.81%, 74.78% and 74.29% employing the Combined-DNN (5D and MFCCs) scheme. The scheme using only the MFCCs allowed to obtain the best average VD accuracy.

Critical analysis: Upon integrating the proposed 5D feature with the MFCCs, the accuracy of frame-level V/UV classification achieved a reasonable balance between the two classes, albeit relatively lower compared to alternative methods. The absence of additional quantitative evaluation metrics impedes a thorough evaluation of the performance and computational efficiency of the VDs. It is crucial to recognize the inherent complexity associated with a DNN-based system and the necessity for abundant training data to achieve satisfactory results.

3.2.2.3 SVM and GMM classifiers, trained with mel-cepstra static and dynamic features

Summary: In the paper "Whisper-to-speech conversion using restricted boltzmann machine arrays" [31], two ML models are used to obtain a VD: a Gaussian Mixture Model (GMM) and a Support Vector Machine (SVM).

Each model is trained using the mel-cepstra static and dynamic features of WS with V/UV data from Dynamic Time Warping (DTW) aligned NS. Approximately 180,000 frames of parallel WS and NS recordings from Whispered Texas Instruments/Massachusetts Institute of Technology Acoustic-Phonetic Continuous Speech Corpus (wTIMIT) database were used for training purposes.

The models were evaluated, using 10,000 frames of testing data from the same dataset. The VD errors were obtained for different lengths of concatenated GMM and SVM input vectors. This evaluation revealed that the optimal context size for the GMM model is ± 3 frames, enabling to achieve a V error of 5.09%, an UV error of 3.77% and a total VD error of 8.86%. This evaluation was repeated for the SVM model. With an optimal context size of ± 5 frames, it was possible to obtain a V error of 4.39%, an UV error of 5.08% and a total VD error of 9.47%. The VD error rate obtained using the GMM was slightly lower.

Critical analysis: The approach employed in this study demonstrates a commendable achievement with a low error rate. By leveraging machine learning (ML), this methodology exhibits potential for generalization. However, the absence of supplementary quantitative evaluation metrics hinders a comprehensive assessment of the VDs' performance and computational efficiency. It is imperative to acknowledge the inherent complexity of a (DNN)-based system and the indispensability of ample training data to attain satisfactory results.

3.2.2.4 FNN classifier, trained using spectral features of whispered and normal speech

Summary: In the paper *"Improvement to a nam-captured whisper-to-speech system"* [32], a Feed Forward Neural Network (FNN) is used to predict the segments from the WS. The continuous output is then converted to a binary VD.

MFCCs are used as spectral feature at each frame. The spectral segment features of WS are constructed by concatenating feature vectors at each current whispered frame ± 8 frames, in order to capture context. Then, the vector dimension is reduced using a Principal Component Analysis (PCA) technique. The authors considered an excitation feature, characterized by the log-scaled fundamental frequency, extracted with fixed-point analysis, and by 5 average dB values of aperiodic components on five frequency bands. The FNN is trained using features obtained from 200 utterance pairs of WS and NS, verbalized by a French native male speaker.

The VD errors were evaluated. The V error was 2.4%, the UV error equaled 4.4% and the total VD error was 6.8%. With the integration of this dedicated VD subsystem, the VD error of the whispered-to-normal speech converter diminished by 2.4%, relatively to the original approach (9.2%).

Critical analysis: The study's approach attained a low error rate in VDs. However, the absence of supplementary evaluation metrics limits a comprehensive assessment of the VD subsystem's performance and computational efficiency. The complexity of the FNN used and the reliance on a large training dataset are noteworthy limitations. Despite these drawbacks, the approach shows potential for generalization and practical application.

3.2.3 Hybrid approaches

Hybrid approaches attempt to estimate a VD, leveraging techniques from both ML and rule-based approaches. Papers which describe hybrid approaches for VD were reviewed.

3.2.3.1 KNN phoneme classification followed by rule-based voicing decision using spectral centroid thresholding

Summary: In the paper *"Voicing decision based on phonemes classification and spectral moments for whisper-to-speech conversion"* [33], a low-resource VD system is proposed, suitable for real-time applications. The proposed system, starts with the classification of WS frames into phoneme classes based on their spectral centroid and spread, using the K-Nearest Neighbors (KNN) algorithm. Then, discriminates V phonemes from their UV counterpart based on class-dependent spectral centroid thresholds. The KNN algorithm is trained using an in-house database of annotated WS.

The proposed approach is compared to a simpler approach using a single centroid threshold. Objective evaluation is performed. Both approaches reach a VD accuracy higher than 91%, but the proposed approach allows avoiding some systematic VD errors. This may allow users to learn to adapt their speech in real-time, to compensate the remaining VD errors.

Critical analysis: By utilizing this approach, the need for individual system calibration was eliminated when the algorithm was trained with a multi-speaker database containing annotated read text. This resulted in a decrease in systematic VD errors for certain phonemes, thereby

creating a more appropriate control space for VD. However, the rule-based nature of the second step in this approach raises concerns about its potential impact on the robustness and ability to generalize of the VD classifier. Moreover, the lack of supplementary evaluation metrics limits a comprehensive evaluation of the performance and computational efficiency of the VD subsystem.

3.3 Chapter summary

This Chapter provided a comprehensive review of different approaches to VD. It included two sections: *"Paper selection"* (3.1) and *"Review of the selected papers"* (3.2).

In the next Chapter *"Methodology"*, the methodology of this research will be presented, focusing on following topics: hardware and software used (4.1), acquisition of a phonetically annotated whispered/normal speech dataset (4.2), dataset preprocessing (4.3), feature engineering (4.4), selection and design of DL-based model architectures (4.5), evaluation metrics definition (4.6), assessment and comparison of model/features subset pairs (4.7), and assessment of the best performing model/features subset pair (4.8).

Chapter 4

Methodology

This Chapter presents the methodology employed in the study. It outlines the steps and procedures followed to address the research objectives and answer the research question, namely hardware and software description (4.1), phonetically annotated WS/NS dataset acquisition (4.2), dataset preprocessing (4.3), feature engineering (4.4), selection and design of DL-based model architectures (4.5), definition of evaluation metrics (4.6), assessment and comparison of all model/features subset pairs (4.7), and assessment of the best performing model/feature subset pair (4.8).

4.1 Hardware and software description

The hardware (4.1.1) and software (4.1.2) used for dataset preprocessing, feature engineering, selection and design of DL-based model architectures, and assessment are described next.

4.1.1 Hardware

- **Central Processing Unit (CPU):** Intel Core i5-8300H;
- **Graphics Processing Unit (GPU):** NVIDIA GeForce GTX 1050;
- **Solid State Drive (SSD):** WD Blue SN570 1 TB NVMe SSD;
- **Random Access Memory (RAM):** VENGEANCE Series 32 GB ($2x16$ GB) DDR4 SODIMM 2666 MHz CL18 Memory Kit.

4.1.2 Software

- **Operating System (OS):** Pop!_OS 22.04 LTS;
- **Programming languages:** Python 3.10.10 [52]; Matrix Laboratory (MATLAB) R2022b [53];
- **Python packages:** Tensorflow 2.11.0 [54]; Plotly 5.14.1 [55]; SciPy 1.10.1 [56]; Scikit-learn 1.2.2 [57]; Keras 2.11.0 [58]; Keras TCN 3.5.0 [59]; Librosa 0.10.0.$post2$ [42]; Pandas 2.0.1 [60];

- **GPU drivers and libraries:** NVIDIA driver 525.89.02; Compute Unified Device Architecture Deep Neural Network library (cuDNN) 8.6.0.163; Compute Unified Device Architecture (CUDA) toolkit 11.8.0.

4.2 Phonetically annotated whispered/normal speech dataset acquisition

The process of acquiring the phonetically annotated WS/NS speech dataset involved the selection, recording, screening and training of participants (4.2.1), the corpus' design, recording protocol and dataset structure (4.2.2) and the phonetic annotation of the recorded speech (4.2.3).

4.2.1 Participants selection, recording, screening and training

This Subsection outlines the processes involved in selecting participants (4.2.1.1), setting up the recording environment and equipment (4.2.1.2), and conducting screening and training (4.2.1.3). These procedures were crucial to ensure the quality and reliability of the collected data.

4.2.1.1 Participant selection

Convenience sampling was used to recruit 17 participants (9 male and 8 female speakers) aged between 22 and 33 years from the Aveiro and Coimbra districts of Portugal. The mean age of the participants was 26 years with a standard deviation of 3 years [4]. All participants were from the North-western Dialects region of Portugal (Dialetos Setentrionais) and had not resided in other regions for extended periods of time [61].

The following inclusion criteria were employed [4]:

- No history of voice disorders;
- No vocal pathology at the time of the recordings as assessed by a voice specialist using a standardized case history form [62];
- No upper respiratory tract infection on recording day;
- EP as first language and from the center of Portugal, where the North-western Dialects (Dialetos Setentrionais) are spoken [63].

The exclusion criteria comprised [4]:

- Impairments in oro-motor structure and function;
- Use of orthodontic (correction) devices;
- Respiratory pathology;
- Laryngopharyngeal reflux;

- Fluency disorders;
- Having been submitted to vocal laryngeal surgery;
- Not being able to produce all the vocal tasks (particularly whispering).

4.2.1.2 Recording environment and equipment

The participants were situated in a quiet environment with a background noise level measuring 15.1 *dBLAeq* (A-weighted time-averaged/equivalent sound pressure level). They were then recorded using a *Sennheiser Ear Set 1* condenser microphone, which was worn on the head. The acoustic information was sampled at a rate of $48,000$ Hz with 16 *bit* resolution per sample [4].

4.2.1.3 Screening and training

A screening and training process similar to the one used in [64] was followed, to ensure that participants could distinguish and produce NS and WS accurately. As no visual representations of the glottal configurations were accessible during data collection, a voice specialist was present to perceptually observe and recognize any deviations from the intended neutral whispering, which was defined as normal adduction and medium loudness of WS [64].

4.2.2 Corpus

This Subsection outlines the design (4.2.2.1), recording (4.2.2.2) and structure (4.2.2.3) of the corpus.

4.2.2.1 Corpus design

The corpus utilized in this study was composed of a range of materials, including 4 sustained sibilants, 4 sustained oral vowels, 12 disyllabic words, 6 Consensus Auditory-Perceptual Evaluation of Voice (CAPE-V) sentences, commonly utilized by clinicians to evaluate voice quality as described in [65], and a phonetically balanced text sourced in [66].

The 4 sibilant fricatives (/s, z, ʃ, ʒ/) and the 4 oral vowels (/i, a, ɔ, u/) define the corners of the EP vowel space [67]. The 12 Consonant-Vowel-Consonant-Vowel (CVCV) disyllabic real words contained the fricatives in initial, mid and final word positions. To maintain a stable vowel height environment (ranging from open-mid to open) across the syllables, the four sibilants were combined with /a/ and /ə/, given that the most frequent syllable type in EP is Consonant-Vowel (CV) [68]. Six sentences and a phonetically balanced text commonly used in Portugal to evaluate voice quality [69] were also part of the corpus. These materials utilized the same set of vowels and fricatives (/i, a, ɔ, u, s, z, ʃ, ʒ/) [4].

4.2.2.2 Corpus recording protocol

Each task was performed 3 times using both NS and WS, except for the text task, which was performed once in each mode. WS can be more traumatic to the larynx than NS [70]. To ensure the safety of the speakers' larynx, the tasks were selected carefully to balance the information gathered on NS and WS mechanisms while avoiding vocal fatigue. A voice specialist was present during all recordings. Due to the potential difficulty and confusion caused by frequently changing speech modes, the tasks were recorded one at a time, starting with NS and then switching to WS [4]. This approach aimed to minimize any potential negative effects of frequent speech mode changes, as noted in [71].

4.2.2.3 Corpus dataset structure

The dataset comprises 54 audio files per participant, with 27 files being dedicated to NS and the remaining 27 to WS. These files incorporate:

- 4 sustained sibilants: /s, z, ʃ, ʒ/;
- 4 sustained EP oral vowels: /i, a, ɔ, u/;
- 12 CVCV disyllabic real words with sibilant fricatives in initial, medial, and final word positions, as depicted in Table 4.1;
- 6 CAPE-V phrases [65]:
  - "A Marta e o avô vivem naquele casarão rosa velho" [a ˈmaɾta i u aˈvo ˈvivɛm naˈkɛlə kazaˈraw ˈrɔzə ˈvɛju] - Production of every EP oral vowel;
  - "Sofia saiu cedo da sala" [suˈfiə sə-ˈiw ˈsɛðu də ˈsaˈla] - Easy onset with /s/ (words with /s/ at syllable onset);
  - "A asa do avião andava avariada" [a ˈaza du aviˈəw ɔ ˈdavə avariˈadə] - All V;
  - "Agora é hora de acabar" [aˈɣɔɾə e ˈɔɾə dɛ aˈkabar] - Elicits hard glottal attack;
  - "Minha mãe mandou-me embora" [ˈmiɲə ˈmamə du ˈɛmboɾə] - Nasal sounds;
  - "O Tiago comeu quatro peras" [u ˈtiagu kuˈmew ˈkuatɾu ˈpeɾas] - Weighted with voiceless stops.
- EP phonetically balanced text, *"The North Wind and the Sun"*, containing 98 words and 196 syllables [66].

4.2.3 Phonetic annotation

The phonetic annotation process encompassed the annotation of sustained and word materials (4.2.3.1), sentences and phonetically balanced text (4.2.3.2), and its reliability verification (4.2.3.3).

Table 4.1: European Portuguese disyllabic words with fricatives [4].

Fricative	Word Initial	Word Medial	Word Final
[s]	<sala> ['salɐ]	<assa> ['asɐ]	<face> 'fas
[z]	<zaro> ['zaɾu]	<asa> ['azɐ]	<vaze> 'vaz
[ʃ]	<chama> ['ʃamɐ]	<acha> ['aʃɐ]	<ache> ['aʃ]
[ʒ]	<jarra> ['ʒaɾɾɐ]	<haja> ['aʒɐ]	<laje> ['laʒ]

4.2.3.1 Sustained and word materials annotation

The boundaries of all the phones from sustained and word materials (8 sustained fricatives and oral vowels; all phones in the 12 disyllabic words) were manually annotated using previously established criteria in [72, 61], based on perceptual and acoustic analysis.

4.2.3.2 Sentence and phonetically balanced text materials annotation

Every occurrence of /i, a, ɔ, u/ and /s, z, ʃ, ʒ/ in sentences and in the phonetically balanced text were annotated.

To annotate V vowel boundaries, a combination of waveform and spectrogram analysis was used in *Praat's 6.0.47 Sound Editor*. The wideband spectrogram with default settings (view range of 0 to 5,000 Hz) was utilized to examine the periodicity of the acoustic signal, second formant (F_2) amplitude, and the f_0 track. Spectrograms with a wider view range (0 to 16,000 Hz) were used to annotate fricatives produced in NS mode. Additionally, constant auditory monitoring was conducted over headphones for all recordings [4].

The process of segmenting WS differs from that of NS [73], and it requires manual and laborious procedures [74, 51]. Segmentation involves visual analysis of waveforms, formant structures in spectrograms (such as F_2 and F_3 onset and offset), and changes in intensity [75]. *Praat*'s default spectrogram settings were adjusted only for the view range, which was set to 0 to 16,000 Hz for both vowels and fricatives. The primary acoustic cues used to annotate WS were the waveforms and spectrograms of frication noise. Phones that were produced with a hard or abrupt glottal attack were not annotated [4].

4.2.3.3 Reliability verification

The speech productions of two participants were randomly selected, annotated and transcribed by a trained phonetician who was not involved in the study and was unaware of its objectives. The point-to-point reliability was determined to be 92.34% [4], which was deemed satisfactory for the purpose of this investigation. The two participants constituted 12% of the speech samples, which is in line with the reported percentage of reliability checks in other studies of WS [76, 73].

4.3 Dataset preprocessing

Dataset preprocessing was an essential step in preparing the phonetically annotated WS/NS speech dataset for further analysis and modeling. It encompassed downsampling of audio files (4.3.1), phone annotation-based segmentation (4.3.2), dataset selection and cleaning (4.3.3), CTV segments labelling (4.3.4), and audio segments normalization (4.3.5). After performing these preprocessing steps, the dataset was ready for feature engineering.

4.3.1 Downsampling of audio files

The audio files in the speech dataset, obtained as explained in the previous Section (4.2), underwent resampling from a Sampling Frequency (SF) of 48000 Hz to 22050 Hz. This resampling enhances the efficiency of the subsequent analysis while preserving a sufficiently detailed spectral content (Nyquist Frequency $= 11,025$ Hz).

4.3.2 Phone annotation-based segmentation

The downsampled speech dataset obtained as explained in last Subsection 4.3.1 underwent segmentation into phones utilizing the phonetic annotations. Subsequently, these segments were systematically arranged into a tabular structure, wherein each entry corresponds to a distinct phone segment. The table encompasses the following 8 attributes for each entry:

- **Sex:** Indicates the sex of the speaker;
- **Speaker Identification (ID):** Identifies the speaker;
- **Task:** Describes the associated task;
- **Speech Mode (SM):** Specifies whether the segment is normal or whispered speech;
- **Sequence Index:** Represents the position of the phone within the task;
- **Segment's Waveform Audio File Format File (SWAV):** Contains a Waveform Audio File Format (WAV) file with the samples of the audio segment;
- **Sampling Frequency (SF):** Indicates the frequency at which the audio segment is sampled;
- **Phonetic Annotation Label (PAL):** Provides the phonetic annotation label for the segment.

By employing phone annotation-based segmentation, the dataset was effectively partitioned into distinct phone segments, facilitating further analysis and processing.

4.3.3 Dataset selection and cleaning

The dataset obtained as described in last Subsection 4.3.2 was selected and cleaned to ensure reliable PALs. For that purpose, the following steps were executed:

1. The table entries were selected based on the following criteria:

$$(\text{PAL} = \text{"silence"}) \vee (\text{length}(\text{PAL}) = 3 \wedge \text{PAL}[-1] \in \{\text{"w"}, \text{"W"}\}) \tag{4.1}$$

$$\text{SM} = \text{"02"} \tag{4.2}$$

   The first condition (4.1) states that the entry's PAL attribute must be either *"silence"* or a three-character string ending in either *"w"* or *"W"*. The second condition (4.2) mandates that the entry's SM attribute must have a value of "02", indicating a WS utterance. The resulting table will solely contain entries that fulfill all of the aforementioned conditions, avoiding most erroneous PALs;

2. Following the selection process, some errors were still detected in the PALs. In response, corrections were applied to each entry in the table: "-" was replaced with "_" and "w" was replaced with "W".

4.3.4 Candidate to voicing segments labelling

A new boolean attribute CTV was added to the dataset obtained as described in last Subsection 4.3.3, enabling the classification of the phone segments between CTV and NCTV.

- Segments with the following Speech Assessment Methods Phonetic Alphabet (SAMPA) phonetic annotation labels were identified as CTV: 1_W, 4_W, 6_W, A_W, E_W, L_W, N_W, O_W, R_W, Z_W, a_W, b_W, d_W, e_W, g_W, i_W, l_W, m_W, n_W, o_W, u_W, v_W, z_W. For each of these segments, the corresponding CTV attribute value was set to 1;
- Conversely, segments with the following SAMPA phonetic annotation labels were identified as NCTV: S_W, f_W, k_W, p_W, s_W, t_W, silence. For each of these segments, the corresponding CTV attribute value was set to 0.

Table 4.2 presents the phonetic annotation labels, their corresponding SAMPA and IPA alphabet symbols, and whether they are considered CTV or NCTV.

4.3.5 Audio segments normalization

The SWAV column in the dataset obtained as described in Subsection 4.3.4 was subjected to normalization, employing a standardization technique.

Each entry's SWAV value was standardized as follows:

Table 4.2: SAMPA phonetic annotation labels of segments, correspondent IPA symbols and candidate to voicing label.

Phonetic annotation label	**SAMPA Symbol**	**IPA Symbol**	**Candidate to voicing**
1_W	1	ɨ	1
4_W	4	ɾ	1
6_W	6	ɒ	1
A_W	A	ɑ	1
E_W	E	ɛ	1
L_W	L	ʎ	1
N_W	N	ŋ	1
O_W	O	ɔ	1
R_W	R	ʁ	1
Z_W	Z	ʒ	1
a_W	a	a	1
b_W	b	b	1
d_W	d	d	1
e_W	e	e	1
g_W	g	ɡ	1
i_W	i	i	1
l_W	l	l	1
m_W	m	m	1
n_W	n	n	1
o_W	o	o	1
u_W	u	u	1
v_W	v	v	1
z_W	z	z	1
S_W	S	ʃ	0
f_W	f	f	0
k_W	k	k	0
p_W	p	p	0
s_W	s	s	0
t_W	t	t	0
silence	sil	‖	0

1. The mean value of all original SWAVs was calculated and then subtracted from each SWAV;
2. The resulting values were divided by the standard deviation of all original SWAVs.

This process can be mathematically represented as:

$$SWAV_{Normalized} = \frac{SWAV_{Original} - \mu}{\sigma} \tag{4.3}$$

In this Equation:

- $SWAV_{Normalized}$: standardized SWAV;
- $SWAV_{Original}$: original, pre-standardization SWAV;
- μ: mean of all original SWAVs;
- σ: standard deviation of all original SWAVs.

Through the application of this normalization technique, the bias originating from the original scales of the SWAVs was eliminated. This facilitated meaningful comparisons and subsequent analyses on the standardized SWAVs.

4.4 Feature engineering

Feature engineering is a crucial step in the analysis of the phonetically annotated WS/NS speech dataset. It involves transforming the speech data into a set of meaningful and representative features that can be used as inputs for the subsequent modeling procedures. It encompassed feature extraction (4.4.1), feature normalization (4.4.2), dataset explosion from segments to frames (4.4.3), class distribution balancing through selective silence frame reduction (4.4.4), context size definition (4.4.5), context-sized sequences dataset generation (4.4.6), BFS definition (4.4.7), and SFS definition (4.4.8).

4.4.1 Feature extraction

The *Librosa Python*'s package was used to perform feature extraction on the preprocessed dataset obtained as described in Section 4.3 [52, 42].

For each table entry, the phone's normalized WAV files and SFs were utilized to compute several spectral features. All feature extraction computations were performed with a Frame Size (FS) of 1,024 samples and a Hop Size (HS) of 512 samples. Unless explicitly specified, the default values were utilized for all remaining parameters.

The following features were extracted:

1. **ZCR (1 feature):** Rate at which the signal changes sign. Results in a scalar value for each frame of audio, leading to 1 feature per frame;
2. **RMS (1 feature):** RMS Energy of each frame. Results in a scalar value, leading to 1 feature per frame;
3. **STFT (512 features):** STFT is applied to each frame of the audio, using an Hann window. The result is a spectrum of 512 frequency bins;

4. **Mel Spectrogram** (128 **features):** The Mel-scaled power spectrogram of the audio is computed, using an Hann window, FFT. For each frame, it produces a 128-bin Mel-frequency spectrum;

5. **STFT Chromagram** (12 **features):** Computes chroma features using STFT. Representations of the audio based on the 12 different pitch classes are obtained. Hence, it produces a 12-dimensional feature vector for each frame;

6. **CQT Chromagram** (12 **features):** Computes chroma features using CQT. Representations of the audio based on the 12 different pitch classes are obtained. Hence, it produces a 12-dimensional feature vector for each frame;

7. **CENS** (12 **features):** Computes chroma features using CENS. Representations of the audio based on the 12 different pitch classes are obtained. Hence, it produces a 12-dimensional feature vector for each frame;

8. **Spectral Centroid** (1 **feature):** Characterizes the center of energy distribution of the spectrum across the frequency range. Results in single value for each audio frame;

9. **Spectral Bandwidth** (1 **feature):** Measures the width of the spectrum around its centroid. Results in single value for each audio frame;

10. **Spectral Contrast** (7 **features):** Measures the difference in amplitude between peaks and valleys in a spectrum. For each audio frame, 7 features are obtained (one for each octave);

11. **Spectral Flatness** (1 **feature):** Indicates the balance between the energy in the harmonic and non-harmonic components of the spectrum. Results in a single value for each audio frame;

12. **Spectral Rolloff** (1 **feature):** Measures the frequency below which a specified percentage of the total spectral energy lies. It is computed using the STFT with a rolloff percentage of 85% (default), resulting in a single value for each audio frame;

13. **Tonnetz** (6 **features):** This method computes the Tonnetz features using chroma computed from the MFCCs. It produces a 6D feature vector for each frame;

14. **MFCCs** (49 **features):** Computes 49 MFCCs. It provides a 49-dimensional feature vector for each frame;

15. **MFCCs Delta** (49 **features):** Computes the first-order difference (Delta) of the MFCCs, using a width of 3 and interpolation mode set to *"nearest"*. Provides a 49-dimensional feature vector for each frame;

16. **MFCCs Delta Delta** (49 **features):** Computes the second-order difference (Delta Delta) of the MFCCs, using a width of 3 and interpolation mode set to *"nearest"*. Provides a 49-dimensional feature vector for each frame;

17. **Polyfeatures (2 features):** Computes coefficients of fitting an order-degree polynomial to the columns of a spectrogram, using the STFT with default parameters. It produces a 2D feature vector per frame.

For more detailed information regarding feature extraction, please refer to the Subsection *"Feature extraction"* of the Section *"Background"* (2.5.1).

4.4.2 Feature normalization

In Section 4.4.1, the extraction of features from the dataset was described. The following step involved normalizing these features to enable unbiased comparisons and more efficient analysis, using a standardization technique.

Each feature column in the dataset consists of various feature values. These were standardized as follows:

1. The mean value of all original feature values in a column was calculated, and subtracted from each individual feature value in that column;

2. The resulting values were divided by the standard deviation of all the original feature values from the same column.

This process can be depicted mathematically as:

$$Feature_{Normalized} = \frac{Feature_{Original} - \mu}{\sigma} \tag{4.4}$$

In this Equation:

- $Feature_{Normalized}$: resultant standardized feature value;
- $Feature_{Original}$: pre-standardization, original feature value;
- μ: mean of all original feature values in the column;
- σ: standard deviation of all original feature values in the column.

Standardizing the features was a crucial part of the data analysis. It eliminated any bias due to the differing scales of the original features. This allowed for an effective comparison of features and enhanced subsequent data analysis.

4.4.3 Dataset explosion from segments to frames

Following the feature extraction and normalization processes described in Subsections 4.4.1 and 4.4.2, a dataframe was obtained where each entry corresponded to a phone, with mapping to an array of feature values, ascertained for each frame with a FS of $1,024$ samples, centered at every HS of 512 samples.

The dataframe was reformatted into a more analytically conducive structure, where each entry corresponds to the feature values extracted from a single frame. This transformation was accomplished through a technique known as explosion, which was applied to the dataframe based on the respective feature arrays. Each resulting entry retains the corresponding attribute values from the original entry, ensuring the preservation of their relationship.

This operation greatly enhances the interpretability of the data, making it more amenable for subsequent ML algorithms and analyses.

4.4.4 Class distribution balancing through selective silence frame reduction

After the dataset explosion process described in Subsection 4.4.3, the CTV/NCTV class distribution of the dataset was analyzed, with the goal of detecting a possible class imbalance. Such an imbalance could potentially interfere with the classifier's learning process, affecting its effectiveness.

A class imbalance was detected: the NCTV class had an overrepresentation, primarily due to the abundance of silence frames. In response, a method for selectively reducing silence frames followed. The method consisted of several steps:

1. Identify continuous silence frames in the dataset;
2. Remove a percentage of silence frames from the middle of these segments.

To determine the proportion of silence frames to remove that optimizes the dataset's class balance, the second step should be performed iteratively.

This selective silence frame reduction strategy served dual purposes:

- It facilitated a balanced class distribution in the dataset;
- Preserved silence frames located near the segment boundaries, considering their potential role in holding critical contextual information.

4.4.5 Context size definition

The binary classification model was conceived to leverage past data to perform present classifications, by accepting sequences of feature vectors as input. Each vector corresponds to a frame of audio data, containing the values of the features extracted using a FS of $1,024$ samples and an HS of 512 samples, resulting in an overlap of $\text{HS}/\text{FS} = 50\%$.

The size of the input sequence of feature vectors, or the Overlapping Context Size (OCS), was defined to a value that ensures that the model captures the entirety of an average-sized word from the dataset. For that purpose, the following quantities were determined:

1. **Average Word Length in Phones (AWLP):** Obtained empirically from the dataset;
2. **Average Phone Length in Samples (APLS):** Also obtained empirically from the dataset;

3. **Average Word Length in Frames (AWLF):** Determined as depicted in Equation (4.5):

$$AWLF = \frac{AWLP \times APLS}{FS} \tag{4.5}$$

4. **Overlapping Context Size (OCS):** The objective consisted in utilizing information from past contiguous frames that correspond to an average word with length AWLF on each classification. For this purpose, the Contiguous Context Size (CCS) should be equal to the AWLF. The model's input consists of feature vectors extracted with overlap. Hence, it is essential to calculate the OCS that is equivalent to the CCS. Equation (4.6) establishes a relation between OCS considering an overlap of HS/FS and the equivalent CCS. Therefore, an input sequence with a number of feature vectors equal to OCS correspond exactly to the feature values extracted from a number of contiguous frames equal to CCS.

$$\begin{aligned} OCS &= CCS \times (1 + \frac{HS}{FS}) \times 2 \\ OCS &= AWLF \times (1 + \frac{HS}{FS}) \times 2 \end{aligned} \tag{4.6}$$

The obtained OCS matches the AWLF calculated in Equation (4.5), ensuring that the model's context captures the entirety of an average word from the dataset.

4.4.6 Context-sized sequences dataset generation

In the exploded dataset obtained as described in Subsection 4.4.3, each data point corresponds to a frame, being characterized by a multidimensional feature vector and the corresponding classification label. The process of generating context-sized sequences from this data represents a critical operation, providing the subsequent ML steps with the temporal context of speech. The following topics explain the process:

1. **Stride:** During the sequence generation process, a stride or step size of 1 was used. The stride refers to the distance moved through the data to form each sequence. With a stride of 1, each successive sequence starts one frame later than the previous sequence, resulting in substantial overlap between the sequences;

2. **Sequence size**: The sequence size was set to the OCS, defined in the previous Subsection 4.4.5;

3. **Sequence labels assignment:** The classification label corresponding to the last frame of each sequence was assigned as the label for that particular sequence. The assigned label was either CTV or NCTV, depending on the label of the last feature vector.

By following these steps, a 3 dimensional array of OCS-sized sequences of multidimensional feature vectors (sequences by frames by features) was obtained.

4.4.7 Baseline feature subset definition

A total of 49 features were chosen to form the BFS, based on the demonstrated empirical success in recent literature.

4.4.8 Selected features subset definition

A comprehensive feature selection process was conducted, encompassing feature dimension analysis (4.4.8.1), feature extraction time analysis (4.4.8.2), and feature selection based on PCC, SCC, ANOVA F-value and RFI (4.4.8.3).

This approach resulted in the selection of 49 features that formed the SFS.

4.4.8.1 Feature dimension analysis

High-dimensional features, such as STFT and Mel Spectrogram, have been used traditionally as high-dimensional representations of audio signals. While these representations are comprehensive, they often contain excessive complexity with redundant or highly correlated data. Consequently, classifiers may suffer from impaired effectiveness.

Fortunately, there are features that offer lower-dimensional representations while still capturing the essential information obtained from high-dimensional features, such as MFCCs. By utilizing these features, redundancy is reduced, and the risk of overwhelming classifiers with highly correlated data is mitigated. This simplification of the feature space enables more efficient analysis and interpretation of audio data.

To optimize the representation of the data, a dimension analysis was conducted on the extracted features. The results guided the feature selection process, allowing for the retention of crucial information while eliminating unnecessary complexity.

4.4.8.2 Feature extraction time analysis

Ensuring compliance with the MAPT constraint is crucial for enabling the online operation of the system. For that purpose, the following expressions were defined:

1. **MAPT:** The MAPT was conservatively defined to be the duration of a single HS at a SF of 22050 Hz, resulting in approximately 23220 μs, as expressed in Equation (4.7):

$$\begin{aligned} MAPT &= HS \times \left(\frac{1}{SF}\right) \\ MAPT &= 512 \times \left(\frac{1}{22{,}050\ Hz}\right) \\ MAPT &= 23{,}220\ \mu s \end{aligned} \tag{4.7}$$

2. **Total Feature Extraction Time (TFET):** The single hop TFET for any subset of features can be obtained by summing the desired single hop Individual Feature Extraction Time (IFET)s, as stated in Equation (4.8).

$$TFET = \sum IFET \tag{4.8}$$

3. **Total Estimated Processing Time (TEPT):** The TEPT was defined as the combined time required for a single hop TFET and a single hop Classifier's Inference Time (CIT).

$$TEPT = TFET + CIT \tag{4.9}$$

4. **MAPT compliance condition:** To ensure compliance, the TEPT cannot surpass the MAPT. This is expressed by the Equation (4.9):

$$\begin{aligned} TEPT &< MAPT \\ TFET + CIT &< MAPT \\ \sum IFET + CIT &< HS \times \left(\frac{1}{SF}\right) \\ \sum IFET + CIT &< 23,220\ \mu s \end{aligned} \tag{4.10}$$

With the necessary expressions defined, the following steps were taken to conduct the feature extraction time analysis and the consequent feature selection:

1. **Average IFET estimation:** The average IFET for each feature was estimated from the first 1,000 phone segments of the database. This estimation adopted a conservative approach, as some feature extraction methods share common steps that have been factored into each IFET;

2. **TFET estimation:** The single hop TFET was obtained for all features by summing the IFETs estimated in the previous step, as depicted in Equation (4.8);

3. **Exclusion of features with the highest IFET:** The features with the highest IFETs were excluded, ensuring that:

   - The MAPT compliance condition was not compromised by the TFET alone;
   - There is a temporal slack for the CIT.

4.4.8.3 Feature selection based on Pearson correlation, Spearman correlation, Analysis of Variance F-Value and Random Forest Importance

This feature selection analysis operated on the 3 dimensional array of context-sized sequences of multidimensional feature vectors (sequences by frames by features) obtained as described in Subsection 4.4.6. The procedure followed these steps:

1. **Data reshaping:** The 3 dimensional array of sequences was transformed into a 2 dimensional array (frames by features). This reshaping ensured that each row corresponded to a frame and each column aligned with a specific feature across all frames;

2. **Classification label replication:** The classification labels, originally associated with each sequence, were repeated for each frame within that sequence. This repetition matched the reshaped structure of the feature vectors and maintained the link between the feature frames and their associated classification labels;

3. **Feature importance metrics:** With the reshaped array and replicated classification labels, various metrics were computed for each feature across all frames, in order to quantify the relationship between each feature and the associated classification labels:

   (a) **PCC:** The PCC quantifies the linear relationship between each feature and the target variable across the entire reshaped data;

   (b) **SCC:** The SCC measures the rank correlation between each feature and the target variable across the entire reshaped data;

   (c) **ANOVA F-value:** For each feature, an ANOVA F-value is calculated as the ratio of the variance of the means of the two classes (between-group variance) to the mean of the variances within each class (within-group variance). A higher F-value score indicates that the feature is more discriminative for the binary classification task;

   (d) **RFI:** In order to obtain an estimation of the importance of each feature, a Random Forest Classifier was trained on the reshaped data. This model generates an importance score for each feature, which is indicative of its contribution to the decision-making process within the model.

4. **Metrics normalization:** To assess the overall importance of the features, a normalization process was applied to the metrics obtained from the four methods, resulting in values ranging from 0 to 100. This normalization enabled easier comparison and interpretation of the scores;

5. **Metrics averaging:** Next, the normalized scores were averaged, with equal weight assigned to each metric, as shown in Equation (4.11):

$$Average\ Score = 0.25 \times PCC + 0.25 \times SCC + 0.25 \times FTest + 0.25 \times RFI \tag{4.11}$$

   The resulting average score provided a comprehensive measure of the feature's importance across all the metrics;

6. **Selected feature subset:** Based on the average scores, a set of 49 features was selected as the top performers, originating the SFS.

For a comprehensive understanding of the feature selection metrics used, please consult the Subsection *"Feature selection"* in the Chapter *"Background"* (2.5.2).

4.5 Selection and design of DL-based model architectures

The strategy for choosing and designing the DL model architectures was guided by the following principles:

1. **Empirical success:** The chosen architecture should have a proven track record of success in classification tasks involving sequential and temporal data. This requirement ensures the selection of architectures with demonstrated efficiency and robustness;

2. **Trainable parameters balance:** The architecture should feature around $200,000$ trainable parameters. This parameter count has been found optimal in preliminary tests, providing a good balance between training time and model performance. This count not only allows for the flexibility required for this study but also establishes a baseline complexity level, making all architectures roughly comparable in terms of complexity.

The models were uniformly compiled, using the binary cross-entropy loss function and the Adaptive Moment Estimation (Adam) learning rate optimization algorithm, with Accuracy serving as the primary performance metric.

4.6 Evaluation metrics definition

To evaluate the different model/feature subset pairs, the evaluation metrics were defined, namely performance (4.6.1) and computational (4.6.2) metrics.

4.6.1 Performance metrics

The performance metrics used to assess the models are outlined below. These metrics provide insight into the effectiveness of the models in various aspects:

- **Accuracy:** A measure of the overall correct predictions made by the model;
- **Precision:** The proportion of true positive predictions out of all positive predictions;
- **Recall:** The proportion of actual positive instances that were correctly identified;
- **Specificity:** The proportion of actual negative instances that were correctly identified;
- **F1 Score:** The harmonic mean of Precision and Recall, providing a balance between these two metrics;
- **AUC-ROC:** Aggregate measure of model performance across all possible classification thresholds.

For a more detailed explanation of these metrics, please refer to the Subsection *"Performance metrics"* in Chapter *"Background"* (2.4.4).

4.6.2 Computational metrics

The computational metrics employed are briefly described below. These metrics shed light on the practical aspects of model training and inference, including computational effort, temporal efficiency, and model complexity.

- **Number of epochs:** The number of times the learning algorithm has worked through the entire training dataset;
- **Training time:** The total amount of time that the model spends in the training phase;
- **Average training time per epoch:** The average amount of time taken to complete each epoch during training;
- **Best epoch:** The epoch at which the model achieved the best performance on the validation set during training;
- **Number of trainable parameters:** The quantity of parameters in the model that can learn and change as the model trains;
- **Inference time:** The time taken by the model to make predictions after it has been trained.

For a more detailed explanation of these metrics, please refer to the Subsection *"Computational metrics"* in Chapter *"Background"* (2.4.5).

4.7 Assessment and comparison of all model/features subset pairs

To identify the best performing model/features subset pair on the phonetically annotated WS/NS speech dataset, an assessment and comparison of all the model/features subset pairs were conducted. It encompassed TTS evaluation (4.7.1), performance comparison across features subsets (4.7.2), and selection of the best performing model/features subset pair (4.7.3).

4.7.1 Train-Test Split evaluation

For every model and for both features subsets, the TTS process was executed 5 times using the GPU, each following these steps:

1. First, the dataset was filtered to contain only the features corresponding to the subset under analysis. This process ensured that each features subset (BFS or SFS) was evaluated individually;
2. Each features subset corresponded to sequences of OCS feature vectors using a stride of 1, with the CTV/NCTV label corresponding to the last feature vector assigned to each sequence;

3. Following this, the sequences of feature vectors and their corresponding labels were randomly shuffled to ensure model robustness and prevent overfitting. The random state for this shuffle operation was set as 41 plus the current iteration number, introducing an element of controlled randomness at each iteration;

4. Next, the sequences of feature vectors and their associated labels were distributed into three segments — a training set, a validation set, and a testing set. This division was made in a 70%, 15%, and 15% ratio respectively. The purpose of this distribution is to allow the model to learn from the training data, fine-tune parameters with validation data, and then evaluate performance using the test data;

5. Each model was trained on the training set and validated on the validation set. To avoid overfitting, an early stop callback was utilized, which halted the training process if the validation loss did not improve for 6 consecutive epochs. When the training process was halted early, the model weights corresponding to the lowest validation loss were restored. Metrics collected during training included: Number of Epochs, Training Time, Average Training Time per Epoch, and Best Epoch;

6. Finally, the model was evaluated on the test set, to assess the models' performance and generalizability. The following metrics were obtained during evaluation: Accuracy, Precision, Recall, Specificity, F1 Score, AUC-ROC, and Inference Time.

By employing this approach, every model was independently trained, validated and evaluated using the two features subsets, enabling a comprehensive comparison of the performances and generalizability of each model/features subset pair.

4.7.2 Performance comparison across features subsets

In order to assess the impact of the SFS on the overall performance of the models, an analysis was conducted to measure the performance metric gains achieved by employing the SFS in comparison to the BFS.

4.7.3 Selection of the best performing model/features subset pair

The best performing model/features subset pair was chosen through the results of TTS evaluation, namely performance and computational metrics.

4.8 Assessment of the best performing model/features subset pair

The best performing model/features subset pair selected in 4.7.3 was subject to a more detailed assessment, in order to validate its effectiveness and efficiency. This process encompassed performance assessment across articulation manner classes (4.8.1), K-FCV evaluation (4.8.2), exemplification of VD segmentation (4.8.3), and verification of the compliance with the MAPT (4.8.4).

4.8.1 Performance assessment across articulation manner classes

The Accuracy of the best performing model/feature subset pair was analysed across different articulation manner classes. For that purpose, the following process was followed:

1. The speech frames were categorized based on articulation manner and voicing attributes. To get more detailed information on voicing and articulation manners, refer to Subsection *"European Portuguese Phonetics"* of the Section *"Background"* (2.1.2);

2. The 5 iterations of TTS evaluation were repeated for the best performing model/features subset pair, with the same random states as mentioned in Subsection 4.7.1. VDs were obtained;

3. The VDs success is evaluated by comparing the obtained values to the ground truth;

4. The Accuracy of this prediction is calculated individually for each articulation manner class, providing a detailed overview of model performance across these classes.

The results of this evaluation provide a clear understanding of the Accuracy with which the system can make VDs across different articulation manner and voicing classes. This critical information can help in refining the model, ensuring its robust performance. Consequently, targeted improvements can be made by focusing on classes where Accuracy might be lower.

4.8.2 K-Fold Cross Validation evaluation

To rigorously substantiate the best performing model/features subset pair's performance and generalizability, K-FCV was executed using the GPU, as follows:

1. Initially, the dataset was filtered to include only the features corresponding to the best performing subset;

2. Each features subset corresponded to sequences of OCS feature vectors using a stride of 1, with the CTV/NCTV label corresponding to the last feature vector assigned to each sequence;

3. The sequences of feature vectors and their corresponding labels were randomly shuffled with a fixed random state of 42, ensuring a controlled level of randomness;

4. K-FCV with K set to 5 was implemented with stratified sampling, ensuring that each fold has the same proportion of CTV and NCTV labels as the entire dataset. In each iteration of the cross-validation, the sequence data was split into a training set, consisting of four out of five folds (80% of the data), and a validation set, consisting of the remaining fold (20% of the data);

5. In each iteration, a new instance of the model was created and compiled. The model was then trained on the training set and validated on the validation set. An early stop callback

was utilized, which halted the training process if the validation loss did not improve for 6 consecutive epochs. The model weights corresponding to the lowest validation loss were restored upon early stopping;

6. For each fold, after training on the training set and validation on the validation set, the model was evaluated on the same validation set, providing an unbiased measure of model effectiveness and generalizability for that fold. Metrics collected during evaluation included: Accuracy, Precision, Recall, Specificity, F1 Score, AUC-ROC, and Inference Time;

7. The entire process was repeated for each of the five folds, changing the composition of the training and validation sets in each iteration so that all data is used for both training and validation at some point. The metrics from each fold were recorded.

The implementation of K-FCV offers a detailed and unbiased evaluation of the model's ability to perform well on new, unseen data. By leveraging all available data for both training and validation, this approach helps to uncover the model's true performance potential. It further strengthens the confidence in the model's predictive capacity and its relevance to real-world scenarios, enhancing the overall reliability of the findings.

4.8.3 Exemplification of voicing decision segmentation

The best performing model/features subset pair selected in 4.7.3 was tested on several tasks from the phonetically annotated WS/NS dataset to exemplify and visualize its operation. The tasks were carefully selected to encompass phones from all articulation manner classes.

4.8.4 Compliance with the Maximum Allowable Processing Time

To validate the system's capability of online operation, the MAPT compliance condition defined in 4.4.8.2 was evaluated. The following steps were taken:

1. The best performing model/features subset pair's TFET and CIT were estimated. The CIT was obtained from the results of TTS evaluation;

2. The MAPT compliance condition was evaluated: the TEPT cannot surpass the MAPT, as expressed in Equation (4.12):

$$
\begin{aligned}
TEPT &< MAPT \\
TFET + CIT &< MAPT \\
\sum IFET + CIT &< HS \times \left(\frac{1}{SF}\right) \\
\sum IFET + CIT &< 512 \times \left(\frac{1}{22{,}050\ Hz}\right) \\
\sum IFET + CIT &< 23{,}220\ \mu s
\end{aligned}
\tag{4.12}
$$

4.9 Chapter summary

In the Chapter *"Methodology"*, various aspects of the research process were discussed in detail. These aspects include the description of the hardware and software utilized (4.1), the acquisition of the phonetically annotated WS/NS dataset (4.2), the preprocessing steps applied to the dataset (4.3), the process of feature engineering (4.4), the selection and design of DL-based model architectures (4.5), the definition of evaluation metrics (4.6), the methods employed to assess and compare pairs of models/features subsets (4.7), and the evaluation of the best performing model/features subset pair (4.8).

In the upcoming Chapter *"Results and discussion"*, a comprehensive overview of the research findings in each of the aforementioned aspects will be presented. This includes the outcomes of the phonetically annotated WS/NS dataset acquisition (5.1), dataset preprocessing (5.2), feature engineering (5.3), selection and design of DL-based model architectures (5.4), assessment and comparison of all model/features subset pairs (5.5), and evaluation of the best performing model/features subset pair (5.6).

Chapter 5

Results and discussion

In this Chapter, a comprehensive overview of the research findings is presented. It covers the results of phonetically annotated WS/NS dataset acquisition (5.1), dataset preprocessing (5.2), feature engineering (5.3), selection and design of DL-based model architectures (5.4), assessment and comparison of all model/features subset pairs (5.5), and assessment of the best performing model/features subset pair (5.6).

5.1 Phonetically annotated whispered/normal speech dataset acquisition

The application of the method described in the Section *"Phonetically annotated WS/NS dataset acquisition"* of the Chapter *"Methodology"* (4.2) originated a phonetically annotated WS/NS speech dataset.

The dataset comprises 54 audio files per each of the 17 participants:

- 27 files are dedicated to NS;
- 27 files are dedicated to WS.

5.2 Dataset preprocessing

The methods described in the Section *"Dataset preprocessing"* of the Chapter *"Methodology"* (4.3) transformed the original phonetically annotated WS/NS dataset into a tabular WS dataset, with the following characteristics:

- The table has $12,718$ entries;
- Each table entry corresponds to a phone segment, with the following attributes:
  - **Sex:** Indicates the sex of the speaker;
  - **Speaker ID:** Identifies the speaker;

- **Task:** Describes the associated task;
- **Speech Mode (SM):** Specifies whether the segment is NS or WS;
- **Sequence Index:** Represents the position of the phone within the task;
- **Segment's Waveform Audio File Format File (SWAV):** Contains a WAV file with the samples of the audio segment;
- **Normalized SWAV:** Contains a normalized WAV file with the samples of the audio segment;
- **Sampling Frequency (SF):** Indicates the SF of the audio segment, which is now $22,050\ Hz$ for all entries;
- **Phonetic Annotation Label (PAL):** Provides the phonetic annotation label for the segment;
- **CTV/NCTV Label:** Binary target label that indicates which phone segments are CTV.

5.3 Feature engineering

In this Section, the findings from the application of the procedures described in Section *"Feature engineering"* of the Chapter *"Methodology"* (4.4) are presented. It encompasses the results of feature extraction (5.3.1), feature normalization (5.3.2), dataset explosion from segments to frames (5.3.3), class distribution balancing through selective silence frames reduction (5.3.4), context size definition (5.3.5), context-sized sequence dataset generation (5.3.6), BFS definition (5.3.7), and SFS definition (5.3.8).

5.3.1 Feature extraction

The method described in Subsection *"Feature extraction"* of the Chapter *"Methodology"* (4.4.1) resulted in the tabular WS dataset described in the previous Section *"Data processing"* (5.2), with additional attributes for each extracted feature. Each additional attribute contains an array of feature values extracted from the phone segment.

5.3.2 Feature normalization

The method described in Subsection *"Feature normalization"* of the Chapter *"Methodology"* (4.4.2) resulted in the tabular WS dataset with extracted feature described in the previous Subsection *"Feature extraction"* (5.3.1) with the features normalized.

5.3.3 Dataset explosion from segments to frames

The method described in Subsection *"Dataset explosion from segments to frames"* of the Chapter *"Methodology"* (4.4.3) resulted in an exploded version of the tabular WS dataset with normalized extracted feature obtained in the previous Subsection "Feature normalization" (5.3.2).

The resulting exploded dataset has the following characteristics:

- 2,417,774 entries;
- Each table entry corresponds to an audio frame obtained using a FS of 1,024 samples and a HS of 512 samples;
- Each exploded entry retains the corresponding attribute values from the original entry, ensuring the preservation of their relationship.

5.3.4 Class distribution balancing through selective silence frame reduction

A class distribution imbalance was detected in the the exploded dataset obtained in last Subsection *"Dataset explosion from segments to frames"* 5.3.3. In the exploded dataset with 241,777 entries:

- The NCTV group had an overrepresentation with 163,262 instances, primarily due to the abundance of silence frames;
- The CTV group had 78,515 instances.

In response, the method described in Subsection *"Class distribution balancing through selective silence frame reduction"* of the Chapter *"Methodology"* (4.4.4) was applied. The precise proportion of silence frames to remove in order to attain an optimal class balance in the dataset was determined iteratively, resulting in a percentage of about 58%.

This action resulted in a balanced dataset, with a total of 157,028 entries:

- The CTV category had 78,515 instances;
- The NCTV category had 78,513 instances.

5.3.5 Context size definition

In this Subsection, the results of the application of the method described in the Subsection *"Context size definition"* of the Chapter *"Methodology"* (4.4.5) are disclosed:

1. The AWLP was calculated empirically from the dataset, resulting in a value of 3.65 phones;
2. The APLS was calculated empirically from the dataset, resulting in a value of 4,002.88 samples;
3. The AWLF was obtained as depicted in Equation (5.1):

$$AWLF = \frac{AWLP \times APLS}{FS} \approx \frac{3.65 \times 4,002.88}{1,024} \approx 16\,\text{frames} \tag{5.1}$$

For computational purposes, the AWLF value was approximated to a base-2 number.

4. The OCS equivalent to an average word with length AWLF, was determined as depicted in Equation (5.2), ensuring that the context captures the entirety of an average word.

$$
\begin{aligned}
OCS &= CCS \times (1 + \frac{HS}{FS}) \times 2 \\
OCS &= AWLF \times (1 + \frac{HS}{FS}) \times 2 \\
OCS &= 16 \times (1 + \frac{512}{1,024}) \times 2 \\
OCS &= 31\, frames
\end{aligned}
\tag{5.2}
$$

5.3.6 Context-sized sequences dataset generation

The method described in Subsection *"Context-sized sequence generation"* of the Chapter *"Methodology"* (4.4.6) was applied to the dataset obtained in Subsection *"Class distribution balancing through selective silence frame reduction"* of the Chapter *"Results"* (5.3.4).

The result was a context-sized sequences WS dataset with the following characteristics:

- The dataset is composed of 156,998 OCS-sized sequences of multidimensional feature vectors;
- Each sequence has a target label assigned. The label can be either CTV or NCTV, according to the original label of its last feature vector.

5.3.7 Baseline features subset definition

The method outlined in Subsection *"Baseline feature subset definition"* of Chapter *"Methodology"* (4.4.7) was performed on the extracted features. The 49 MFCCs were selected as the BFS due to their widespread usage in state-of-the-art VD approaches, as confirmed in Chapter *"Voicing decision approaches — a review"* (3) [29, 30, 31, 32].

5.3.8 Selected features subset definition

The analytical processes described in Subsection *"Selected features subset definition"* of the Chapter *"Methodology"* (4.4.8) were conducted on the OCS-sized sequences dataset obtained in 5.3.6.

The feature selection results obtained from feature dimension analysis (5.3.8.1), feature extraction time analysis (5.3.8.2), and feature selection based on PCC, SCC, ANOVA F-value, and RFI (5.3.8.3), contributed to the definition of the SFS.

5.3.8.1 Feature dimension analysis

The analysis described in the Subsubsection *"Feature dimension analysis"* of the Chapter *"Methodology"* (4.4.8.1) was conducted on the extracted features.

It was observed that the STFT and the Mel spectrogram, with their respective dimensions of 512 and 128, were excessively complex representations of the data. Furthermore, it was recognized that certain features offer lower-dimensional representations while still capturing the essential information obtained from those high-dimensional features. Consequently, the STFT and the Mel spectrogram were excluded from the SFS.

5.3.8.2 Feature extraction time analysis

The analysis described in the Subsubsection *"Feature extraction time analysis"* of the Chapter *"Methodology"* (4.4.8.2) was conducted on the feature subset obtained in 5.3.8.1. The following results were obtained:

1. **Average IFET estimation:** Table 5.1 presents the average single hop IFET for each feature estimated from the first 1,000 phone segments of the database;

Table 5.1: Single hop Individual Feature Extraction Times.

Feature group	**IFET** (μs)
Zero Crossing Rate	79.24 ± 0.08
Root Mean Square	112.01 ± 0.11
Short-time Fourier Transform	182.14 ± 0.08
Spectral Rolloff	212.50 ± 0.15
Spectral Centroid	214.68 ± 0.13
Spectral Flatness	218.31 ± 0.16
Spectral Bandwidth	244.29 ± 0.16
Spectral Contrast	347.92 ± 0.22
Polynomial Features	367.24 ± 0.58
Mel Spectrogram	435.85 ± 0.57
Chroma STFT	548.00 ± 0.42
Mel-frequency Cepstral Coefficients	681.47 ± 0.44
Mel-frequency Cepstral Coefficients (Delta2)	705.78 ± 0.44
Mel-frequency Cepstral Coefficients (Delta)	716.52 ± 0.44
Tonnetz	737.37 ± 0.45
Chroma CENS	10573.66 ± 1.75
Chroma CQT	14938.69 ± 1.80

2. **TFET estimation:** The single hop TFET was obtained for all features by summing the estimated IFETs. As shown in Equation (5.3), the TFET for a single hop when extracting all features ($\text{TFET}_{Allfeatures}$) compromises the MAPT compliance condition;

$$\begin{aligned} TEPT_{\text{All features}} &< MAPT \\ TFET_{\text{All features}} + CIT &< MAPT \\ 62,631\ \mu s + CIT &\not< 23,220\ \mu s \end{aligned} \tag{5.3}$$

3. **Exclusion of feature with the highest IFET:** The features with the highest IFETs (CENS Chromagram and CQT Chromagram) were excluded. As outlined in Equation (5.4), under these conditions:

   (a) The MAPT compliance condition is not compromised by the TFET alone;

   (b) There is an available slack for the CIT, of $17,417\ \mu$s.

$$\begin{aligned} TEPT_{\text{W/o CENS and CQT Chromagrams}} &< MAPT \\ TFET_{\text{W/o CENS and CQT Chromagrams}} + CIT &< MAPT \\ 5,803\ \mu s + CIT &< 23,220\ \mu s \\ CIT &< 17,417\ \mu s \end{aligned} \tag{5.4}$$

5.3.8.3 Feature selection based on Pearson correlation, Spearman correlation, Analysis of Variance F-value and Random Forest Importance

The feature selection process described in the Subsubsection *"Feature selection based on Pearson correlation, Spearman correlation, ANOVA F-value and Random Forest Importance"* of the Chapter *"Methodology"* (4.4.8.3) was conducted on the feature subset obtained in last Subsubsection 5.3.8.1. The objective was the selection of featues based on 4 metrics: PCC, SCC, ANOVA F-value, and RFI. The analysed features, along with their scores for each of the 4 metrics evaluated, as well as their corresponding average scores, are presented in Table A.2.

The top performing 49 features were selected as the SFS, based on their average scores.

Then, the SFS features were analysed and classified into 12 distinct groups of features. The feature groups and their corresponding contributions within the SFS were listed, as follows:

1. **MFCCs**: This category dominates the feature subset, comprising 16 features that account for 32.65% of the subset;

2. **STFT Chromagram**: The features in this category contribute the second-largest portion of the subset, with 12 features amounting to 24.49%;

3. **Tonnetz**: This category consists of 6 features, contributing to 12.24% of the subset;

4. **Spectral contrast**: This category contributes with 5 features or 10.20% of the subset;

5. **Poly features**: This category includes 2 features, representing 4.08% of the subset;

6. **MFCCs Delta**: This category also contributes with 2 features, amounting to 4.08% of the subset;

7. **Additional single features**: The following categories have only one feature, contributing each to 2.04% of the subset:
   (a) RMS;
   (b) Spectral Bandwidth;
   (c) Spectral Rolloff;
   (d) Spectral Flatness;
   (e) Spectral Centroid;
   (f) ZCR.

A visualization of the SFS composition is presented in Figure 5.1.

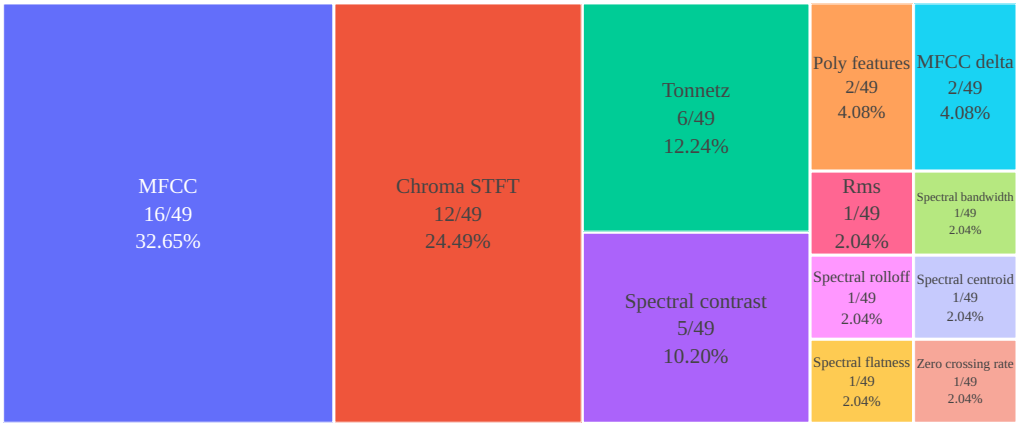

Figure 5.1: Selected Features Subset composition.

5.4 Selection and design of DL-based model architectures

This Section presents the selected and designed architectures, by conducting the method described in the Section *"Selection and design of DL-based model architectures"* of the *"Methodology"* Chapter (4.5). The architectures selection and design was based in two principles: demonstrated empirical success, and trainable parameters balancing (to about 200,000 parameters). It resulted in several architectures, namely CNN (5.4.1), Separable CNN (5.4.2), ResNet (5.4.3), LSTM (5.4.4), GRU (5.4.5), TCN (5.4.6), and Transformer (5.4.7). For more detailed information on these architectures, refer to Subsection *"Deep Learning-based models"* of the Chapter *"Background"* (2.4.1).

5.4.1 Convolutional Neural Network

The first layer of the CNN architecture is a spatial dropout with a rate of 0.2 applied to the input tensor. The next 6 layers are 1D convolutional layers with a kernel size of 5, Rectified Linear

Unit (ReLU) activation, padding "same", each one followed by batch normalization. The numbers of filters for each layer are 34, 34, 68, 68, 136, and 136, and the dilation rates are 0, 2, 4, 6, 8, 10. Next, a global average pooling layer reduces the dimensions, followed by a dense layer with ReLU activation and 136 units. Then, a dropout layer with a rate of 0.5 is used for regularization purposes. Finally, a dense layer with sigmoid activation produces the binary classification output.

5.4.2 Separable Convolutional Neural Network

The first layer of the Separable-CNN architecture consists of one spatial dropout of 0.2 applied to the input tensor. The next 6 layers are 1D separable convolutional layers with a kernel size of 5, ReLU activation function, padding "same", each one followed by batch normalization. The numbers of filters for each layer are 64, 64, 128, 128, 256, and 256, and the dilation rates are 0, 2, 4, 6, 8, and 10. A global average pooling is applied to reduce the dimensions, followed by a dense layer with ReLU activation and 256 units, and a dropout layer with a rate of 0.5. Finally, a dense layer with 1 unit and sigmoid activation produces the binary classification output.

5.4.3 Residual Network

The first layer of the ResNet architecture is a spatial dropout layer with a rate of 0.2 applied to the input tensor. The next, is a 1D convolutional layer with 16 filters and a kernel size of 3. The next 5 layers are residual blocks, each one followed by batch normalization. The number of filters of each block are 32, 32, 64, 64, 128. The dilation rates are 2, 2, 4, 4, 8. Next, a global average pooling layer is used to reduce dimensions, followed by a dense layer with 128 units and ReLU activation. After, batch normalization is applied, followed by a dropout layer with a rate of 0.5. Finally, a dense layer with 1 unit and sigmoid activation produces the binary classification output.

5.4.4 Long Short-Term Memory

The first layer of the LSTM architecture is a Spatial dropout of 0.2 applied to the input tensor. Then, 6 LSTM layers take place, each one returning sequences and followed by batch normalization: 2 with 25 units, 2 with 50 units and 2 with 100 units. Next, global average pooling is applied to reduce the dimensions, followed by a dense layer with ReLU activation and 100 units. Then, a dropout layer with a rate of 0.5 is used for regularization purposes. Finally, a dense layer with sigmoid activation produces the binary classification output.

5.4.5 Gated Recurrent Unit

The first layer of the GRU architecture is a spatial dropout of 0.2 applied to the input tensor. Next, there are 6 GRU layers returning sequences, each followed by batch normalization: 2 with 29 units, 2 with 58 and the last 2 with 116. Then, global average pooling is applied to reduce the dimensions, followed by a dense layer with ReLU activation and 116 units, and dropout layer

with a rate of 0.5. Finally, a dense layer with sigmoid activation produces the binary classification output.

5.4.6 Temporal Convolutional Network

The first layer of the TCN architecture is a TCN layer with:

- 64 filters in each stage;
- Kernel size of 3;
- 2 stacks;
- Dilation rates of 1, 2, 4, and 8;
- Skip connections enabled;
- Dropout rate of 0.2 applied to the TCN output;
- ReLU activation;
- Batch normalization enabled.

Then, a dense layer with 64 units and ReLU activation is applied to the TCN output, followed by a dropout layer, with a rate of 0.5. Finally, a dense layer with 1 unit and sigmoid activation produces the binary classification output.

5.4.7 Transformer

The first layer of the Transformer architecture is a spatial dropout of 0.2 applied to the input tensor. Next, a dense layer with 64 units takes place, followed by batch normalization. 4 transformer blocks follow, with: a multi-head attention layer with 4 attention heads and a size of each attention head for query and key of 16; a dropout layer with a rate of 0.2; layer normalization after each self-attention; a FNN with two dense layers, ReLU activation, and 256 hidden units; a dropout layer with a rate of 0.1 after the FNN; layer normalization after each FNN; batch normalization after each transformer block. Global average pooling is applied to the transformer blocks' output, followed by a dense layer with ReLU activation and 64 units. Then, a dropout layer with a rate of 0.5 takes place. Finally, a dense layer with sigmoid activation produces the binary classification output.

5.5 Assessment and comparison of all model/features subset pairs

The results of the procedures described in Section *"Assessment and comparison of all model/features subset pairs"* of the Chapter *"Methodology"* are presented next, encompassing TTS using the BFS (5.5.1), TTS using the SFS (5.5.2), performance comparison across features subsets (5.5.3), and selection of the best performing model/features subset pair (5.5.4).

5.5.1 Train-Test Split using the baseline features subset

5 iterations of TTS evaluation using the 49 MFCCs BFS were conducted on the following models: TCN, GRU, LSTM, ResNet, CNN, Separable CNN, and Transformer. Evaluation metrics were retrieved, namely performance (5.5.1.1) and computational (5.5.1.2) metrics.

5.5.1.1 Performance metrics

The performance metrics obtained from the 5 iterations of TTS evaluation using the 49 MFCCs BFS are presented in Table 5.2 and Figure 5.2, including:

- **Accuracy:**
  - The LSTM model tops the list with an Accuracy of 96.51% ± 0.18%;
  - It is followed by the Separable CNN model (96.10% ± 0;20%), the GRU model (96.02% ± 0.44%), the CNN model (96.00% ± 0.12%), and the ResNet model (95.88% ± 0.24%);
  - The TCN model has an Accuracy of 95.47% ± 0.23%;
  - The Transformer model performs the lowest with an Accuracy of 92.39% ± 0.62%.
- **Precision:**
  - Again, the LSTM model is at the top with a Precision of 96.75% ± 0.38%;
  - The Separable CNN model follows closely with 96.37% ± 0.55%;
  - Next, we have the GRU model (96.35% ± 0.62%), the ResNet model (96.23% ± 0.81%), and the TCN model (95.77% ± 0.62%);
  - The CNN model has a Precision of 95.99% ± 0.27%;
  - The Transformer model performs the lowest in this category with 92.63% ± 0.95%.
- **Recall:**
  - Once more, the LSTM model leads with a Recall of 96.25% ± 0.63%;
  - It is followed by the GRU model (95.66% ± 0.65%), the Separable CNN model (95.80% ± 0.31%), the CNN model (96.01% ± 0.35%), and the ResNet model (95.49% ± 0.69%);
  - The TCN model has a Recall of 95.14% ± 0.55%;
  - The Transformer model ranks last with a Recall of 92.10% ± 1.55%.
- **Specificity:**
  - The LSTM model continues to lead with a Specificity of 96.77% ± 0.39%;
  - The GRU, Separable CNN, and ResNet models follow with Specificity scores of 96.38% ± 0.61%, 96.39% ± 0.59%, and 96.27% ± 0.83%, respectively;

  - The TCN model's Specificity is 95.81% ± 0.61%, while the CNN model's is 95.99% ± 0.33%;
  - The Transformer model performs the lowest with a Specificity of 92.68% ± 1.06%.

- **F1 Score:**
  - The LSTM model has the highest F1 Score with 96.50% ± 0.19%;
  - The Separable CNN model has an F1 Score of 96.08% ± 0.17%, followed by the GRU model with 96.00% ± 0.44%, the CNN model with 96.00% ± 0.12%, and the ResNet model with 95.86% ± 0.23%;
  - The TCN model has an F1 Score of 95.45% ± 0.25%;
  - The Transformer model has the lowest F1 Score of 92.35% ± 0.65%.

- **AUC-ROC:**
  - The LSTM model again takes the top spot with an AUC-ROC of 99.53% ± 0.06%;
  - It is followed by the GRU and Separable CNN models, both with an AUC-ROC of 99.40% ± 0.11% and 99.40% ± 0.06%, respectively;
  - The CNN and ResNet models have AUC-ROC values of 99.36% ± 0.03% and 99.33% ± 0.08%, respectively;
  - The TCN model has an AUC-ROC of 99.19% ± 0.06%;
  - The Transformer model performs the lowest with an AUC-ROC of 97.96% ± 0.26%.

In this TTS evaluation of various classifiers using the baseline 49 MFCCs BFS, the LSTM model consistently demonstrated superior performance across multiple metrics. The Separable CNN model also performed well, closely following the LSTM model in most metrics. The GRU, CNN, and ResNet models showed competitive results but slightly lower than the top performers. The TCN model achieved relatively lower scores in comparison. The Transformer model consistently had the lowest performance across all metrics, indicating its limitations in this particular task.

Based on these results, it can be concluded that the LSTM model is the most effective classifier when using the 49 MFCCs BFS as input, showcasing strong performance and robustness.

5.5.1.2 Computational metrics

The computational metrics obtained from the 5 iterations of TTS evaluation using the 49 MFCCs BFS are presented in Table 5.3 and Figure 5.3 including:

- **Number of parameters:**
  - The CNN model has the highest number of parameters with 208,659, closely followed by the Transformer model with 208,641 parameters;

Table 5.2: Classifiers' performance metrics obtained from 5 iterations of Train-Test Split evaluation using the 49 MFCCs Baseline Features Subset.

Model	**Accuracy**	**Precision**	**Recall**	**Specificity**	**F1 Score**	**AUC**
TCN	95.47 ± 0.23	95.77 ± 0.62	95.14 ± 0.55	95.81 ± 0.61	95.45 ± 0.25	99.19 ± 0.06
GRU	96.02 ± 0.44	96.35 ± 0.62	95.66 ± 0.65	96.38 ± 0.61	96.00 ± 0.44	99.40 ± 0.11
LSTM	96.51 ± 0.18	96.75 ± 0.38	96.25 ± 0.63	96.77 ± 0.39	96.50 ± 0.19	99.53 ± 0.06
ResNet	95.88 ± 0.24	96.23 ± 0.81	95.49 ± 0.69	96.27 ± 0.83	95.86 ± 0.23	99.33 ± 0.08
CNN	96.00 ± 0.12	95.99 ± 0.27	96.01 ± 0.35	95.99 ± 0.33	96.00 ± 0.12	99.36 ± 0.03
Separable CNN	96.10 ± 0.20	96.37 ± 0.55	95.80 ± 0.31	96.39 ± 0.59	96.08 ± 0.17	99.40 ± 0.06
Transformer	92.39 ± 0.62	92.63 ± 0.95	92.10 ± 1.55	92.68 ± 1.06	92.35 ± 0.65	97.96 ± 0.26

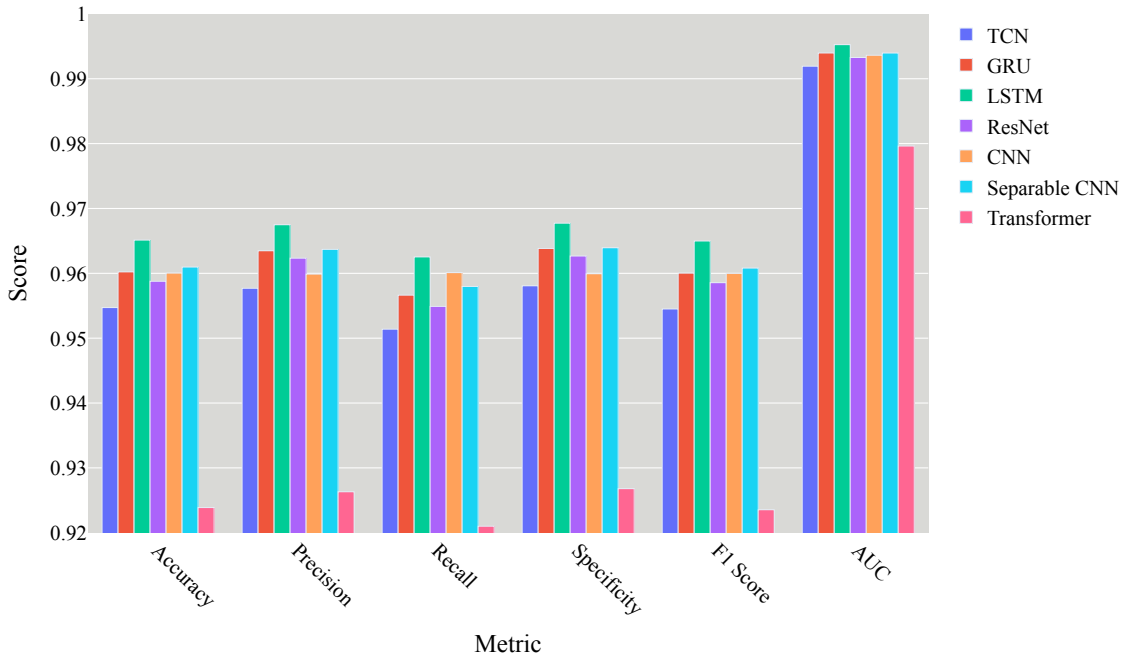

Figure 5.2: Classifiers' performance metrics obtained from 5 iterations of Train-Test Split evaluation using the 49 MFCCs Baseline Features Subset.

  - The TCN and GRU models have 206,273 and 206,191 parameters, respectively;
  - The Separable CNN model has slightly fewer parameters at 204,086;
  - The LSTM model has 200,401 parameters, and finally, the ResNet model has the fewest parameters with 198,913.

- **Training time:**
  - The Separable CNN model requires the most time for training, with a mean of 43.75 ± 12.95 minutes;
  - The Transformer model follows with 18.71 ± 4.31 minutes;
  - Both the LSTM and ResNet models require 16.28 ± 2.14 and 16.28 ± 2.81 minutes, respectively. The TCN model has a slightly lower training time with 16.40 ± 2.85 minutes;
  - The GRU model requires 15.25 ± 6.81 minutes, and the CNN model has the shortest training time of 11.89 ± 2.78 minutes;

- **Number of epochs:**
  - The LSTM model takes the most epochs to complete training, with a mean of 55.40 ± 10.81;
  - It is followed by the GRU model with 49.20 ± 21.14 epochs;
  - The ResNet model completes training in 44.20 ± 11.17 epochs, while the Separable CNN model takes 41.00 ± 14.49 epochs;
  - The TCN model needs 33.80 ± 3.49 epochs, and the CNN model requires 36.60 ± 8.73 epochs;
  - The Transformer model takes the fewest epochs, with a mean of 35.00 ± 6.63 epochs.

- **Best epoch:**
  - The LSTM model has the highest best epoch value of 49.40 ± 10.81, followed by the GRU model at 43.20 ± 21.14;
  - The ResNet model's best epoch is at 38.20 ± 11.17, while the Separable CNN model is at 35.00 ± 14.49;
  - The TCN model's best epoch is at 27.80 ± 3.49, and the CNN model's best epoch is 30.60 ± 8.73;
  - The Transformer model has the earliest best epoch at 29.00 ± 6.63.

- **Average epoch training time:**
  - The Separable CNN model takes the most time per epoch, with 1.09 ± 0.15 minutes;

  - It is followed by the Transformer and TCN models, with 0.53 ± 0.06 and 0.48 ± 0.05 minutes, respectively;
  - The ResNet model requires 0.38 ± 0.04 minutes per epoch, while the CNN model takes 0.33 ± 0.07 minutes;
  - The LSTM and GRU models require the least time per epoch, with 0.30 ± 0.03 and 0.31 ± 0.03 minutes, respectively.

- **Inference time:**
  - The LSTM and Transformer models have the highest inference times, with 0.26 ± 0.08 ms and 0.27 ± 0.06 ms, respectively;
  - The GRU model follows with 0.24 ± 0.08 ms;
  - The TCN model has an inference time of 0.20 ± 0.07 ms, while the ResNet model requires 0.18 ± 0.07 ms;
  - Both the CNN and Separable CNN models have the shortest inference times, with 0.12 ± 0.04 ms and 0.12 ± 0.03 ms, respectively.

These findings highlight the trade-offs between number of parameters, training time, epoch count, and inference time for each model, enabling informed decision-making based on specific computational requirements.

5.5.2 Train-Test Split using the selected features subset

5 iterations of TTS evaluation using the SFS were conducted on the following models: TCN, GRU, LSTM, ResNet, CNN, Separable CNN, and Transformer. Evaluation metrics were retrieved, namely performance (5.5.2.1) and computational (5.5.2.2).

5.5.2.1 Performance metrics

The performance metrics obtained from the 5 iterations of TTS evaluation using the SFS are presented in Table 5.4 and Figure 5.4, including:

- **Accuracy:**
  - The TCN model leads with an Accuracy of 98.72% $\pm$ 0.16%;
  - It is followed by the GRU model (98.26% $\pm$ 0.20%), the LSTM model (98.22% $\pm$ 0.34%), the ResNet model (97.45% $\pm$ 0.37%), the CNN model (97.32% $\pm$ 0.22%), and the Separable CNN model (97.15% $\pm$ 0.29%);
  - The Transformer model performs the lowest with an Accuracy of 94.77% $\pm$ 0.51%.

- **Precision:**
  - The TCN model is at the top with a Precision of 98.71% $\pm$ 0.22%;

Table 5.3: Classifiers' computational metrics obtained from 5 iterations of Train-Test Split evaluation using the 49 MFCCs Baseline Features Subset.

Model	Num. parameters	Training time (min)	Num. epochs	Best epoch	Avg. epoch training time (min)	Inference time (ms)
TCN	206273	16.40 ± 2.85	33.80 ± 3.49	27.80 ± 3.49	0.48 ± 0.05	0.20 ± 0.07
GRU	206191	15.25 ± 6.81	49.20 ± 21.14	43.20 ± 21.14	0.31 ± 0.03	0.24 ± 0.08
LSTM	200401	16.28 ± 2.14	55.40 ± 10.81	49.40 ± 10.81	0.30 ± 0.03	0.26 ± 0.08
ResNet	198913	16.28 ± 2.81	44.20 ± 11.17	38.20 ± 11.17	0.38 ± 0.04	0.18 ± 0.07
CNN	208659	11.89 ± 2.78	36.60 ± 8.73	30.60 ± 8.73	0.33 ± 0.07	0.12 ± 0.04
Separable CNN	204086	43.75 ± 12.95	41.00 ± 14.49	35.00 ± 14.49	1.09 ± 0.15	0.12 ± 0.03
Transformer	208641	18.71 ± 4.31	35.00 ± 6.63	29.00 ± 6.63	0.53 ± 0.06	0.27 ± 0.06

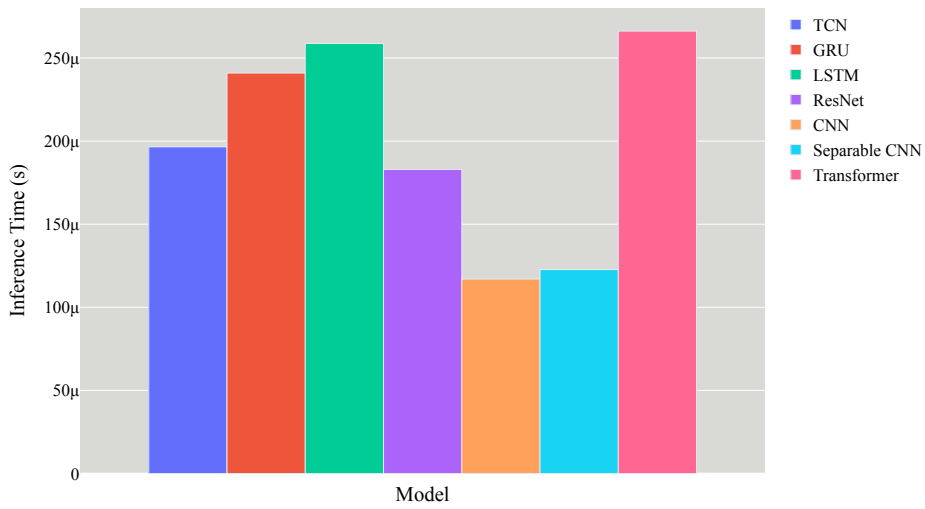

Figure 5.3: Classifiers' inference time obtained from 5 iterations of Train-Test Split evaluation using the 49 MFCCS Baseline Features Subset.

  - The LSTM model follows closely with 98.37% ± 0.31%.
  - Next, appears the GRU model (98.12% ± 0.17%), the ResNet model (97.61% ± 0.49%), and the Separable CNN model (97.35% ± 0.50%);
  - The CNN model has a Precision of 97.21% ± 0.41%;
  - The Transformer model performs the lowest in this category with 95.39% ± 0.42%.

- **Recall:**
  - The TCN model leads with a Recall of 98.74% ± 0.22%;
  - It is followed by the GRU model (98.41% ± 0.29%), the LSTM model (98.07% ± 0.48%), the CNN model (97.44% ± 0.55%), and the ResNet model (97.29% ± 0.57%);
  - The Separable CNN model has a Recall of 96.94% ± 0.34%;
  - The Transformer model ranks last with a Recall of 94.08% ± 1.17%.

- **Specificity:**
  - The TCN model leads with a Specificity of 98.71% ± 0.23%;
  - The LSTM model, GRU model, and Separable CNN model follow with Specificity scores of 98.37% ± 0.33%, 98.12% ± 0.18%, and 97.37% ± 0.53%, respectively;
  - The ResNet model's Specificity is 97.61% ± 0.51%, while the CNN model's is 97.20% ± 0.45%;
  - The Transformer model performs the lowest with a Specificity of 95.46% ± 0.48%.

- **F1 Score:**
  - The TCN model has the highest F1 Score with 98.72% ± 0.15%;
  - The GRU model has an F1 Score of 98.26% ± 0.20%, followed by the LSTM model (98.22% ± 0.34%), the ResNet model (97.45% ± 0.37%), the CNN model (97.32% ± 0.22%), and the Separable CNN model (97.14% ± 0.27%);
  - The Transformer model has the lowest F1 Score of 94.72% ± 0.56%.

- **AUC-ROC:**
  - The TCN model again takes the top spot with an AUC-ROC of 99.91% ± 0.01%;
  - It is followed by the GRU model (99.84% ± 0.03%), the LSTM model (99.82% ± 0.04%), the ResNet model (99.70% ± 0.09%), the CNN model (99.68% ± 0.05%), and the Separable CNN model (99.66% ± 0.06%);
  - The Transformer model performs the lowest with an AUC-ROC of 99.03% ± 0.16%.

In this TTS evaluation of various classifiers using the SFS as input, the TCN model consistently outperformed the others in terms of Accuracy, Precision, Recall, Specificity, F1 score, and AUC-ROC. The GRU and LSTM models also performed well, closely trailing the TCN model in most metrics. However, the Transformer model consistently had the lowest scores across all performance metrics. Based on these results, it can be concluded that the TCN model is the most effective classifier for the given task when using the SFS as input, demonstrating superior performance and robustness.

5.5.2.2 Computational metrics

The computational metrics obtained from the 5 iterations of TTS evaluation using the SFS are presented in Table 5.5 and Figure 5.5, including:

- **Number of parameters:**
  - The CNN model has the highest number of parameters with 208,659, closely followed by the Transformer model with 208,641 parameters;
  - The TCN and GRU models have 206,273 and 206,191 parameters, respectively;
  - The Separable CNN model has slightly fewer parameters at 204,086;
  - The LSTM model has 200,401 parameters, and finally, the ResNet model has the fewest parameters with 198,913.
- **Training time:**
  - The Separable CNN model requires the most time for training, with a mean of 31.15 ± 8.02 minutes;
  - The TCN model follows with 25.13 ± 5.03 minutes;
  - The LSTM model requires 17.16 ± 4.24 minutes, and the GRU and Transformer models require 16.73 ± 3.14 and 16.60 ± 2.86 minutes, respectively;
  - The ResNet model has a slightly lower training time with 15.49 ± 5.46 minutes, and the CNN model has the shortest training time of 10.04 ± 1.61 minutes.
- **Number of epochs:**
  - The LSTM model takes the most epochs to complete training, with a mean of 60.00 ± 17.18;
  - It is followed by the GRU model with 56.80 ± 11.30 epochs and the TCN model with 54.80 ± 12.05 epochs;
  - The ResNet model completes training in 43.00 ± 15.56 epochs, while the Transformer model takes 33.00 ± 5.66 epochs;
  - The CNN model requires 34.40 ± 6.23 epochs, and the Separable CNN model takes the fewest epochs, with a mean of 30.80 ± 8.07 epochs.

Table 5.4: Classifiers' performance metrics obtained from 5 iterations of Train-Test Split evaluation using the 49 Selected Features Subset.

Model	**Accuracy**	**Precision**	**Recall**	**Specificity**	**F1 Score**	**AUC**
TCN	98.72 ± 0.16	98.71 ± 0.22	98.74 ± 0.22	98.71 ± 0.23	98.72 ± 0.15	99.91 ± 0.01
GRU	98.26 ± 0.20	98.12 ± 0.17	98.41 ± 0.29	98.12 ± 0.18	98.26 ± 0.20	99.84 ± 0.03
LSTM	98.22 ± 0.34	98.37 ± 0.31	98.07 ± 0.48	98.37 ± 0.33	98.22 ± 0.34	99.82 ± 0.04
ResNet	97.45 ± 0.37	97.61 ± 0.49	97.29 ± 0.57	97.61 ± 0.51	97.45 ± 0.37	99.70 ± 0.09
CNN	97.32 ± 0.22	97.21 ± 0.41	97.44 ± 0.55	97.20 ± 0.45	97.32 ± 0.22	99.68 ± 0.05
Separable CNN	97.15 ± 0.29	97.35 ± 0.50	96.94 ± 0.34	97.37 ± 0.53	97.14 ± 0.27	99.66 ± 0.06
Transformer	94.77 ± 0.51	95.39 ± 0.42	94.08 ± 1.17	95.46 ± 0.48	94.72 ± 0.56	99.03 ± 0.16

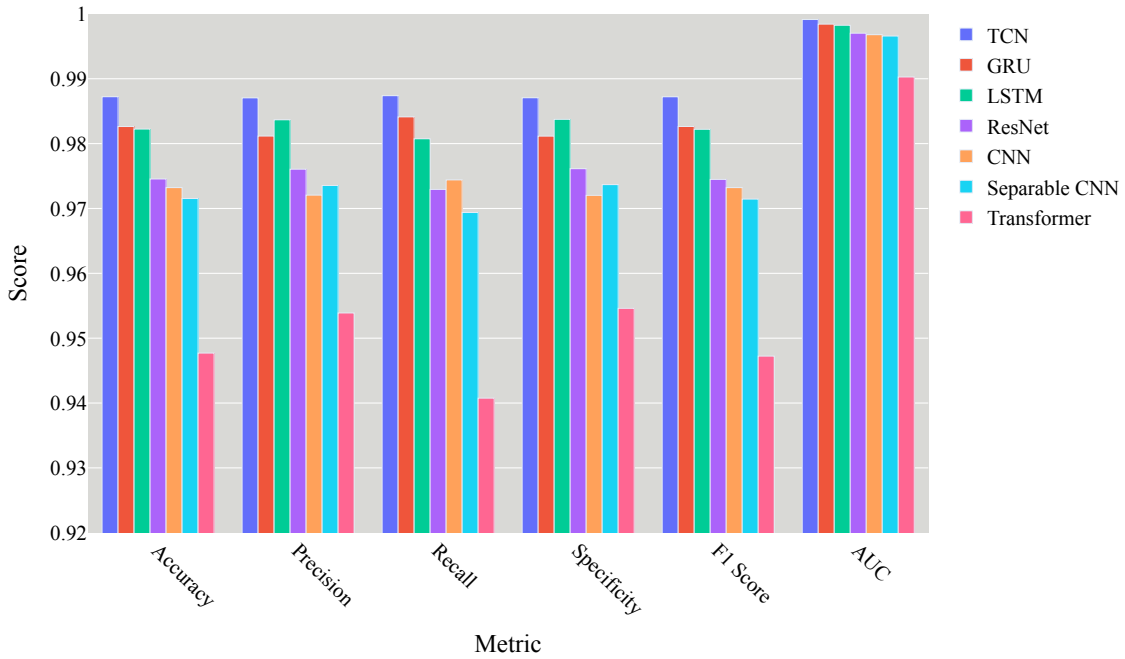

Figure 5.4: Classifiers' performance metrics obtained from 5 iterations of Train-Test Split evaluation using the 49 Selected Features Subset

- **Best epoch:**
  - The LSTM model has the highest best epoch value of 54.00 ± 17.18, followed by the GRU model at 50.80 ± 11.30 and the TCN model at 48.80 ± 12.05;
  - The ResNet model's best epoch is at 37.00 ± 15.56, while the Transformer and CNN models' best epochs are at 27.00 ± 5.66 and 28.40 ± 6.23, respectively;
  - The Separable CNN model has the earliest best epoch at 24.80 ± 8.07.
- **Average epoch training time:**
  - The Separable CNN model takes the most time per epoch, with 1.01 ± 0.01 minutes;
  - It is followed by the Transformer and TCN models, with 0.50 ± 0.01 and 0.46 ± 0.01 minutes, respectively;
  - The ResNet model requires 0.36 ± 0.01 minutes per epoch, while the LSTM, GRU, and CNN models require the least time per epoch, each with 0.29 ± 0.01 or 0.29 ± 0.02 minutes.
- **Inference time:**
  - The LSTM model has the highest inference time, with 0.28 ± 0.09 ms;
  - The Transformer model has a slightly lower inference time of 0.27 ± 0.06 ms, followed by the GRU model with 0.25 ± 0.08;
  - The TCN and ResNet models require 0.22 ± 0.08 and 0.20 ± 0.07 ms, respectively;
  - The Separable CNN and CNN models have the shortest inference times, with 0.13 ± 0.03 ms and 0.12 ± 0.04 ms, respectively.

These findings highlight the trade-offs between number of parameters, training time, epoch count, and inference time for each model, enabling informed decision-making based on specific computational requirements.

5.5.3 Performance comparison across features subsets

The SFS displays superior performance over the BFS. This is confirmed by the improved performance metrics observed across all models, as depicted in Figures 5.6 and 5.7.

Table 5.6 presents the verified performance absolute gains obtained by using the SFS instead of the BFS. For instance, the TCN model exhibits absolute gains of 3.25% in Accuracy, 2.94% in Precision, 3.60% in Recall, 2.90% in Specificity, 3.27% in F1 Score, and 0.72% in AUC-ROC. Similar enhancements are verified in all other models.

This evidence supports the preference for the SFS. Its consistent higher performance suggests it better captures relevant information for this classification task.

Table 5.5: Classifiers' computational metrics obtained from 5 iterations of Train-Test Split evaluation using the 49 Selected Features Subset.

Model	**Num. parameters**	**Training time (min)**	**Num. epochs**	**Best epoch**	**Avg. epoch training time (min)**	**Inference time (ms)**
TCN	206273	25.13 ± 5.03	54.80 ± 12.05	48.80 ± 12.05	0.46 ± 0.01	0.22 ± 0.08
GRU	206191	16.73 ± 3.14	56.80 ± 11.30	50.80 ± 11.30	0.29 ± 0.01	0.25 ± 0.08
LSTM	200401	17.16 ± 4.24	60.00 ± 17.18	54.00 ± 17.18	0.29 ± 0.02	0.28 ± 0.09
ResNet	198913	15.49 ± 5.46	43.00 ± 15.56	37.00 ± 15.56	0.36 ± 0.01	0.20 ± 0.07
CNN	208659	10.04 ± 1.61	34.40 ± 6.23	28.40 ± 6.23	0.29 ± 0.01	0.12 ± 0.04
Separable CNN	204086	31.15 ± 8.02	30.80 ± 8.07	24.80 ± 8.07	1.01 ± 0.01	0.13 ± 0.03
Transformer	208641	16.60 ± 2.86	33.00 ± 5.66	27.00 ± 5.66	0.50 ± 0.01	0.27 ± 0.06

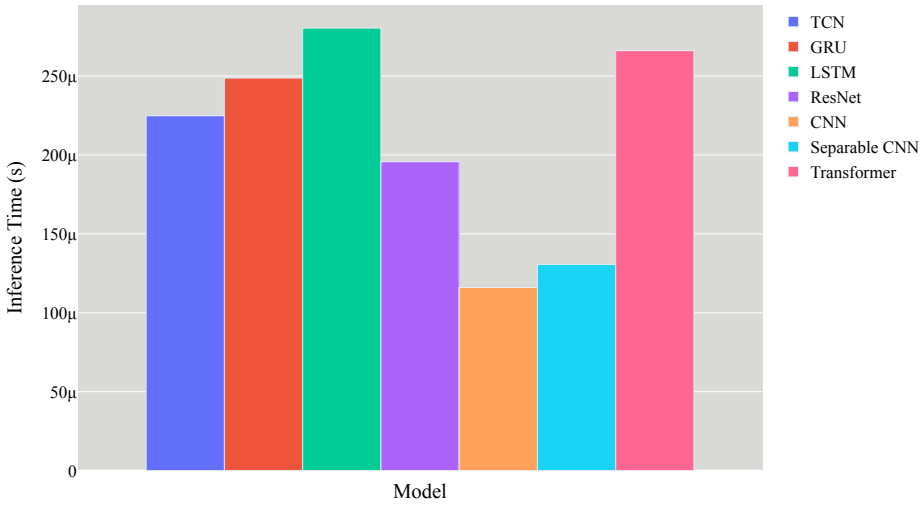

Figure 5.5: Classifiers' inference time obtained from 5 iterations of Train-Test Split evaluation using the 49 Selected Features Subset.

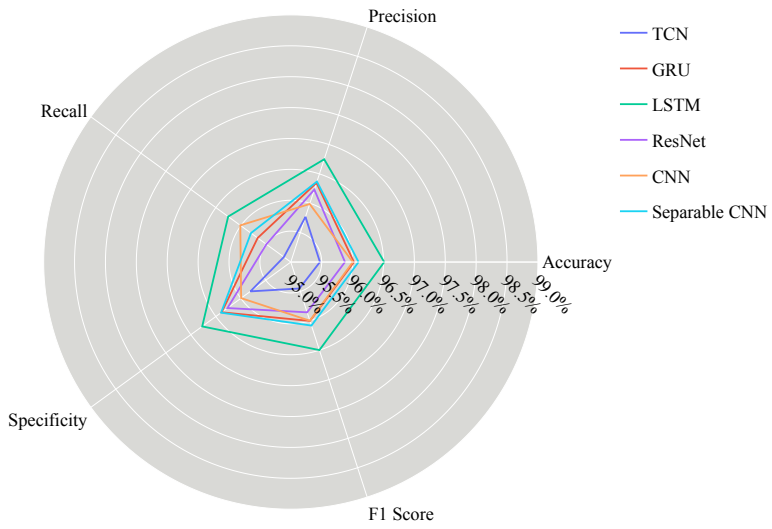

Figure 5.6: Classifiers' performance metrics obtained from 5 iterations of Train-Test Split evaluation using the 49 MFCCs Baseline Features Subset (web plot).

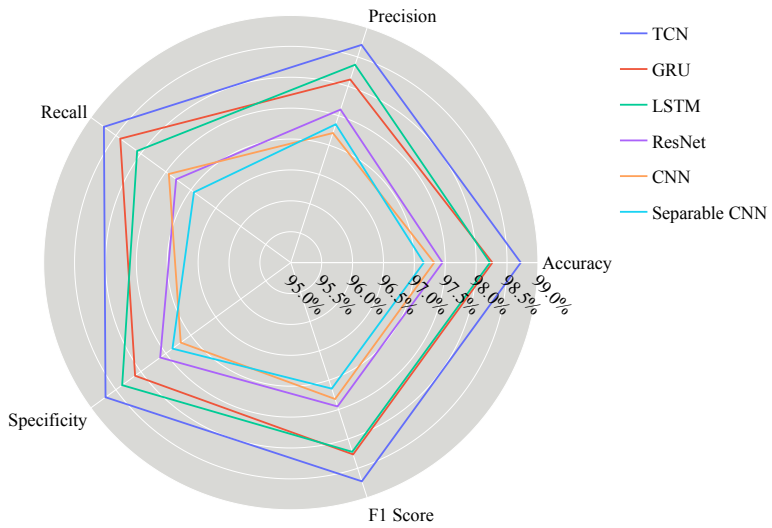

Figure 5.7: Classifiers' performance metrics obtained from 5 iterations of Train-Test Split evaluation using the 49 Selected Features Subset (web plot).

Table 5.6: Classifers' performance metrics absolute gains obtained by using the 49 Selected Features Subset instead of the 49 MFCCs Baseline Features Subset.

Model	Accuracy	Precision	Recall	Specificity	F1 Score	AUC
TCN	3.25	2.94	3.60	2.90	3.27	0.72
GRU	2.24	1.77	2.75	1.73	2.26	0.44
LSTM	1.71	1.62	1.82	1.60	1.72	0.30
ResNet	1.58	1.37	1.80	1.35	1.59	0.37
CNN	1.32	1.22	1.43	1.21	1.32	0.32
Separable CNN	1.06	0.99	1.14	0.97	1.06	0.26
Transformer	2.38	2.76	1.97	2.78	2.37	1.06

5.5.4 Selection of the best performing model/features subset pair

Based on the TTS evaluation of performance and computational metrics, the TCN model with the SFS as input emerges as the most favorable model/features subset pairing. This choice is influenced by the following reasons:

- **Performance metrics:** The TCN model outperformed all other considered models across all performance metrics. This model achieved the highest Accuracy, Precision, Recall, Specificity, F1 Score, and AUC-ROC. Such results demonstrate the model's ability to deliver accurate and consistent classifications;

- **Computational efficiency:** While the TCN model did not have the best computational metrics, they were well within acceptable limits given its superior predictive performance;

- **Feature importance:** The SFS has been determined to contain the most valuable information for the task at hand. Using this subset, the TCN model effectively leveraged the information contained in these features, yielding the best results;

- **Stability:** The standard deviations of the performance metrics for the TCN model are relatively low, indicating that the model performance is stable and not highly sensitive to variations in the data.

Therefore, considering the balance between performance, computational efficiency, and stability, the TCN model trained with the SFS has been chosen as the best performing model/features subset pair for the given task.

5.6 Assessment of the best performing model/features subset pair

The results of the procedures described in Section *"Assessment of the best performing model/features subset pair"* of the Chapter *"Methodology"* are presented next, encompassing performance assessment across articulation manner classes (5.6.1), K-FCV (5.6.2), exemplification of VD segmentation (5.6.3), and compliance with the MAPT (5.6.4).

5.6.1 Performance assessment across articulation manner classes

Based on the TTS evaluation results, the Accuracy of the best performing model/feature subset pair (TCN/SFS) was analysed across different articulation manner classes. Table 5.7 presents the results of this analysis.

The results reveal a high level of proficiency in VD decision across all articulation manners, substantiating the robustness of the model. The voiced plosives class obtained a relatively lower Accuracy, potentially due to underrepresentation.

Table 5.7: Temporal Convolutional Network model's classification Accuracy for each articulation manner class, obtained from the 5 iterations of the Train-Test Split evaluation using the 49 Selected Features Subset.

Phonetic class	Accuracy	Correct classifications	Total samples
voiced approximant	99.23 ± 0.20	3124 ± 29	3148 ± 30
vowel	99.07 ± 0.12	6141 ± 65	6198 ± 69
silence	98.87 ± 0.16	9101 ± 52	9205 ± 42
unvoiced fricative	98.36 ± 0.34	2051 ± 48	2085 ± 46
voiced nasal	97.90 ± 1.01	234 ± 20	239 ± 19
voiced fricative	97.87 ± 0.71	1685 ± 23	1722 ± 25
unvoiced plosive	97.22 ± 1.12	486 ± 15	500 ± 18
voiced plosive	94.46 ± 0.79	427 ± 20	452 ± 20

5.6.2 K-Fold Cross Validation

Performance and computational metrics were obtained from K-FCV evaluation of the TCN/SFS pair, using the GPU.

5.6.2.1 Performance metrics

As presented in Table 5.8, the TCN model achieved high performance in all performance metrics, with an Accuracy of 98.97% ± 0.19%, Precision of 99.00% ± 0.26%, Recall of 98.94% ± 0.17%, Specificity of 99.00% ± 0.26%, F1 Score of 98.97% ± 0.18%, and AUC-ROC of 99.94% ± 0.02%.

Table 5.8: Temporal Convolutional Network model's classification performance metrics obtained from K-Fold Cross Validation evaluation with a K of 5, using the 49 Selected Features Subset.

Model	Accuracy	Precision	Recall	Specificity	F1 Score	AUC
TCN	98.97 ± 0.19	99.00 ± 0.26	98.94 ± 0.17	99.00 ± 0.26	98.97 ± 0.18	99.94 ± 0.02

5.6.2.2 Computational metrics

Table 5.9 presents the computational metrics obtained from K-FCV evaluation of the TCN/SFS model/features subset pair. With 206,273 parameters, the TCN model required a training time of 26.59 minutes ± 5.35, with an average of 49.40 epochs ± 10.06. The best epoch achieved during training was at epoch 43.40 ± 10.06. Each epoch took an average of 0.54 minutes to train, with no observed variation. During inference, the TCN model exhibited an inference time of 0.28 ms ± 0.02.

Table 5.9: Temporal Convolutional Network model's computational metrics obtained from K-Fold Cross Validation evaluation with a K of 5, using the 49 Selected Features Subset.

Model	Num. parameters	Training time (min)	Num. epochs	Best epoch	Avg. epoch training time (min)	Inference time (ms)
TCN	206273	26.59 ± 5.35	49.40 ± 10.06	43.40 ± 10.06	0.54 ± 0.00	0.28 ± 0.02

5.6.3 Exemplification of voicing decision segmentation

The procedure described in Section *"Exemplification of VD segmentation"* of the Chapter *"Methodology"* was applied: the best performing model/features subset pair selected in 5.5.4 was tested on several tasks from the phonetically annotated WS/NS dataset to exemplify and visualize its operation. The tasks were carefully selected to encompass phones from all articulation manner classes, as follows: *"fisga"* (Figure 5.8); *"luta"* (Figure 5.9); *"nuca"* (Figure 5.10); *"zaro"* (Figure 5.11); *"viga"* (Figure 5.12).

This process allowed to verify visually the correct operation of the VD subsystem:

- A VD is obtained at each frame with FS of 512 samples;

- The VDs obtained correspond to the ground truth.

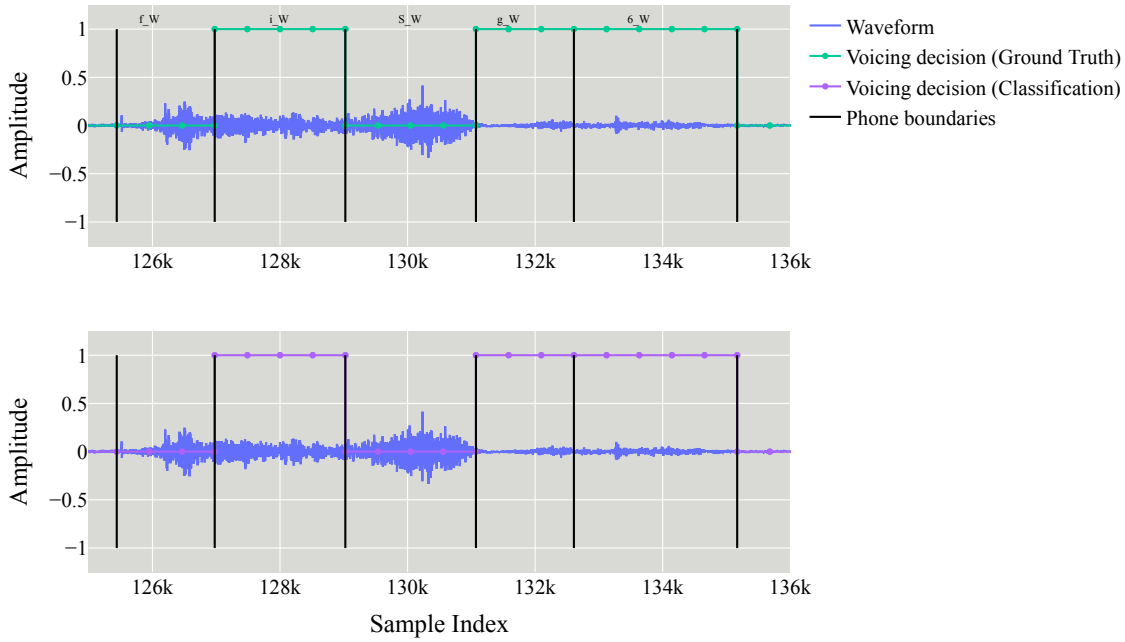

Figure 5.8: Voicing Decision segmentation example of the word "fisga".

5.6.4 Compliance with the Maximum Allowable Processing Time

To validate the system's capability of online operation, the method described in Section *"Compliance with the Maximum Allowable Processing Time"* of the Chapter *"Methodology"* (4.8.4) was followed for the TCN/SFS model/features subset pair.

The MAPT compliance condition was evaluated. For that purpose, the following steps were taken:

1. The pair's TFET was estimated from Table 5.1, as in Equation (5.5):

$$
\begin{aligned}
TFET &= \sum IFET \\
&= IFET_{MFCC} + IFET_{STFTChroma} + IFET_{Tonnetz} \\
&\quad + IFET_{SpectralContrast} + IFET_{PolyFeatures} + IFET_{MFCCDelta} \\
&\quad + IFET_{RMS} + IFET_{SpectralBandwidth} + IFET_{SpectralCentroid} \\
&\quad + IFET_{SpectralFlatness} + IFET_{SpectralRolloff} + IFET_{ZCR} \\
&= 4{,}479.55\ \mu s \\
&\approxeq 4.48\ ms
\end{aligned}
\tag{5.5}
$$

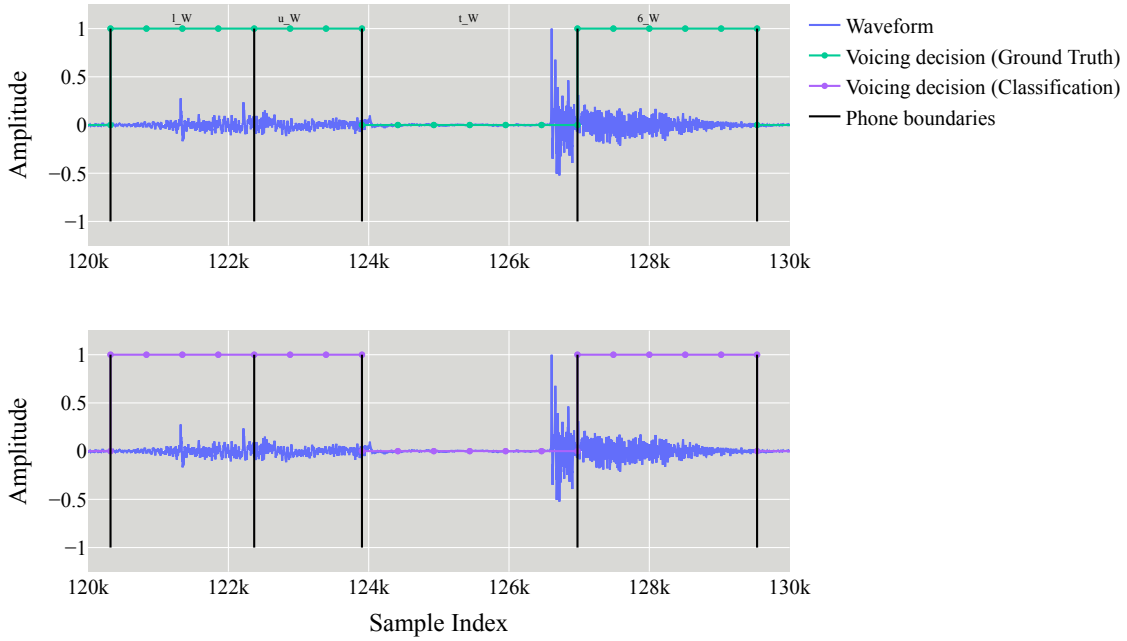

Figure 5.9: Voicing Decision segmentation example of the word "luta".

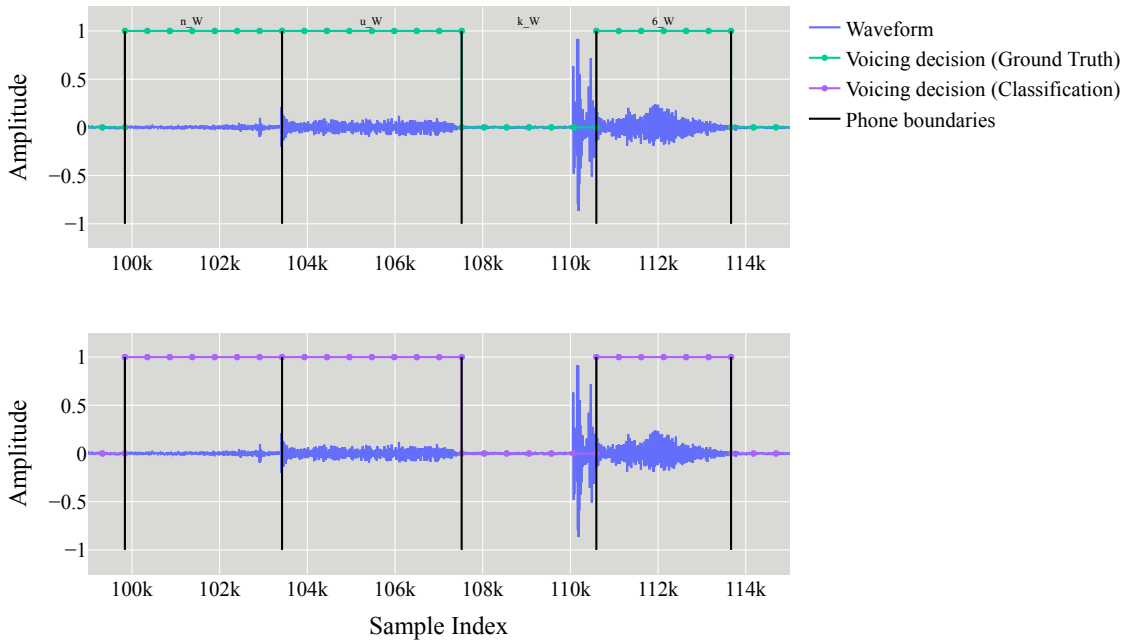

Figure 5.10: Voicing Decision segmentation example of the word "nuca".

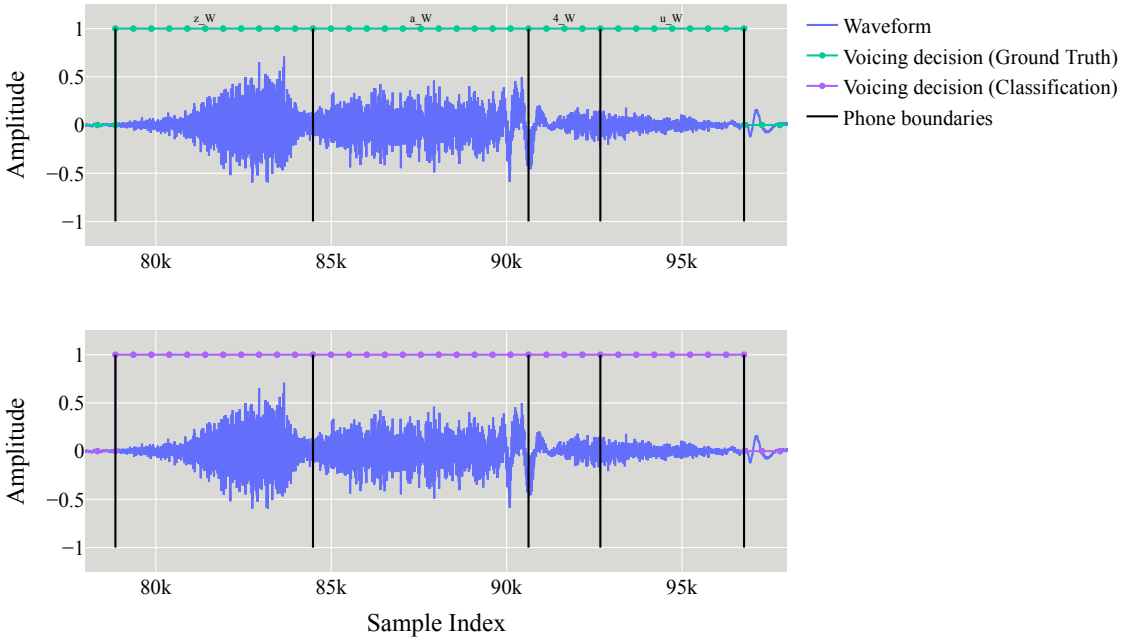

Figure 5.11: Voicing Decision segmentation example of the word "zaro".

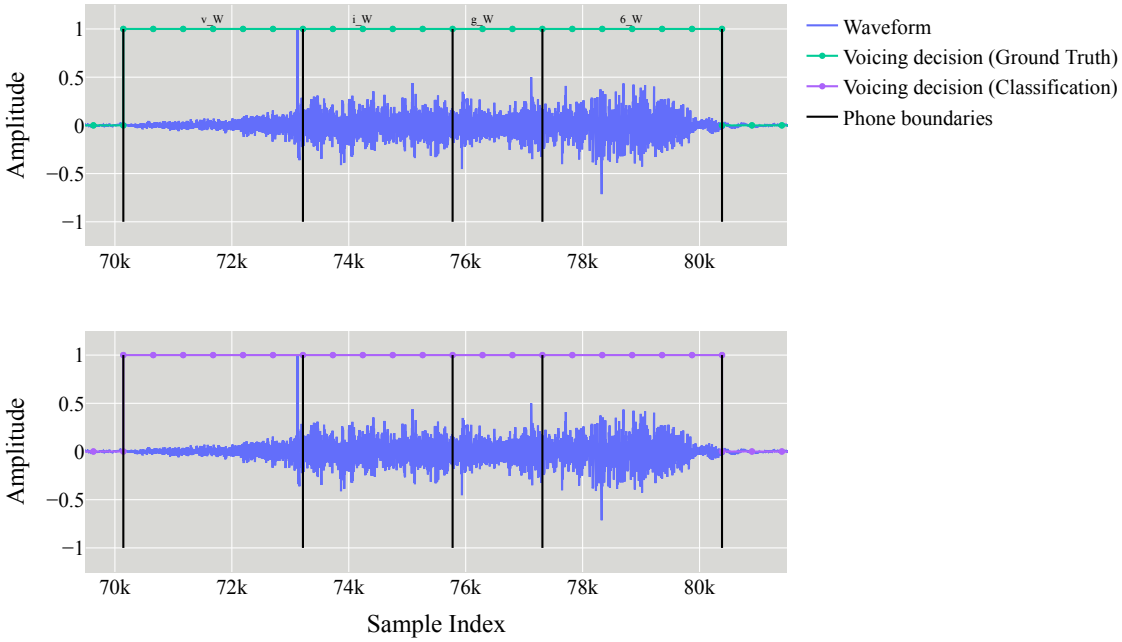

Figure 5.12: Voicing Decision segmentation example of the word "viga".

2. The pair's CIT was estimated as the average of the 5 iterations of TTS evaluation (refer to Table 5.5):

$$CIT = 0.22\ ms \tag{5.6}$$

3. The MAPT compliance condition was verified: the TEPT does not surpass the MAPT as expressed in Equation (5.7):

$$\begin{aligned} TEPT &< MAPT \\ TFET + CIT &< MAPT \\ \sum IFET + CIT &< HS \times \left(\frac{1}{SF}\right) \\ \sum IFET + CIT &< 512 \times \left(\frac{1}{22{,}050\ Hz}\right) \\ \sum IFET + CIT &< 23.22\ ms \\ 4.48\ ms + 0.22\ ms &< 23.22\ ms \\ 4.70\ ms &< 23.22\ ms \end{aligned} \tag{5.7}$$

Even though conservative estimates were used for both the MAPT and the IFETs, the condition was verified with a comfortable margin of 18.52 *ms*. This provides strong evidence that the online operation of the VD subsystem is viable. Additionally, it allows for a generous slack of 18.51 *ms* for the operation of the other subsystems within the broader whispered-to-normal conversion system.

5.7 Chapter summary

This Chapter *"Results and Discussion"* presented the findings of the research and includes discussions. It covered several sections, including the acquisition of the phonetically annotated WS/NS dataset (5.1), dataset preprocessing (5.2), feature engineering (5.3), selection and design of DL-based model architectures (5.4), assessment and comparison of all model/features subset pairs (5.5), and the assessment of the best performing model/features subset pair (5.6).

In the next Chapter *"Conclusions"*, the overall conclusions drawn from the research will be presented, including a summary of the key findings (6.1), the answer to the research questions through validation of the hypotheses (6.2), the contributions, innovations, and implications of the research work (6.3), as well as the limitations encountered and proposed avenues for future work to address them (6.4).

Chapter 6

Conclusions

In this Chapter, the conclusions from the research work are presented, encompassing a summary of the key findings (6.1), the revisiting of the research question and hypotheses (6.2), the statement of the contributions, innovations, implications (6.3), limitations and proposed future work (6.4) of the research work.

6.1 Summary of key findings

The key findings obtained with the application of the proposed methodologies are summarized and presented next, encompassing the dataset preprocessing (6.1.1), feature engineering (6.1.2), selection and design of DL-based model architectures (6.1.3), assessment and comparison of all model/features subset pairs (6.1.4), and assessment of the best performing model/features subset pair (6.1.5).

6.1.1 Dataset preprocessing

The dataset underwent preprocessing, including downsampling audio files, segmenting them based on phone annotations, and organizing them into a table. A subset of the dataset was then selected and cleaned according to specific criteria. Segments were labeled as CTV or NCTV based on phonetic annotations. Lastly, the audio segments were standardized.

6.1.2 Feature engineering

Feature engineering involved extracting various acoustic features such as ZCR, RMS, STFT, Mel Spectrogram, STFT Chromagram, CQT Chromagram, CENS, Spectral Centroid, Spectral Bandwidth, Spectral Contrast, Spectral Flatness, Spectral Rolloff, Tonnetz, MFCCs, MFCCs Delta, MFCCs Delta Delta, and Polynomial features. The dataset's features were standardized. Frames of 1024 samples with a HS of 512 samples were created for each feature. To balance the class distribution, silence frames were selectively eliminated. An average-sized word was defined as the CCS. Sequences of OCS-sized feature vectors were generated, with each sequence assigned

the label of the last frame. Feature dimension and extraction time analysis were conducted. Feature selection was performed based on PCC, SCC, ANOVA F-Value, and RFI. Ultimately, a SFS composed of the 49 top-performing features was selected based on average scores from the 4 metrics.

6.1.3 Selection and design of DL-based model architectures

DL-based model architectures were selected based on their empirical success and a balanced number of trainable parameters (around 200,000). The choice of architectures aimed to ensure proven success in classification tasks involving sequential and temporal data, while maintaining a baseline complexity level.

6.1.4 Assessment and comparison of all model/features subset pairs

Models were trained and evaluated using a TTS approach. Performance and computational metrics were obtained for model evaluation. A comparison was conducted to assess the effects of different feature subsets on model performance. The SFS yielded the best results across all models. Based on the performance metrics obtained from TTS evaluation, the TCN model using the SFS was selected as the best performing model/feature subset pair, with an Accuracy of $98.72\% \pm 0.16$, Precision of $98.71\% \pm 0.22$, Recall of $98.74\% \pm 0.22$, Specificity of $98.71\% \pm 0.23$, F1 Score of $98.72\% \pm 0.15$ and AUC-ROC of $99.91\% \pm 0.01$.

6.1.5 Assessment of the best performing model/features subset

The best performing model/feature subset was evaluated using K-FCV to substantiate its effectiveness and generalizability. Performance and computational efficiency were assessed and validated. The model's VD Accuracy for different articulation manner classes was further evaluated. Next, the model was tested on tasks from the phonetically annotated WS dataset to graphically exemplify its operation. Finally, the model's compliance with the MAPT was verified and validated, ensuring online operation capability.

6.2 Research question and hypotheses

The Chapter *"Introduction"* detailed the research question and hypotheses in Section *"Research question and hypotheses"* (1.3). The research question posed was:

- **Research question:** *"What is the effectiveness and efficiency of a carefully chosen Deep Learning (DL)-based model which performs online frame-based VDs in European Portuguese (EP) Whispered Speech (WS), utilizing a Selected Feature Subset (SFS) as input?"*

The hypotheses formulated included H1, H2, and H3, positing on effectiveness, efficiency, and improvements due to SFS, respectively:

- **H1:** *A carefully chosen DL-based model **effectively** performs online frame-based VDs in EP WS*;
- **H2:** *A carefully chosen DL-based model performs online frame-based VDs in EP WS **efficiently**, taking less than the MAPT to process and decide on the input features*;
- **H3:** *A carefully chosen SFS, when used as input, **improves the effectiveness and efficiency** of a DL-based model performing online frame-based VDs in EP WS, compared to a BFS.*

Each hypothesis was subsequently addressed and validated in the following sections (6.2.1, 6.2.2, and 6.2.3), thereby contributing to answering the research question.

6.2.1 Validation of Hypothesis 1

The research findings supported this hypothesis, demonstrating effective VDs by a DL-based model in EP WS contexts.

The validation of H1 is derived from the results of the TTS and K-FCV evaluations (5.5.2, 5.6.2) which substantiated the effectiveness and generalizability of the chosen model.

6.2.2 Validation of Hypothesis 2

The validation of H2 is confirmed by the compliance with MAPT (5.6.4), demonstrating the feasibility of online operation of the TCN/SFS pair.

6.2.3 Validation of Hypothesis 3

H3 was validated through the comparison of the performances obtained using SFS and BFS:

- The comparative assessment across feature subsets (5.5.3) shows that an optimized SFS outperforms a traditional BFS consisting exclusively of MFCCs;
- Despite having similar feature space dimensions (5.3.8, 5.3.7), thus offering comparable complexity and efficiency, the SFS surpasses the BFS in effectiveness by concentrating the most relevant features for the VD task.

6.3 Contributions, innovations and implications

This research work has made significant contributions and innovations in the field of online frame-based VD for EP WS. The implications of these findings are substantial, particularly in improving communication abilities of voice patients. The contributions (6.3.1), innovations (6.3.2), and implications (6.3.3) of the research work are further detailed next.

6.3.1 Contributions

A comparative assessment of multiple models and feature subset pairs was conducted to determine the feasibility of various DL architectures for online frame-based EP WS VD, using objective performance and computational efficiency metrics. Based on the results, general guidelines were established regarding the suitability of DL architectures for VD.

Feature selection techniques were employed to define a SFS and compare it with a commonly used BFS. The SFS consistently outperformed the BFS in all models examined, providing valuable insights into the importance of specific audio features in distinguishing between CTV and NCTV phones.

The best performing model/feature subset pairing (TCN/SFS) was identified and subjected to rigorous evaluation to validate its effectiveness, generalizability, efficiency, and feasibility of online operation. This evaluation process led to the development of the final online frame-based EP WS VD subsystem, confirming all the aforementioned aspects.

When compared to current state-of-the-art methodologies, the proposed VD subsystem demonstrated superior performance metrics.

All the methods utilized during the research were thoroughly explained, ensuring reproducibility.

6.3.2 Innovations

Unlike other solutions that operate in different languages, the proposed VD subsystem was specifically designed for EP.

The VD subsystem classifies the current frame based on the preceding 16 frames of speech. This number of frames corresponds to the AWLF, so that the model's context captures the entirety of an average word from the dataset.

Feature selection was used to define a SFS from all the extracted features, providing insights on the relevance of each feature for VD. The resulting subset was compared to a BFS widely used in state-of-the-art approaches, achieving increased effectiveness.

In contrast to many surveyed state-of-the-art studies lacking objective metrics for assessing the effectiveness and efficiency of their VD approaches, this study employed two distinct evaluation techniques (TTS and K-FCV), providing a comprehensive set of performance and computational metrics.

6.3.3 Implications

The contributions and innovations of this study resulted in an efficient and effective DL-based online frame-based VD subsystem tailored for EP WS. Integration of this subsystem into a broader whispered-to-normal conversion system has the potential to significantly enhance WS reconstruction. By adopting this technology, individuals with vocal communication disabilities or impairments can improve their communication ability and overcome the limitations imposed by their

health conditions. The inclusion of this subsystem potentially empowers these individuals to surpass their communicative restrictions, enhancing both human-human and human-machine interaction and ultimately improving their quality of life.

6.4 Limitations and future work

In this Chapter, the limitations of the research work were identified, and future work was proposed to overcome them. The Chapter focused on several aspects, namely time and computational resources (6.4.1), data (6.4.2), feature engineering (6.4.3), selection and design of DL-based model architectures (6.4.4), and assessment of model/features subset pairs (6.4.5).

6.4.1 Time and computational resources

This section presents the limitations (6.4.1.1) and proposed future work (6.4.1.2) concerning time and computational resources.

6.4.1.1 Limitations

The limitations related to time and computational resources have a significant impact on the research process. Inadequate access to computational resources, specifically limited RAM and lack of a powerful GPU, poses challenges in the field of ML. The following limitations are associated with these challenges:

- **Increased time consumption:** Insufficient computational resources can lead to increased time consumption during the experimental process. The lack of computational power limits the flexibility to apply various methodologies and techniques, which can hinder the overall study efficiency. Longer execution times impede the ability to explore alternative approaches, experiment with different parameters, and conduct comprehensive analyses, ultimately affecting the breadth and depth of the research;

- **Restrictions on utilizing computationally expensive methodologies:** Limited computational resources restrict the ability to employ computationally expensive methodologies. Some advanced techniques, such as complex DL architectures and feature selection algorithms, often require substantial computational power during execution. The lack of sufficient resources limits the adoption of these methodologies, which may offer valuable insights and improved performance in addressing research objectives.

6.4.1.2 Future work

To address the limitations related to time and computational resources, future work should focus on the following aspects:

- **Acquiring improved computational resources:** Efforts should be made to secure access to more powerful computational resources, such as high-performance CPUs, GPUs, and larger RAM. This would enable the application of more complex methodologies and techniques, facilitating comprehensive analyses and reducing time consumption;

- **Exploring cloud-based solutions:** The utilization of cloud-based computational resources should be considered. Cloud platforms offer scalable and on-demand resources, which can be particularly beneficial when local computational resources are limited.

By addressing these limitations, future research can overcome the constraints imposed by time and computational resources, enabling more efficient, extensive and impactful investigations.

6.4.2 Data

This section presents the limitations (6.4.2.1) and proposed future work (6.4.2.2) concerning the data utilized in this research.

6.4.2.1 Limitations

In a DL problem, the availability and quality of the data are fundamental premises for the success of the subsequent analysis. The limitations identified in this research work regarding the utilization of data were stated as follows:

- **Size and phonetic class balance of the dataset:** The limitations related to the size and phonetic class balance of the dataset used in this research work have significant implications for the success of DL models:

  - **Limited representation of the real-world problem:** A small dataset may fail to capture the full diversity and complexity of the real-world problem it aims to solve. Additionally, class imbalance, where certain phonetic classes are underrepresented compared to others, can impact the model's ability to learn and generalize across all classes. Inadequate representation in the dataset may lead to biased or incomplete learning, limiting the model's performance;

  - **Increased risk of overfitting:** With a small dataset, DL models are more prone to overfitting. Overfitting occurs when the model memorizes the training examples instead of learning generalizable patterns. This phenomenon can severely impact the model's ability to generalize to unseen data, rendering it ineffective for practical use. The risk of overfitting is amplified in small datasets where the model's capacity to learn diverse patterns and variations is limited;

- **Diversity of datasets:** This research work relied on a proprietary in-house dataset of phonetically annotated WS in EP, which introduced several limitations:

- **Limited reproducibility:** By using a proprietary dataset, the reproducibility and universal acceptance of the findings may be compromised. Reproducing the results and validating the proposed methods on independent datasets becomes challenging without access to the same dataset used in the study. The lack of diverse datasets from different sources limits the ability to generalize the findings and evaluate the robustness of the proposed approaches;
- **Limited comparability:** The limited availability of diverse datasets hampers the rigorous comparison of the effectiveness and performance of the proposed methods against other state-of-the-art approaches. Without benchmark datasets that are commonly used in the research community, it becomes difficult to establish a fair and comprehensive comparison. The absence of such comparisons may limit the insights gained from the study and hinder the understanding of the proposed methods' relative strengths and weaknesses.

6.4.2.2 Future work

To mitigate the limitations related to the dataset, future work can focus on increasing the size and reducing/eliminating the phonetic class imbalance of the dataset:

- **Dataset expansion:** Efforts should be made to expand the dataset by collecting more data or applying data augmentation techniques. Increasing the dataset size would provide a more comprehensive representation of the underlying phonetic classes and improve the generalization capability of the models;
- **Addressing class imbalance:** Techniques such as oversampling, undersampling and class-weighting can be applied to balance the representation of different phonetic classes in the dataset. This would alleviate the bias introduced by class imbalance and improve the models' ability to learn and generalize across all classes;
- **Enhancing reproducibility and comparability:** To enhance the reproducibility, universality, and comparability of the research findings, future work should consider the application of the proposed methods to speech datasets in other languages, particularly those that share a significant number of phonemes with EP. By evaluating the methods on diverse datasets, researchers can gain insights into the models' performance across different linguistic contexts and broaden the applicability of the research outcomes.

By addressing the reduced size and phonetic class imbalance, and exploring diverse datasets, future research can improve the reliability and performance of the models trained on the dataset.

6.4.3 Feature engineering

This section presents the limitations (6.4.3.1) and the proposed future work (6.4.3.2) concerning the feature engineering process conducted in this research.

6.4.3.1 Limitations

Despite the considerable accomplishments in this work, certain limitations related to the feature engineering process should be acknowledged, primarily due to time and computational resource constraints. These limitations include:

- **Exploratory Data Analysis (EDA) limitations:** Due to time and resource constraints, conducting a comprehensive EDA was challenging. EDA is an essential step in understanding data distribution, identifying patterns, and gaining insights that can inform the feature engineering process. Although the available EDA provided valuable insights within the given constraints, additional time and resources would have allowed for a more thorough exploration of the data;
- **Limited feature selection:** The feature selection process was performed to a certain extent, considering the available resources and time limitations. Feature selection techniques are essential for identifying relevant and discriminative features for the task at hand. Although the selected features hold value within the available resources and time limitations, it is important to acknowledge that incorporating more extensive feature selection methods could potentially yield further improvements in the model's performance and generalization ability;
- **Optimization of Overlapping Context Size (OCS):** Given the time and resource constraints, a comprehensive exploration of OCS optimization was challenging. The OCS determines the size of the overlapping context window used for feature extraction, and different values can affect the model's ability to capture temporal dependencies and contextual information. Although the chosen OCS enabled high effectiveness and efficiency, a more exhaustive investigation would have provided additional insights into the optimal configuration.

6.4.3.2 Future work

To enhance the feature engineering process and improve the effectiveness of DL-based VD subsystems, future work should consider the following:

- **Thorough EDA:** Conducting a more comprehensive EDA, including univariate and multivariate analyses, can provide deeper insights into the data distribution, identify patterns, and uncover potential relationships among variables. This would facilitate better feature extraction and selection, leading to improved system performance;
- **Advanced feature selection techniques:** Employing a wider range of feature selection techniques, including filter, wrapper, and embedded methods, can help identify the most relevant and discriminative features for the task at hand. Comprehensive feature selection would ensure that the models focus on the most informative aspects of the input data and improve the models' performance and generalization capability;

- **Optimization of context size:** Conducting systematic investigations to optimize the context size can enhance the models' ability to capture temporal dependencies and contextual information. Exploring different context sizes and evaluating their impact on the model's performance can lead to more accurate and robust VD systems.

By addressing these aspects of feature engineering, future research can enhance the quality of feature representation and improve the overall performance of DL-based VD systems.

6.4.4 Selection and design of DL-based model architectures

This section presents the limitations (6.4.4.1) and the proposed future work (6.4.4.2) concerning the process of selection and design of DL-based model architectures conducted in this research.

6.4.4.1 Limitations

Despite the substantial progress achieved in this study, it is important to consider the following limitations imposed by time and computational constraints during the selection and design of DL-based model architectures:

- **Limited model architectures analyzed:** The study involved analyzing a focused set of model architectures. While these choices allowed for in-depth exploration, it is possible that alternative architectures not included in this analysis could have provided additional valuable insights and potentially effectiveness/efficiency for the given task. A broader analysis encompassing a larger pool of model architectures could have further enriched the evaluation;

- **Relatively low number of parameters:** In order to ensure experimental efficiency and flexibility within the given constraints, a decision was made to keep the number of parameters in the DL models relatively low. This approach allowed for efficient utilization of available resources. However, it is worth considering that a higher number of parameters might have unlocked the potential for capturing more intricate data representations and potentially achieving even better performance;

- **Limited hyperparameter optimization:** Due to the limitations in time and computational resources, the optimization of hyperparameters, including learning rate, batch size, and regularization techniques, was carried out to a lesser extent. While the selected hyperparameters were able to provide meaningful results, a more extensive exploration and fine-tuning of these parameters could have further enhanced the models' performance and generalization capabilities.

6.4.4.2 Future work

To further enhance the selection and design of DL-based model architectures, future work should consider the following:

- **Complex DL models:** Implementing more complex DL models with increased model capacity, such as deeper or wider architectures, can enhance the models' ability to capture complex data representations. These architectures can learn more intricate patterns and improve the performance of voice decision systems;
- **Automatic ML techniques:** Utilizing Auto ML techniques can streamline the model selection and design process, as well as hyperparameter optimization. Automated approaches, such as genetic algorithms or Bayesian optimization, can efficiently explore the model space and identify optimal architectures and hyperparameters. This would potentially lead to more effective and efficient VDs.

By incorporating these strategies into the selection and design process of DL-based model architectures, future research can leverage the potential of advanced architectures and optimize their performance.

6.4.5 Assessment of model/features subset pairs

This section presents the limitations (6.4.5.1) and the proposed future work (6.4.5.2) concerning the process of assessing model/features subset pairs conducted in this research.

6.4.5.1 Limitations

The assessment of the efficiency and complexity of different model/features subset pairs in this study was extensive, demonstrating significant progress. However, certain limitations should be acknowledged:

- **Absence of the Floating-point Operations per Second (FLOPS) metric:** Although the analysis of models' efficiency and complexity was thorough, it lacked an evaluation using the number of FLOPS. Incorporating this metric would provide valuable insights into the computational requirements of the models and enable a detailed comparison of inference times across different deployment hardware;
- **No integration with a whispered-to-normal speech conversion system:** Although the assessment of VD subsystem was comprehensive, integration with the broader whispered-to-normal voice conversion system was not implemented, since it was not a goal of this study. This integration could further substantiate the effectiveness and efficiency of the VD subsystem. Furthermore, it would allow conducting subjective evaluations using the MOS to assess the perceptual quality of the broader system.

6.4.5.2 Future work

To address these limitations and provide a more comprehensive understanding of the efficiency, complexity, and overall performance of the proposed model/features subset pairs, the following future work is proposed:

- **Evaluation using FLOPS:** Conducting an evaluation that incorporates the FLOPS metric will provide valuable insights into the computational requirements of the models and enable meaningful comparisons of inference times across different deployment hardware;
- **Integration with a whispered-to-normal speech conversion system:** Integrating the VD subsystem with the whispered-to-normal voice conversion system will allow for objective and subjective evaluations of the entire system. This integration will provide further validation of the effectiveness and efficiency of the VD subsystem and enable perceptual quality assessments using the MOS.

By addressing these areas of assessment, future research can gain a deeper understanding of the computational efficiency, performance, and perceptual quality of the DL-based VD subsystem within the broader whispered-to-normal speech conversion system.

6.5 Chapter summary

This Chapter *"Conclusions"* presents the final outcomes of the research conducted in the context of this dissertation. This Chapter encompasses several important aspects, including a summary of the key findings (6.1), the answer to the research questions through validation of the hypotheses (6.2), the contributions, innovations, and implications of the research work (6.3), as well as the limitations encountered and proposed avenues for future work to address them (6.4).

In summary, this concluding chapter provides a comprehensive overview of the research, tying together the main elements of the study and offering insights into its significance and potential future directions.

Appendix A

Appendix

1

Table A.1: Summary of the literature review on voicing decision approaches.

Title	Reference	Year	Description	Classifier	Features	Training data	Evaluation	Advantages	Disadvantages
Voicing decision based on phonemes classification and spectral moments for whisper-to-speech conversion	[33]	2022	First, classification of whispered signal frames into phoneme classes based on their spectral centroid and spread is conducted. Then, discrimination between voiced phonemes and their unvoiced counterpart based on class-dependent spectral centroid thresholds is performed to estimate the voicing decision.	ML based (KNN) + Rule based	KNN: Spectral centroid, spectral spread. Rule based: Spectral centroid.	In-house database of annotated whispered speech. 10 different speakers (5 female and 5 male). For each one: 114 sequences of 3 phonemes, 19 steady phonemes, and 63 words.	Objective evaluation. KNN $+Rulebased:$ Accuracy $\approx 91\%$ Baseline (single global threshold on the raw spectral centroid of a signal frame): Accuracy $\approx 91\%$	Individual system calibration is avoided by training the algorithm on a pre-annotated multispeaker database of read text. Reduces systematic voicing errors for some phonemes, opening the path to a more suitable control space for voicing decision. Low-resource approach, suitable for real-time applications.	Second step of the approach is rule based. It may compromise robustness and generalization capability of classifier.
Glottal flow synthesis for whisper-to-speech conversion	[49]	2020	Spectral centroid thresholds are used to estimate voicing decision.	Rule-based	Spectral centroid.	-	Objective evaluation. Voiced error $= 7.6\%$ Unvoiced error $= 10.1\% Totalerror = 17.7\%$.	Low complexity approach, suitable for real-time applications.	Approach is rule based. It may compromise robustness and generalization capability of classifier. High error rate.
Whispered speech to neutral speech conversion using bidirectional LSTMs	[29]	2018	A BLSTM model is employed to predict the voicing decision.	ML based (BLSTM)	MFCCs, velocity and acceleration computed from the smooth spectrum of whispered speech. Excitation parameter obtained from STRAIGHT analysis of neutral speech.	Parallel training data of whispered and neutral speech. 60 sentences taken from the MOCHA-TIMIT database from six subjects, three males and three females. The subjects were asked to speak each sentence in neutral and whispered modes separately.	Quantitative evaluation. Total error $\approx 8\%$	Low error rate. Generalization capability,	High complexity. Requires large training databases.

Continued on next page

Table A.1: Summary of the literature review on voicing decision approaches. (Continued)

Title	Reference	Year	Description	Classifier	Features	Training data	Evaluation	Advantages	Disadvantages
A robust voiced/unvoiced phoneme classification from whispered speech using the "color" of whispered phonemes and deep neural network	[30]	2017	A DNN is used to estimate the voicing decision. A 5D engineered feature is considered, based on the decomposition of the whispered speech spectrum in a linear combination of a set of colored noise spectra.	ML based (DNN)	MFCC features computed from a dictionary, constructed using spectra of five colored noises.	In-house annotated whispered speech database. 4 female and 3 male speakers. Each of the seven speakers whispered about 450 phonetically balanced sentences from the MOCHA-TIMIT.	Objective evaluation. MFCC-DNN Voiced accuracy = 73.63% Unvoiced accuracy = 78.51% Average accuracy = 76.06% Combined-DNN (5D+MFCC) Voiced accuracy = 73.81% Unvoiced accuracy = 74.78% Average accuracy = 74.29%	Balanced frame-level V/UV classification accuracy using the Combined-DNN scheme.	High complexity. Low accuracy.
Implantation of voicing on whispered speech using frequency-domain parametric modelling of source and filter information	[50]	2016	Temporal and frequency band energy thresholds are used to estimate the voicing decision.	Rule based	Temporal and frequency-band energy variations.	-	-	Low complexity approach, suitable for real-time applications.	Approach is rule based. It may compromise robustness and generalization capability of classifier.
Whisperto-speech conversion using restricted boltzmann machine arrays	[31]	2014	GMM and SVM models are used to obtain a voicing decision, trained using the mel-cepstra static and dynamic features of whispered speech.	ML based (GMM, SVM)	Mel-cespstra static and dynamic features.	Aproximately 180000 frames of parallel whisper and speech recordings from wTIMIT database. Male and female.	Objective evaluation. GMM $+-$ 5 frames Voiced error = 5.09% Unvoiced error = 3.77% Total error = 8.86% SVM $+-5 frames$ Voiced error = 4.39% Unvoiced error = 5.08% Total error = 9.47%	Low error rate. Generalization capability,	High complexity. Requires large training databases.
Improvement to a nam-captured whisper-to-speech system	[32]	2010	FNN trained with MFFCs feature is used to predict the segments from whispered speech. A threshold is used to convert the continuous output to a voicing decision.	ML based (FNN)	MFCCs	200 utterance pairs of whisper and normal speech, verbalized by a French native male speaker.	Objective evaluation. GMM Voiced error = 3.3% Unvoiced error = 5.9% Total error 9.2% FNN Voiced error 2.4% Unvoiced error = 4.4% Total error = 6.8%	Low error rate. Generalization capability,	High complexity. Requires large training databases.

Continued on next page

Table A.1: Summary of the literature review on voicing decision approaches. (Continued)

Title	Reference	Year	Description	Classifier	Features	Training data	Evaluation	Advantages	Disadvantages
Reconstruction of normal sounding speech for laryngectomy patients through a modified celp codec	[51]	2010	Temporal and frequency band energy thresholds are used to estimate the voicing decision.	Rule based	Temporal and frequency-band energy variations.	-	-	Low complexity approach, suitable for real-time applications.	Approach is rule based. It may compromise robustness and generalization capability of classifier.

Table A.2: Feature selection scores.

Feature	**Pearson**	**Spearman**	**ANOVA F-value**	**Random Forest Importance**	**Average Score**
poly_features_1	92.67	100	85.31	100	94.49
poly_features_2	100	99.26	100	74.07	93.33
tonnetz_2	95.31	93.03	90.48	85.9	91.18
tonnetz_6	91.57	90.12	83.25	72.62	84.39
mfcc_1	92.52	88.82	85.06	69.87	84.07
rms_1	79.97	88.4	62.9	83.31	78.65
tonnetz_4	81.75	78.73	65.82	78.02	76.08
spectral_bandwidth_1	60.07	56.98	35.03	55.4	51.87
spectral_rolloff_1	63.82	59.43	39.65	41.49	51.1
spectral_contrast_7	69.08	64.46	46.58	13.33	48.36
mfcc_6	67.27	57.09	44.14	17.25	46.43
spectral_flatness_1	50.61	61.95	24.76	31.08	42.1
mfcc_2	47.02	45.77	21.36	43.34	39.37
spectral_centroid_1	47.69	51.62	21.97	32.01	38.32
spectral_contrast_5	51.99	52.61	26.13	12.77	35.88
spectral_contrast_6	50.58	51.86	24.74	16.21	35.85
mfcc_3	42.78	40.03	17.63	32.99	33.36
mfcc_5	51.5	43.91	25.64	11.05	33.02
spectral_contrast_4	51.7	46.66	25.84	1.57	31.44
mfcc_13	47.13	44	21.46	7.51	30.03
chroma_stft_4	38.47	37.36	14.24	12.65	25.68
chroma_stft_3	39.12	37.67	14.72	7.32	24.71
mfcc_delta_1	28.33	37.21	7.69	19.9	23.28
chroma_stft_2	35.08	33.54	11.84	10.18	22.66

Continued on next page

Table A.2: Feature selection scores. (Continued)

tonnetz_3	38.32	29.94	14.14	7.94	22.58
chroma_stft_10	35.89	34.08	12.38	6.27	22.15
chroma_stft_1	33.98	32.69	11.13	9.82	21.91
chroma_stft_5	32.37	31.44	10.06	7.18	20.26
chroma_stft_12	31.68	28.84	9.69	8.13	19.58
mfcc_10	33.95	28.42	11.07	4.23	19.42
zero_crossing_rate_1	24.28	25.25	5.68	21.84	19.26
chroma_stft_11	32.23	28.62	10	4.38	18.81
chroma_stft_9	28.91	27.69	8.02	3.54	17.04
mfcc_17	27.32	25.84	7.16	1.67	15.5
mfcc_4	19.19	21.52	3.56	16.59	15.21
mfcc_12	23.85	24.01	5.46	4.36	14.42
spectral_contrast_3	24.69	24.38	5.84	0.56	13.87
chroma_stft_6	23.1	22.71	5.11	3.06	13.5
tonnetz_5	21.56	19.58	4.46	6.8	13.1
tonnetz_1	22.65	20.67	4.93	3.57	12.96
chroma_stft_8	22.61	21.86	4.9	2.27	12.91
mfcc_7	19.95	17.3	3.84	7.48	12.14
mfcc_9	18.85	19.08	3.45	6.6	12
mfcc_27	22.33	19.66	4.79	0.95	11.93
mfcc_8	20.78	18.79	4.14	3.45	11.79
mfcc_23	20.6	19.37	4.07	1.03	11.27
chroma_stft_7	19.17	18.59	3.52	2.65	10.98
mfcc_delta_3	14.54	14.32	2.02	7.11	9.5
mfcc_25	15.61	14.52	2.34	0.63	8.28

References

[1] International Phonetic Association. *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet*. Cambridge University Press, June 1999.

[2] Ian Vince McLoughlin. *Speech and Audio Processing: A MATLAB®-Based Approach*. Cambridge University Press, Cambridge, 2016. doi:10.1017/CBO9781316084205.

[3] Francesc Alías, Joan Claudi Socoró, and Xavier Sevillano. A Review of Physical and Perceptual Feature Extraction Techniques for Speech, Music and Environmental Sounds. *Applied Sciences*, 6(5):143, May 2016. doi:10.3390/app6050143.

[4] Luis M. T. Jesus, Sara Castilho, Aníbal Ferreira, and Maria Conceição Costa. Discriminative segmental cues to vowel height and consonantal place and voicing in whispered speech. *Journal of Phonetics*, 97:101223, March 2023. doi:10.1016/j.wocn.2023.101223.

[5] W Tecumseh Fitch. The evolution of speech: a comparative review. *Trends in cognitive sciences*, 4(7):258–267, 2000.

[6] Herbert H Clark. Using language. 1996.

[7] Martin J Pickering and Simon Garrod. *Understanding dialogue: Language use and social interaction*. Cambridge University Press, 2021.

[8] Rohan Kumar Das and Haizhou Li. On the importance of vocal tract constriction for speaker characterization: The whispered speech study. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7119–7123. IEEE, 2020.

[9] João P Silva, Clara F Cardoso, Marco A Oliveira, Luís MT Jesus, and Aníbal JS Ferreira. A comparative study of european portuguese stop consonants and fricatives in whispered and normal speech for real-time operation of voice conversion. *PROCEEDINGS E REPORT*, page 53, 2021.

[10] João Silva, Marco Oliveira, and Aníbal Ferreira. Flexible parametric implantation of voicing in whispered speech under scarce training data. In *2020 28th European Signal Processing Conference (EUSIPCO)*, pages 416–420, 2021. doi:10.23919/Eusipco47968.2020.9287684.

[11] Marco A Oliveira. Machine learning approaches for whisper to normal speech conversion: A survey. *U. Porto Journal of Engineering*, 8(2):202–212, 2022.

[12] Zealear David Li Yike, Garrett Gaelyn. Current treatment options for bilateral vocal fold paralysis: A state-of-the-art review. *Clin Exp Otorhinolaryngol*, 10(3):203–212, 2017. doi:10.21053/ceo.2017.00199.

[13] Oleksandr Butskiy, Bhavik Mistry, and Neil K Chadha. Surgical interventions for pediatric unilateral vocal cord paralysis: a systematic review. *JAMA Otolaryngology–Head & Neck Surgery*, 141(7):654–660, 2015.

[14] FRANS J. M. HILGERS, ANNEMIEKE H. ACKERSTAFF, NEIL K. AARONSON, PAUL F. SCHOUWENBURG, and NICO VAN ZANDWIJK. Physical and psychosocial consequences of total laryngectomy. *Clinical Otolaryngology & Allied Sciences*, 15(5):421–425, 1990. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2273.1990.tb00494.x, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1365-2273.1990.tb00494.x, doi:https://doi.org/10.1111/j.1365-2273.1990.tb00494.x.

[15] Merete Salveson Engeseth, Nina Rydland Olsen, Silje Maeland, Thomas Halvorsen, Adam Goode, and Ola Drange Røksund. Left vocal cord paralysis after patent ductus arteriosus ligation: a systematic review. *Paediatric Respiratory Reviews*, 27:74–85, 2018.

[16] Gabriella Sharpe, Vera Camoes Costa, Wendy Doubé, Jodi Sita, Chris McCarthy, and Paul Carding. Communication changes with laryngectomy and impact on quality of life: a review. *Quality of Life Research*, 28:863–877, 2019.

[17] Aníbal Ferreira. Implantation of voicing on whispered speech using frequency-domain parametric modelling of source and filter information. In *2016 International Symposium on Signal, Image, Video and Communications (ISIVC)*, pages 159–166, 2016. doi:10.1109/ISIVC.2016.7893980.

[18] Edwin M-L Yiu, Katherine Verdolini Abbott, and Estella P-M Ma. Application of the icf in voice disorders. In *Seminars in Speech and Language*, volume 28, pages 343–350. © Thieme Medical Publishers, 2007.

[19] Jiří Mertl, Eva Žáčková, and Barbora Řepová. Quality of life of patients after total laryngectomy: the struggle against stigmatization and social exclusion using speech synthesis. *Disability and Rehabilitation: Assistive Technology*, 13(4):342–352, 2018.

[20] Luis MT Jesus, Sara Castilho, Aníbal Ferreira, and Maria Conceição Costa. Discriminative segmental cues to vowel height and consonantal place and voicing in whispered speech. *Journal of Phonetics*, 97:101223, 2023.

[21] Gokul Srinivasan, Aravind Illa, and Prasanta Kumar Ghosh. A study on robustness of articulatory features for automatic speech recognition of neutral and whispered speech. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5936–5940. IEEE, 2019.

[22] Prithvi RR Gudepu, Gowtham P Vadisetti, Abhishek Niranjan, Kinnera Saranu, Raghava Sarma, M Ali Basha Shaik, and Periyasamy Paramasivam. Whisper augmented end-to-end/hybrid speech recognition system-cyclegan approach. In *INTERSPEECH*, pages 2302–2306, 2020.

[23] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*, 2022.

[24] Teng Gao, Qing Pan, Jian Zhou, Huabin Wang, Liang Tao, and Hon Keung Kwan. A novel attention-guided generative adversarial network for whisper-to-normal speech conversion. *Cognitive Computation*, pages 1–15, 2023.

[25] Dominik Wagner, Sebastian P Bayerl, Héctor A Cordourier Maruri, and Tobias Bocklet. Generative models for improved naturalness, intelligibility, and voicing of whispered speech. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 943–948. IEEE, 2023.

[26] Kishor Barasu Bhangale and Mohanaprasad Kothandaraman. Survey of deep learning paradigms for speech processing. *Wireless Personal Communications*, 125(2):1913–1949, 2022.

[27] Santiago Pascual De La Puente. Efficient, end-to-end and self-supervised methods for speech processing and generation. 2020.

[28] Shogo Seki, Hirokazu Kameoka, Takuhiro Kaneko, and Kou Tanaka. Non-parallel whisper-to-normal speaking style conversion using auxiliary classifier variational autoencoder. *IEEE Access*, 2023.

[29] G. Nisha Meenakshi and Prasanta Kumar Ghosh. Whispered Speech to Neutral Speech Conversion Using Bidirectional LSTMs. In *Interspeech 2018*, pages 491–495. ISCA, September 2018. doi:10.21437/Interspeech.2018-1487.

[30] G. Nisha Meenakshi and Prasanta Kumar Ghosh. A Robust Voiced/Unvoiced Phoneme Classification from Whispered Speech Using the 'Color' of Whispered Phonemes and Deep Neural Network. In *Interspeech 2017*, pages 503–507. ISCA, August 2017. doi:10.21437/Interspeech.2017-1388.

[31] Jing-jie Li, Ian Mcloughlin, Li-Rong Dai, and Zhen-Hua Ling. Whisper-to-speech conversion using restricted Boltzmann machine arrays. *Electronics Letters*, 50:1781–1782, November 2014. doi:10.1049/el.2014.1645.

[32] Viet-Anh Tran, Gérard Bailly, Hélène Lœvenbruck, and Tomoki Toda. Improvement to a NAM-captured whisper-to-speech system. *Speech Communication*, 52(4):314–326, April 2010. doi:10.1016/j.specom.2009.11.005.

[33] Luc Ardaillon, Nathalie Henrich Bernardoni, and Olivier Perrotin. Voicing decision based on phonemes classification and spectral moments for whisper-to-speech conversion. In *Interspeech 2022*, Proceedings of Interspeech, pages 2253–2257, Incheon, South Korea, September 2022. ISCA. doi:10.21437/interspeech.2022-10675.

[34] Aníbal Ferreira and Fernando Pereira. *Comunicações audiovisuais: tecnologias, normas e aplicações*. Number 26 in Ensino da Ciência e da Tecnologia. IST Press, Lisboa,Portugal, 2009.

[35] Marco António da Mota Oliveira. Modelização de filtro de trato vocal para reconstrução de voz disfónica. February 2020.

[36] S. S. Stevens, J. Volkmann, and E. B. Newman. A Scale for the Measurement of the Psychological Magnitude Pitch. *The Journal of the Acoustical Society of America*, 8(3):185–190, January 1937. doi:10.1121/1.1915893.

[37] S. Wang, A. Sekey, and A. Gersho. An objective measure for predicting subjective quality of speech coders. *IEEE Journal on Selected Areas in Communications*, 10(5):819–829, June 1992. doi:10.1109/49.138987.

[38] Gunnar Fant. *Acoustic Theory of Speech Production: With Calculations Based on X-Ray Studies of Russian Articulations*. Walter de Gruyter, 1970.

[39] D. O'Shaughnessy. Linear predictive coding. *IEEE Potentials*, 7(1):29–32, February 1988. doi:10.1109/45.1890.

[40] M. W. Spratling. A review of predictive coding algorithms. *Brain and Cognition*, 112:92–97, March 2017. doi:10.1016/j.bandc.2015.11.003.

[41] Brian McFee, Colin Raffel, Dawen Liang, Daniel Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. Librosa: Audio and Music Signal Analysis in Python. In *Python in Science Conference*, pages 18–24, Austin, Texas, 2015. doi:10.25080/Majora-7b98e3ed-003.

[42] Brian McFee, Matt McVicar, Daniel Faronbi, Iran Roman, Matan Gover, Stefan Balke, Scott Seyfarth, Ayoub Malek, Colin Raffel, Vincent Lostanlen, Benjamin van Niekirk, Dana Lee, Frank Cwitkowitz, Frank Zalkow, Oriol Nieto, Dan Ellis, Jack Mason, Kyungyun Lee, Bea Steers, Emily Halvachs, Carl Thomé, Fabian Robert-Stöter, Rachel Bittner, Ziyao Wei, Adam Weiss, Eric Battenberg, Keunwoo Choi, Ryuichi Yamamoto, CJ Carr, Alex Metsai, Stefan Sullivan, Pius Friesch, Asmitha Krishnakumar, Shunsuke Hidaka, Steve Kowalik, Fabian Keller, Dan Mazur, Alexandre Chabot-Leclerc, Curtis Hawthorne, Chandrashekhar Ramaprasad, Myungchul Keum, Juanita Gomez, Will Monroe, Viktor Andreevitch Morozov, Kian Eliasi, nullmightybofo, Paul Biberstein, N. Dorukhan Sergin, Romain Hennequin, Rimvydas Naktinis, beantowel, Taewoon Kim, Jon Petter Åsen, Joon Lim, Alex Malins, Darío Hereñú, Stef van der Struijk, Lorenz Nickel, Jackie Wu, Zhen Wang, Tim Gates, Matt Vollrath, Andy Sarroff, Xiao-Ming, Alastair Porter, Seth Kranzler, Voodoohop, Mattia Di Gangi, Helmi Jinoz, Connor Guerrero, Abduttayyeb Mazhar, toddrme2178, Zvi Baratz, Anton Kostin, Xinlu Zhuang, Cash TingHin Lo, Pavel Campr, Eric Semeniuc, Monsij Biswal, Shayenne Moura, Paul Brossier, Hojin Lee, and Waldir Pimenta. librosa/librosa: 0.10.0.post2, March 2023. URL: https://doi.org/10.5281/zenodo.7746972, doi:10.5281/zenodo.7746972.

[43] Toshihiko Abe, Takao Kobayashi, and Satoshi Imai. Harmonics tracking and pitch extraction based on instantaneous frequency. In *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference On*, volume 1, pages 756–759 vol.1, June 1995. doi:10.1109/ICASSP.1995.479804.

[44] Malcolm Slaney. Technical Report #1998-010 Interval Research Corproation malcolm@interval.com.

[45] Christian Schörkhuber and Anssi Klapuri. CONSTANT-Q TRANSFORM TOOLBOX FOR MUSIC PROCESSING.

[46] Anssi Klapuri and Manuel Davy. *Signal Processing Methods for Music Transcription*. January 2006. doi:10.1007/0-387-32845-9.

[47] Dan-Ning Jiang, Lie Lu, Hong-Jiang Zhang, Jian-Hua Tao, and Lian-Hong Cai. Music type classification by spectral contrast feature. In *Proceedings. IEEE International Conference on Multimedia and Expo*, pages 113–116, Lausanne, Switzerland, 2002. IEEE. doi:10.1109/ICME.2002.1035731.

[48] Christopher Harte, Mark Sandler, and Martin Gasser. Detecting harmonic change in musical audio. In *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia*, AMCMM '06, pages 21–26, New York, NY, USA, October 2006. Association for Computing Machinery. doi:10.1145/1178723.1178727.

[49] Olivier Perrotin and Ian V. McLoughlin. Glottal Flow Synthesis for Whisper-to-Speech Conversion. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:889–900, 2020. doi:10.1109/TASLP.2020.2971417.

[50] Aníbal Ferreira. Implantation of voicing on whispered speech using frequency-domain parametric modelling of source and filter information. In *2016 International Symposium on Signal, Image, Video and Communications (ISIVC)*, pages 159–166, November 2016. doi:10.1109/ISIVC.2016.7893980.

[51] Hamid Sharifzadeh, Ian Mcloughlin, and Farzane Ahmadi. Reconstruction of Normal Sounding Speech for Laryngectomy Patients Through a Modified CELP Codec. *IEEE transactions on bio-medical engineering*, 57:2448–58, October 2010. doi:10.1109/TBME.2010.2053369.

[52] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.

[53] The MathWorks Inc. Matlab version: 9.13.0 (r2022b), 2022. URL: https://www.mathworks.com.

[54] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. URL: https://www.tensorflow.org/.

[55] Plotly Technologies Inc. Collaborative data science, 2015. URL: https://plot.ly.

[56] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi:10.1038/s41592-019-0686-2.

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

[58] François Chollet et al. Keras. https://keras.io, 2015.

[59] Philippe Remy. Temporal convolutional networks for keras. https://github.com/philipperemy/keras-tcn, 2020.

[60] The pandas development team. pandas-dev/pandas: Pandas, February 2020. URL: https://doi.org/10.5281/zenodo.3509134, doi:10.5281/zenodo.3509134.

[61] Daniel Pape and Luis MT Jesus. Stop and Fricative Devoicing in European Portuguese, Italian and German. *Language and Speech*, 58(2):224–246, June 2015. doi:10.1177/0023830914530604.

[62] Ferreira M., Luis Jesus, Pedro Sá-Couto, and Helena Vilarinho. University of Aveiro's Standardised Voice Case History Form. May 2014.

[63] Luísa Segura. Variedades dialetais do português europeu. *Gramática do português*, 1:85–142, 2013.

[64] Ramya Konnai, Ronald C. Scherer, Amy Peplinski, and Kenneth Ryan. Whisper and Phonation: Aerodynamic Comparisons Across Adduction and Loudness. *Journal of Voice*, 31(6):773.e11–773.e20, November 2017. doi:10.1016/j.jvoice.2017.02.016.

[65] Luís Jesus, Ana Inês Tavares, and Andreia Hall. Cross-Cultural Adaption of the GRBAS and CAPE-V Scales for Portugal and a New Training Programme for Perceptual Voice Evaluation. *Advances in Speech-language Pathology*, (13):29–255, September 2017. doi:10.5772/intechopen.69643.

[66] Luis M. T. Jesus, Ana Rita S. Valente, and Andreia Hall. Is the Portuguese version of the passage 'The North Wind and the Sun' phonetically balanced? *Journal of the International Phonetic Association*, 45(1):1–11, April 2015. doi:10.1017/S0025100314000255.

[67] Paola Escudero, Paul Boersma, Andréia Schurt Rauber, and Ricardo A. H. Bion. A cross-dialect acoustic description of vowels: Brazilian and European Portuguese. *The Journal of the Acoustical Society of America*, 126(3):1379–1393, September 2009. doi:10.1121/1.3180321.

[68] Marina Vigário, Maria João Freitas, and Sónia Frota. Grammar and Frequency Effects in the Acquisition of Prosodic Words in European Portuguese. *Language and Speech*, 49(2):175–203, June 2006. doi:10.1177/00238309060490020301.

[69] Luis M. T. Jesus, Sara Castilho, Marta Alves, and Andreia Hall. An Open Access Standardised Voice Evaluation Protocol. *Journal of Voice*, 0(0), October 2021. doi:10.1016/j.jvoice.2021.09.010.

[70] Adam D. Rubin, Veeraphol Praneetvatakul, Shirley Gherson, Cheryl A. Moyer, and Robert T. Sataloff. Laryngeal Hyperfunction During Whispering: Reality or Myth? *Journal of Voice*, 20(1):121–127, March 2006. doi:10.1016/j.jvoice.2004.10.007.

[71] Marzena Żygis, Daniel Pape, Laura L. Koenig, Marek Jaskuła, and Luis M. T. Jesus. Segmental cues to intonation of statements and polar questions in whispered, semi-whispered and normal speech modes. *Journal of Phonetics*, 63:53–74, July 2017. doi:10.1016/j.wocn.2017.04.001.

[72] Marisa Lousada, Luis M. T. Jesus, and Andreia Hall. Temporal acoustic correlates of the voicing contrast in European Portuguese stops. *Journal of the International Phonetic Association*, 40(3):261–275, December 2010. doi:10.1017/S0025100310000186.

[73] Slobodan T. Jovičić and Zoran Šarić. Acoustic Analysis of Consonants in Whispered Speech. *Journal of Voice*, 22(3):263–274, May 2008. doi:10.1016/j.jvoice.2006.08.012.

[74] Yohann Meynadier and Yulia Gaydina. Aerodynamic and durational cues of phonological voicing in whisper. In *Interspeech*, page 335, August 2013.

[75] Willemijn F. L. Heeren. Coding pitch differences in voiceless fricatives: Whispered relative to normal speech. *The Journal of the Acoustical Society of America*, 138(6):3427–3438, December 2015. doi:10.1121/1.4936859.

[76] W. F. L. Heeren and V. J. van Heuven. The interaction of lexical and phrasal prosody in whispered speech. *The Journal of the Acoustical Society of America*, 136(6):3272–3289, December 2014. doi:10.1121/1.4901705.