UNIVERSIDADE FEDERAL FLUMINENSE

LEONARDO MANHÃES GOMES

An Exploratory Analysis of CNN's Robustness to Speaker Identification in Multi-Language Scenarios

NITERÓI

2022

UNIVERSIDADE FEDERAL FLUMINENSE

LEONARDO MANHÃES GOMES

An Exploratory Analysis of CNN's Robustness to Speaker Identification in Multi-Language Scenarios

Dissertation presented to the Computing Graduate Program of the Universidade Federal Fluminense in partial fulfillment of the requirements for the degree of Master in Computer Science. Field: COMPUTER SCIENCE.

Orientador:
JOSÉ VITERBO

Co-orientadora:
FLÁVIA BERNARDINI

NITERÓI

2022

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

G633a Gomes, Leonardo Manhães
An Exploratory Analysis of CNN's Robustness to Speaker Identification in Multi-Language Scenarios / Leonardo Manhães Gomes ; José Viterbo, orientador ; Flávia Bernardini, coorientadora. Niterói, 2022.
142 f. : il.

Dissertação (mestrado)-Universidade Federal Fluminense, Niterói, 2022.

DOI: http://dx.doi.org/10.22409/PGC.2022.m.07885802728

1. Convolution neural network. 2. Robustness analysis. 3. Speaker identification. 4. Speaker-focused speech processing. 5. Produção intelectual. I. Viterbo, José, orientador. II. Bernardini, Flávia, coorientadora. III. Universidade Federal Fluminense. Instituto de Computação. IV. Título.

CDD -

Bibliotecário responsável: Debora do Nascimento - CRB7/6368

LEONARDO MANHÃES GOMES

An Exploratory Analysis of CNN's Robustness to Speaker Identification in Multi-Language Scenarios

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: CIENCIA DA COMPUTACÃO.

Aprovada em abril de 2022.

BANCA EXAMINADORA

D.Sc. JOSÉ VITERBO - Orientador, UFF

D.Sc. FLÁVIA BERNARDINI - Coorientadora, UFF

D.Sc. FLÁVIO LUIZ SEIXAS, UFF

D.Sc. CRISTIANO MACIEL, UFMT

Niterói

2022

Dedication: To my 3 precious children Lavínia, Laura and Matheus, my inspirations for all challenges.

"Persistence is the shortest path to the success."
Charles Chaplin

Acknowledgments

I thank God for the faith and persistence so requested and graced. I thank my parents Cícero and Helena for their support, prayers and great encouragement. To my wife Lívia for her help and partnership in times of difficulty and overload. To our children Lavínia, Laura and Matheus for being the inspiration of my triumphs and for being my joys in times of difficulty. To my mentors Viterbo and Flávia for their teachings and for being examples of academic professionalism.

Resumo

O uso de tecnologia para processamento de fala apresentou um grande crescimento nos últimos anos, atuando em diversas áreas como forense, civil ou comercial e contribuindo com a automação de dispositivos eletrônicos. Dentre os dispositivos atualmente providos com serviços de processamento de fala estão *smartphones*, *tablets*, consoles de jogos, dispositivos de navegação controlados por voz e dispositivos inteligentes de assistência operando através de comandos de voz. Portanto, houve um aumento na quantidade de pesquisas relacionadas às tecnologias de processamento de fala e interação por voz, utilizadas por ambientes e dispositivos inteligentes. Aliado a isso, nos últimos anos o uso de Redes Neurais Profundas (RNP) tornou-se um tópico de pesquisa importante em aprendizado de máquina, alcançando também um avanço em áreas de processamento de fala. Em muitas pesquisas as RNPs apresentaram uma excelente capacidade de aprendizagem de características relacionadas à fala e ao orador, tornando-se uma tecnologia bastante utilizada em pesquisas de processamento de fala. Diante deste cenário, inicialmente conduzimos uma Revisão Sistemática da Literatura (RSL) tendo como objetivo entender o estado da arte sobre o Processamento de Fala com Foco no Orador (PFFO) usando Redes Neurais Artificiais (RNA). Foram identificadas 7 áreas de Processamento de Fala, diversas arquiteturas RNA e outras tecnologias e referenciais teóricos relacionados ao tema. Apesar do conhecimento enriquecedor obtido pela RSL, observamos a ausência de trabalhos que propusessem uma análise sobre a robustez das RNAs. Consideramos a robustez uma propriedade importante para um modelo computacional e devido à sua ausência na literatura analisada, tornou-se um problema oportuno e fundamental criarmos este trabalho que realiza uma análise exploratória para avaliar a robustez de uma CNN em cenários de multi-idiomas, usando o Coeficiente Cepstral de Frequência Mel (CCFM) como método para captura das características do orador. Os cenários exploram características distintas relacionadas a um conjunto de dados de oradores que falam multi-idiomas e seus resultados são apresentados e analisados. Como contribuições este trabalho traz as informações fornecidas pela RSL que promove o enriquecimento do conhecimento relacionado à PFFO usando RNA, o termo PFFO, que não foi encontrado por nós em nenhum trabalho da literatura e por último a contribuição metodológica trazida por este trabalho, que apresenta um plano experimental organizado, contendo diferentes cenários de multi-idiomas que exploram uma tarefa de Identificação do Orador (IO) e são executados através de um roteiro de forma organizada e padronizada.

Palavras-chave: Rede neural convolucional, Identificação de Orador, Processamento de fala, Rede neural artificial, Processamento de fala com foco no orador, Coeficiente cepstral de frequência Mel.

Abstract

The use of technology for speech processing has shown great growth in recent years, acting in several areas such as forensics, civil or commercial and contributing to the automation of electronic devices. Among the devices currently provided with speech processing services are smartphones, tablets, game consoles, voice-controlled navigation devices and smart assistance devices operating through voice commands. Therefore, there has been an increase in the amount of research related to technologies for speech processing and voice interaction, used by smart environments and devices. Allied to this, in recent years the use of Deep Neural Networks (DNN) has become an important research topic in machine learning, also achieving a breakthrough in speech processing areas. In many researches DNNs showed an excellent ability to learn features related to speech and speakers, becoming a technology widely used for speech processing research. Given this scenario, we initially conducted a Systematic Literature Review (SLR) aiming to understand the state of the art on Speaker-Focused Speech Processing (SFSP) using Artificial Neural Networks (ANN). Seven areas of Speech Processing, several ANN architectures and other technologies and theoretical references related to the topic were identified. Despite the enriching knowledge obtained by SLR, we observed the absence of works that proposed some sort of analysis about ANN's robustness. We consider robustness an important property for a computational model and, due to its absence in the analyzed literature, it became an opportune and fundamental problem to create this work that performs an exploratory analysis to evaluate the CNN's robustness in multi-language scenarios, using the Mel-Frequency Cepstral Coefficient (MFCC) as a method for capturing speaker features. The scenarios explore distinct characteristics related to a dataset of multi-lingual speakers and their results are presented and analyzed. As contributions this work presents the information provided by the SLR, that promotes enriching knowledge related to the SFSP using ANN, the term SFSP, which was not found by us in any other work of the literature and finally the methodological contribution brought by this work, which presents an organized experimental plan, containing different multi-language scenarios that explore a Speaker Identification (SI) task and are executed through a roadmap in an organized and standardized way.

Keywords: Convolution neural network, Mel-frequency cepstral coefficient, Robustness analysis, Speaker identification, Speech processing, Speaker-focused speech processing.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

1D	:	One-Dimensional
AFIS	:	Automated Fingerprint Identification System
ANN	:	Artificial Neural Network
AP	:	Average Pooling
ATM	:	Automated Teller Machine
BN	:	Bottleneck
CDBN	:	Convolutional Deep Belief Network
CNN	:	Convolutional Neural Network
CONV	:	Convolution
DBN	:	Deep Belief Network
DL	:	Deep Learning
DNN	:	Deep Neural Network
EER	:	Equal Error Rate
FC	:	Fully Connected
FN	:	False Negative
FP	:	False Positive
GMM	:	Gaussian Mixture Models
GRU	:	Gated Recurrent Unit
HMM	:	Hidden Markov Models
IDE	:	Integrated Development Environment
IHC	:	Inner Haircell Coefficients
LPC	:	Linear Prediction Coefficients
LSTM	:	Long Short-Term Memory
MFCC	:	Mel-Frequency Cepstral Coefficient
MP	:	Max Pooling
PLDA	:	Probabilistic Linear Discriminant Analysis
PLP	:	Perceptual Linear Prediction
PNCC	:	Power Normalized Cepstral Coefficients
POOL	:	Pooling

Lista de Abreviaturas e Siglas

PPV	:	Positive Predictive Value
QA	:	Quality Assessment
RASTA PLP	:	Relative Spectral PLP
ResNet	:	Residual Network
RNA	:	Rede Neural Artificial
RNN	:	Recurrent Neural Network
RQ	:	Research Question
SA	:	Speaker Adaptation
SC	:	Speaker Clustering
SD	:	Speaker Diarization
SFSP	:	Speaker-Focused Speech Processing
SFE	:	Speaker-specific Feature Extraction
SI	:	Speaker Identification
SITW	:	Speakers In The Wild
SIWIS	:	Spoken Interaction With Interpretation in Switzerland
SLR	:	Systematic Literature Review
SM	:	Speaker Modeling
SR	:	Speaker Recognition
SS	:	Speaker Segmentation
SSD	:	Speaker Spoofing Detection
STFT	:	Short-Time Fourier Transform
SV	:	Speaker Verification
SVM	:	Support Vector Machine
TN	:	True Negative
TP	:	True Positive
TPR	:	True Positive Rate
UBM	:	Universal Background Model

Sumário

Capítulo 1

Introduction

Using technology for speech processing offers great possibilities for automation, acting in several areas such as forensics, civil or commercial [60]. In the last years there has been a great growth in the use of portable devices equipped with microphones to capture the user's speech in various environments and applications. Such devices include smartphones, tablets, gaming consoles, voice-controlled navigation devices and, more recently, several voice-controlled systems [112]. Smart assistance devices, such as Amazon Echo or Google Home, work as endpoints for Intelligent Virtual Assistants (IVA), such as Alexa, Siri, Google Now or Cortana, which are software agents running as cloud services to process voice commands [27]. Such devices have helped to increase the number of applications related to voice processing and to expand the research on voice interaction technologies in smart environments [77].

In voice interaction, the user's speech is the input for systems or applications [54]. Individual users' speech have to be processed in order to extract speaker features. Speech recognition, which focus on identifying the spoken words [73], is fundamental for implementing vocal interfaces and as such has long been studied [110]. On the other hand, there are applications in which the main purpose is identifying a person by his or her voice, i.e., the recognition is focused on the speaker. Hence, for many different purposes, such as, searching multimedia libraries based on speaker identity, user authentication in access control or for personal identification in forensics [53], the speech processing focusing on the speaker has gained attention over the last years. Although speech is basically a non-stationary signal used for transferring a message via words from a speaker to a listener [103], it has been shown that its analysis is capable of providing additional information about the speaker such as age [20], gender [63], language being spoken [130], emotional state [74] and others. As such, speech provides a wider range of possibilities for security

applications when compared to other biometric features, such as iris and fingerprint [103].

In recent years, the use of Deep Neural Network (DNN) became a hot research topic in machine learning, also achieving a breakthrough in speech recognition [87] and in other speech processing activities. DNNs have presented so far an excellent ability to automatically learn feature representations from high-dimensional input data, as a result of their outstanding performance in many areas [134]. In addition to the use of DNN in the generation of a network model, many researchers have also started to use DNN to extract features from the speaker. According to [61], many researchers have investigated better ways of generating speaker-specific representations and consequently improving the detection of speaker change points in thir speeches. DNNs and deep auto-encoders have been shown to perform quite satisfactorily for this task. One of the most popular DNNs is the Convolutional Neural Network (CNN). It take this name from mathematical linear operation between matrixes called convolution [115]. CNN has an excellent performance in machine learning problems [5]. It has been popular in pattern recognition for non-relational data, such as images and sound processing [134].

Nonetheless, a large number of researches can be found in the literature proposing Deep Learning (DL) models to tackle different nuances or approaches for the Speaker Recognition (SR) problem. Such approaches may be, for instance, the Speaker Identification (SI), which through the speech of a person consists of identifying him in a known population of speakers [55] [21], or the Speaker Verification (SV), which consists in deciding if a speaker is whom he claims to be [21]. Actually, for many authors the SR concept comprises both SI and SV tasks [99] [103] [7] [14]. Besides these tasks, other approaches that are closely related to the SR process are the Speaker Segmentation (SS) [61], the Speaker Diarization (SD) [76], the Speaker Spoofing Detection (SSD) [138] and the Speaker Adaptation (SA) [1]. All of these complementary approaches have in common human speech as the main input and some sort of speaker-related classification as the output.

Considering the diversity of applications related to Speaker-Focused Speech Processing (SFSP) and the great use of ANN, mainly DNN, in researches of speech recognition [87] and other speech processing activities, we decided to initially carry out a Systematic Literature Review (SLR) to identify the state of the art on SFSP using ANN, where we could verify a great variety of information. This SLR presents 7 speech processing areas, different ANN architectures and models and feature extraction methods, as well as other important information related to the topic. A total of 336 articles were collected by SLR and we noticed that there were none that explored ANN's robustness analysis. This ab-

sence motivated the realization of this work because we understand that robustness is an important property for any computational model and because we believe in its contribution to the literature. Faced with this situation, therefore, we proposed an exploratory analysis of CNN's robustness when executing a SI task for multi-language scenarios, using the Mel-Frequency Cepstral Coefficient (MFCC) as a method of capturing speaker features. Each experimental scenario explores distinct characteristics related to a dataset of multi-lingual speakers and their results are presented and analyzed. As contributions this work brings the information provided by the SLR that promotes enriching knowledge related to the SFSP using ANN, the term SFSP, which was not found by us in any other work and therefore, it is suggested by us to represent research that uses speech processing with a focus on learning speaker features. Finally, the methodological contribution created by this work, which presents an organized experimental plan, containing different multi-language scenarios that explore a SI task and are executed through a roadmap in an organized and standardized way.

1.1 Problem Definition

Speech Processing currently has many application possibilities [60]. Some examples of areas in which Speech Processing works are: SI [26], SV [12], SR [44], SS [61], SD [76], SSD [138] and SA [1]. Each of them has different specializations and objectives. To reach their objectives and present good performances, each solution usually need to overcome certain obstacles that are present in analyzed audio and that are seen as problems to be solved. These problems are represented in audio as noise [49] [61], low quality recordings [26] [118] [141], very short audios [12] [66], among others. Specifically for a SI solution, some problems presented by articles that make it difficult to identify the speaker are: people's voices in the background [44] [62] [126], audios recorded indoors or under real-life conditions [136], different languages spoken by the same person [14] [77], speakers from diferente ethnicities or nationalites [28] [131], pronunciation of short sentences [12] [66], telephone or microphone conversations [98] [65].

This understanding of the challenges related to SI research and other speech processing areas was acquired by us from a SLR that analyzed the state of the art related to the topic "SFSP using ANN", which is explained in detail in Chapter 2. However, although this SLR has brought us great knowledge about ANN performing tasks related to SFSP, we observed the absence of works that proposed some sort of analysis about ANN's robustness. According to [108] and [143] robustness is the state where the technology or

process performance is minimally sensitive to factors causing variability. In [69] and [9], robustness is defined as a property that allows a system to maintain its functions despite external and internal perturbations. A system's robustness is understood by [139] as its capacity to guarantee a desired property in face of the largest set of deviant environmental behaviors. According to [122], robustness of an algorithm is its sensitivity to discrepancies between the assumed model and reality.

Therefore, we did not identify any work that evaluated how an ANN would behave in face of variations related to the specific SFSP task for which the ANN has been trained. How much variations in a dataset could impact and influence ANN performance when executing some SFSP task? We consider this analysis of ANN's robustness a complex problem but due to its absence in the literature it becomes an opportune and fundamental problem. Faced with this situation, we thought it would be very important to create a work proposal that would carry out an exploratory analysis to evaluate the robustness of a particular ANN. Analyzing studies with this critical view, we identified a CNN architecture, in one of the articles highlighted by SLR [26], as a good example of an ANN that could be used by an experimental analysis to test its robustness when performing an SI task. And we also think that a good example of a SI problem, that could be explored by this robustness analysis, is the identification of speakers who speak different languages. Another article highlighted by SLR [14], presented a dataset with speakers who speak two and three languages and which has good potential for carrying out experiments in a robustness analysis.

1.2 Objective

Having robustness as an important property for a computational model and due to its absence in the analyzed literature, we present as the main objective of this work the challenge of executing an exploratory analysis of CNN's robustness when performing a SI task in multi-language scenarios. To achieve this main objective, we present the following specific objectives:

- Identify the state of the art related to SFSP using ANN, through a SLR;
- Evaluate the execution of an experimental plan to explore the CNN's robustness performing a SI task.

Robustness analysis evaluated the CNN performance from an experimental plan, following an execution roadmap. To plan and achieve the main objective we rely on the knowledge

acquired from SLR. CNN architecture analyzed was presented by the article [26]. Multilingual scenarios were based on the dataset of multilingual speakers presented by article [14]. Both articles were highly rated by SLR.

1.3 Methodology

This work developed a research with exploratory purpose, under a qualitative-quantitative approach, classified as to its nature as a basic research and as to the technical procedures as a bibliographic and experimental research. According to [47], an exploratory research aims to provide greater familiarity with the problem, making it more explicit or with the objective of constituting hypotheses. Such research has as its main objective the improvement of ideas or the discovery of intuitions. Its planning is quite flexible so that it makes it possible to consider the most varied aspects related to the fact studied. Qualitative approach brings, as a contribution to the research work, a mixture of rational and intuitive procedures capable of helping to better understand the phenomena [104]. Quantitative approach, on the other hand, considers that everything can be quantified, which means translating into numbers, opinions and information to classify and analyze them [30]. The use of both approaches classifies the research as quali-quantitative. Basic research, also referred to as pure research, is interested in generate new scientific knowledge and is, at most, only indirectly involved with how that knowledge will be applied to specific, practical or real problems [57]. A bibliographic research is elaborated from published theoretical references, such as books, articles, periodicals, analyzing and discussing the various scientific contributions that are usually collected through a bibliographic review [15]. Experimental research essentially consists of determining an object of study, selecting the variables that would be able to influence it, defining the forms of control and observation of the effects that the variable produces on the object [47].

Initially, in Chapter 2, this work elaborates a process for the execution of an SLR with the objective of identifying the state of the art of SFSP solutions using ANN. For the collection of articles by the SLR, the search tools of the scientific libraries were used. During the conduction of the SLR, the selected articles were submitted to a Quality Assessment (QA) method, created in this work. The analysis of the most prominent articles in the QA allowed the selection of a CNN from the article [26], a feature extraction method and a dataset containing speakers who speak 2 and 3 languages. An experimental plan, shown in Chapter 4, was then elaborated containing scenarios with different proposals that explored the CNN performance when executing an SI task, allowing to obtain results that

evaluated its robustness. These scenarios represent variations of the same dataset that explore specific situations. The dataset selected for CNN training was presented by the article [14], which is another highly classified by SLR, and contains speakers who speak two or three languages. Its characteristic of multi-languages per speaker contributed to the creation of a diversified experimental plan, for this CNN analysis. In Chapter 5, Accuracy and F1 metrics were used to record results of validation and tests during the CNN learning process. To record the results of the experimental scenarios, the Class and Probabilistic prediction methods were used, which served to evaluate the CNN performance. An exploratory data analysis was done individually for each experimental scenario and finally, a conclusive analysis about the CNN's robustness was made in Chapter 6. To analyze the results of the experimental scenarios, the method of diagnostic analysis was used, which seeks to analyze the relationship between the causes and the effects caused by an experiment performed by CNN. According to [34] [35], the deffinition of "diagnostic analyses" is: the mode of identifying systemic problems and explain their causes can be called.

In summary, for achieving our main and specific objectives, we executed the following actions:

1. Conduction of a SLR to identify the state of the art on SFSP using ANN, taking note of the technologies and characteristics related to this topic, which include: SFSP areas such as SI, SV, SR, SS, SC, SD, SSD and SA; grouping of the solutions identified in the articles into Speaker Modeling (SM) and Speaker-specific Feature Extraction (SFE) solutions; identification of ANN architectures, identification of Features Extraction methods, identification of main metrics, identifying the variety of audio problems presented by the datasets and used in ANN training.
2. Elaboration and execution of an experimental plan to evaluate the robustness of a CNN architecture performing a SI task, considering the following steps:
   (a) Selection of a CNN architecture from a highly rated article by the SLR;
   (b) Selection of a dataset containing speakers who speak two or three languages, from an article highly rated by the SLR;
   (c) Construction of a CNN model based on the CNN architecture selected;
   (d) Execution of experimental scenarios using the CNN model, exploring dataset variations, that include: reduction in the number of speakers, reduction in the number of audios per speaker and variation in the size of speaker audios;
   (e) Execution of experimental scenarios using the CNN model, exploring the main

characteristic of this dataset: the relationship between speakers and languages;

3. Analysis and discussion of results related to the CNN's robustness.

1.4 Organization

This work is organized as follows: Chapter 2 describes the SLR carried out to understand the state of the art related to the use of ANN for SFSP and Chapter 3 explains the technologies and the theoretical reference, selected from the SLR and used by this work, which includes the CNN architecture and the feature extraction method. In Chapter 4, entitled Materials and Methods, we present all the planning prepared for the construction of this exploratory analysis, containing: the presentation of the experimental plan, the explanation of the audio dataset with multiple languages, the organization of general steps for the plan experimental execution and the detailing of experimental scenarios that explore different situations involving the audio dataset. In Chapter 5, the results of CNN training and experimental scenarios are presented, in a consolidated view in tables and graphs, and an individual analysis of each scenario is exposed. Finally, in Chapter 6, the conclusions and a general analysis about the CNN's robustness and about this work are presented.

Capítulo 2

Our Literature Review

In this Chapter we present our methodology for conducting a SLR, aiming at understanding the state-of-the-art on the use of ANNs for SFSP. At the end of the chapter, the SI problem to be analyzed and the ANN architecture selected to carry out this analysis are presented.

2.1 Methodology for Conducting our SLR

The main objective for conducting our SLR was to identify the state-of-the-art in research involving ANN for SFSP. The intention is to learn about what is currently being explored in this type of research. We were interested in knowing what types of ANN are being used, what solution architectures are being proposed, what types of audio datasets have been used and what scenarios or characteristics specific audio signals were explored in research focusing on the speaker. Figure 2.1 shows the steps we executed for conducting our SLR. Step 1 is Planning, which aims to define what is researched and how SLR is conducted. This step involves specifying the Research Questions (RQs) and developing the Review Protocol. Step 2 is Conduction, which aims to execute the SLR from what was defined in the Planning step. This step involves the Search Execution through a search sentence whose topics are directly linked to the research objective, carrying out the Study Selection, conducting a QA on the selected studies and performing the Data Extraction and Analisys of study contents to answer RQs. Step 3 is Conclusion, in which aims to formalize the closure of the SLR. This step involves recording the Interpretation of Results based on the analysis of the studies and the responses obtained by the RQs. Answers to our RQs are in Chapter 2.4 and Results Discussion is in Chapter 2.5.

Some publications were very important for the understanding, adaptations and con-

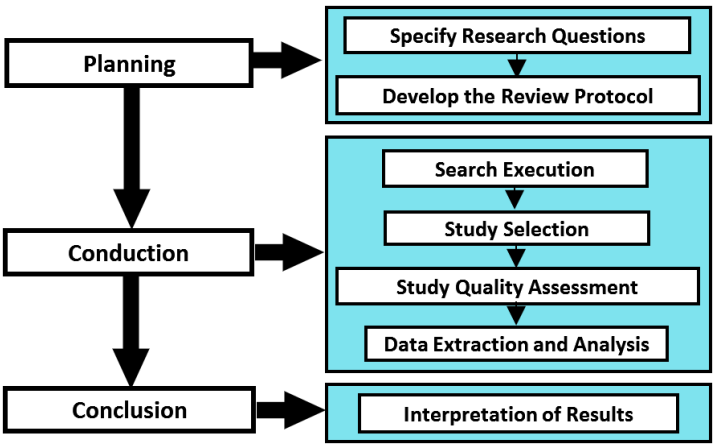


Figura 2.1: Systematic Literature Review Process adapted from [19].

duction of the SLR performed by this research. Articles [68] [111] contributed to the conceptual understanding of SLR and its importance in research. In [19] there was a contribution to the adaptation of the figure 2.1, which represents the SLR process of this research, and a contribution to the learning in conducting the SLR process. [70] contributed to the learning in conducting the SLR process and to the QA elaboration. Article [119] contributed to the learning in conducting the SLR process and to the QA elaboration and the presentation of selected articles from scientific libraries. Through the article [46] the PICOC criterion became known and we incorporated it into the SLR process.

In what follows, we present in more details how we executed each of the tasks in each step.

2.1.1 Planning

Planning is the first step for executing the SLR. It encompasses the execution of activities to specify RQs and to develop the SLR protocol.

Specify RQs: For [119], specifying RQs is the most important part of any systematic review as they conduct all of its methodology. For having this strategic role in the review process, the RQs in this systematic review were structured with the help of the Population, Intervention, Comparison, Outcome, Context (PICOC) criteria, as used by [119], whose meaning and scope of action are explained by [46] and [132] and mentioned in the Table 2.1.

The RQs addressed to this SLR are mentioned below.

- RQ1: What are the SFSP areas found and the percentages of the selected studies

Tabela 2.1: Application of the PICOC criteria for the elaboration of RQs

Criterion	Meaning	Scope
Population	Who or What? The population in which the evidence is collected.	Works published for SFSP using ANN, from January 2015 to February 2019.
Intervention	How? What technology, tool or procedure is being studied?	Use of ANNs for SFSP.
Comparison	Compared to what / what is the alternative?	Performance comparison between different ANNs architectures.
Outcome	What are we trying to accomplish, improve, effect?	Identification of ANNs with better performance results for SFSP.
Context	Under what circumstances?	Automation and security.

that work in each one of them?

- RQ2: What types of ANN architectures with the best performance have been used by studies focusing on SM and SFE?
- RQ3: About conventional feature extraction methods utilization:
  - RQ3.1: What conventional feature extraction methods were used feeding SM ANN?
  - RQ3.2: What conventional feature extraction methods were used as a baseline for comparison with SFE ANN or feeding SFE ANN?
- RQ4: What were the main metrics used by each SFSP areas?
- RQ5: What were the characteristics explored by the studies in the audio datasets?
- RQ6: Which studies had QA with a grade higher than 75% of the maximum grade? About these studies, briefly report the ANN architecture that presented the best performance, the audio aspects explored and the performance measurements presented.

SLR protocol: We constructed the following Search String to collect the papers to be analyzed: (“audio”) AND (“neural network”) AND (“speaker identity” OR “speaker identification” OR “speaker recognition” OR “speaker verification” OR “speaker detection”). We defined the following inclusion criteria for selecting studies for this SLR:

1. Use of ANNs in the phases of modeling or feature extraction for SFSP systems;
2. Availability of material in electronic format, integral, on the web and published in English between January 2015 and February 2019.

Our exclusion criteria were the following:

1. Works that were not speaker-focused and did not present the use of ANN in the modeling or feature extraction stages;
2. Studies that did not clearly present the solution and architecture of the ANN adopted; and
3. Studies that did not show the performance results of the ANN.

According to [68], in addition to the general inclusion/exclusion criteria, it is considered essential to assess the "quality" of primary studies. The benefits provided by a QA can be as follows:

- To provide more detailed inclusion/exclusion criteria;
- To investigate whether quality differences provide an explanation for differences in study results;
- As a means of weighting the importance of individual studies when results are being synthesised;
- To guide the interpretation of findings and determine the strength of inferences;
- To guide recommendations for further research.

Therefore, a QA of the studies selected by this SLR was carried out and for this purpose 10 questions were presented below as QA questions:

- QA1: Was the architecture of the proposed solution clearly and completely presented?
- QA2: Is the ANN the main method to be assessed by the study?
- QA3: Was there clarity in the explanation of the ANN's performance?
- QA4: Was the architecture of the ANN used clearly and completely presented?
- QA5: How was the ANN performance compared to the other methods presented by the study?
- QA6: Does the study describe the dataset used and demonstrate quality and contribution to generating reliability in the results?
- QA7: Do the results of the solutions appear clear and reliable by the way they were presented?
- QA8: Is the ANN text-independent?
- QA9: Does the study have any additional or specific characteristics that represent evolution, innovation or versatility? If the answer is yes, explain these characteristics.

- QA10: Does the research objective present a situation of real relevance in everyday life?

Quality measurement aims to analyze mainly the level of detail and performance of the solution proposed by the study and the level of adherence of the study to the SLR objective. Objective answers were established for each assessment question and a grade was assigned to each answer. The lowest grade refers to the lowest quality answer and the highest to the highest quality answer. A weight was also assigned to each assessment question. The highest weight values were attributed to the assessment questions interpreted by us as being more specific to the objective of this SLR. The final grade of each assessment question is the result of multiplying its weight by the grade of the answer given to the evaluated study. The final assessment grade for each study is the result of the sum of the grades for each question, with the highest grade being 63 and the lowest 0. The possible answers to the assessment questions are recorded in Table 2.2.

At the end of the QA execution, we intended to obtain a list of the studies ordered in descending order by the assessment grade. In theory, studies with higher grades will be presented as solution projects based on ANNs with better quality for this SLR. A clarification in particular to QA8 is that the highest grade was established if the ANN is text-independent because according to the authors of the studies, these are systems with greater flexibility and versatility in relation to ANNs that are text-dependent. According to [103], Text-Dependent systems can be used only for co-operative users and the user needs to utter whatever is being prompted by the system. In Text-Independent system, there are no constraints on the words that can be used. So this type of system provides more flexibility for the users, but is more difficult to implement.

Data Extraction and Analysis: Information were extracted from each selected study which helped in its cataloging, assessment and to answer the SLR's RQs. The information extracted for cataloging were: title, summary, year of publication, authors and publisher, extracted through library search engines. The main data identified to assist in the assessment and responses to the RQs were: SFSP areas, types of ANNs, ANN architectures, dataset used, characteristics explored in the audios, performance evaluation metrics, results of comparison with other methods, use new theories or technologies. The publication files of the selected studies were obtained in PDF format. The RQ6 will identify the studies that obtain a grade higher than 75% in the QA. For these studies, a detailed analysis will be made verifying the solutions presented, the research objectives, the performance results presented and the audio aspects explored.

Tabela 2.2: Answers to the QA questions of the SRL.

QA	Weight	Replies					Maximum Grade by QA
		Grade 0	Grade 1	Grade 2	Grade 3	Grade 4	
QA1	3	It was not presented or presents an obvious solution given the scientific understanding of the analyzed context.	It is not clear or is incomplete.	Understandable but with little detail or due to complexity it could be better explained.	Clear and complete.		9
QA2	3		It is not the main method.	Divides focus with another method.	It is the main method.		9
QA3	2	No.	Incomplete or insufficient explanation or due to complexity could be better explained.	Yes.			4
QA4	4	It was not presented.	It is not clear or is incomplete.	Understandable but with little detail.	Clear and complete.		12
QA5	3	No result has been demonstrated to assess performance.	Demonstrates results but does not compare with any other method.	Compares with its own methods or with those of other articles not considered state-of-the-art or performs much less than the state of the art.	Performance minimally below the state of the art or demonstrates superiority but not as convincing (lack of detail).	State-of-the-art performance equivalent or superior.	12
QA6	3	Does not present datasets or uses inconsistent or poorly reliable datasets to generate results.	Presentes coherent datasets but uses little data or does not mention the amount of data used.	Presentes coherent datasets using large amounts of data and presenting planning for its uses.			6
QA7	2	No results, clarity or reliability.	The results could have been more detailed to contribute with clarity and reliability.	Clearly and apparently reliable results.			4
QA8	2	It is not, does not mention or makes it clear.	Yes.				2
QA9	2	No.	Yes.	Features additional or specific features.			4
QA10	1	No.	Yes.				1
				MAXIMUM GRADE			**63**

Tabela 2.3: Number of studies grouped by year of publication

Scientific Library	Number of Studies per Year					TOTAL
	2015	2016	2017	2018	2019	
ACM Digital Library	1	0	8	5	1	15
Engineering Village	5	11	13	18	0	47
ScienceDirect	25	41	55	60	31	212
Scopus	7	11	22	21	1	62
TOTAL	38	63	98	104	33	**336**

2.1.2 Conduction

When conducting our SLR, we carried out the following activities: Search Execution, Studies Selection, QA of the studies and Data Extraction and Analysis using the selected studies.

Search Execution: We used the following bases of scientific articles: ACM Digital Library, Engineering Village, ScienceDirect and Scopus. According to [51], who make a comparison between 28 academic consultation bases and evaluate 26 requisites, ACM Digital Library, ScienceDirect and Scopus have a significant content depth and are among the 14 academic bases advised as principal search systems, motivating their choices as sources for our literary search. We chose Scopus because it is a reputable scientific library containing 82 million documents dating back to 1788, 17 million author profiles, 80,000 institutional profiles and 1.7 billion cited references dating back to 1970 [37]. ScienceDirect is Elsevier's premier platform for peer-reviewed academic literature, containing 19 million articles & chapters, 2,650 peer-reviewed journals, 43,000 Ebooks, and 1.4 million open access articles, serving academic institutions, government organizations and research & development units across a variety of industries [36] [40]. Another reference for publications chosen was Engineering Village, which is a platform of indexing and abstracting databases in engineering and related fields. It provides access to 12 patent and engineering literature databases covering a wide range of trusted engineering sources [39] including Ei Compendex, which is considered the most comprehensive database for engineering literature [38]. The ACM Digital Library is a research, discovery and networking platform containing a comprehensive bibliographic database focused exclusively on the field of computing [4].

We used our previously defined Search String, restricting the search to articles published between January 2015 and February 2019 and in research areas related to Computer Science. The result of the Search Execution presented a total of 336 studies for the 4 libraries, as can be seen in Table 2.3.

Study Selection: We selected the studies to be analyzed executing 5 phases: executing the Search String in the selected bases, reading the study titles, reading the study abstracts, reading the studies and grouping the selected studies. The first four phases selected the studies adhering to the objective of the SLR and the last phase separated the studies that focus on SM and SFE. We used our inclusion and exclusion criteria for selecting the studies in all the first four phases. The selection process started with 336 studies and ended with 34 studies. Tables 2.4 and 2.5 show the number of studies selected in each phase of the process, for each of the 4 scientific libraries. The difference between the Tables 2.4 and 2.5 is Table 2.4 presents studies that focus on SM and Table 2.5 shows studies that focus on SFE. Tables 2.6 and 2.7 list the selected studies focusing on SM and SFE, respectively. The following acronyms were used in these tables to classify the profiles of selected studies that present ANNs that perform SFSP: SM (Speaker Modeling), SFE (Speaker-specific Feature Extraction), SI (Speaker Identification), SV (Speaker Verification), SR (Speaker Recognition), SS (Speaker Segmentation), SD (Speaker Diarization), SSD (Speaker Spoofing Detection) and SA. It is important to note that the study S11, from the Engineering Village library, is accounted for both Tables 2.4 and 2.5 and listed in both study groups, shown in Tables 2.6 and 2.7, as it presents ANN solutions for both SM and SFE. Therefore, the 34 studies resulted in a total of 35 QAs performed.

Tabela 2.4: Selection process of studies focusing on SM

Scientific Library	Search Execution	Title Reading	Abstract Reading	Study Reading	Selected SM Studies	Percentage of Selected
ACM Digital Library	15	10	3	2	2	13.33%
Engineering Village	47	27	18	16	11	23.40%
ScienceDirect	212	46	26	5	1	0.47%
Scopus	62	39	17	11	8	12.90%
Total Selected	**336**	**122**	**64**	**34**	**22**	6.55%

Tabela 2.5: Selection process of studies focusing on SFE

Scientific Library	Search Execution	Title Reading	Abstract Reading	Study Reading	Selected SFE Studies	Percentage of Selected
ACM Digital Library	15	10	3	2	0	0.00%
Engineering Village	47	27	18	16	6	12.77%
ScienceDirect	212	46	26	5	4	1.89%
Scopus	62	39	17	11	3	4.84%
Total Selected	**336**	**122**	**64**	**34**	**13**	3.87%

Data Extraction: After the reading and selection phases, data from the studies were extracted in order to carry out the Literature Analysis and answer the RQs. Tables 2.8 e 2.9 list some information extracted from the studies focusing on SM and SFE, respectively. In the column "Types of ANNs", a categorization of the types of ANNs

Tabela 2.6: List of selected studies focusing on SM

Cod	Study	Authors	Year	Library	Profile
S1	Extracting sub-glottal and Supra-glottal features from MFCC using convolutional neural networks for speaker identification in degraded audio signals	Anurag Chowdhury e Arun Ross	2017	Engineering Village	SI
S2	Audio classification using attention-augmented convolutional neural network	Yu Wu, Hua Mao e Zhang Yi	2018	Scopus	SI
S3	Discriminative deep audio feature embedding for speaker recognition in the wild	Simone Bianco, Elia Cereda e Paolo Napoletano	2018	Engineering Village	SR
S4	Deep speaker embeddings for short-duration speaker verification	Gautam Bhattacharya, Jahangir Alam e Patrick Kenny	2017	Engineering Village	SV
S5	Speakers In The Wild (SITW): The QUT speaker recognition system	Houman Ghaemmaghami, Md Hafizur Rahman, Ivan Himawan, David Dean, Ahilan Kanagasundaram, Sridha Sridharan e Clinton Fookes	2016	Engineering Village	SR
S6	A Simple Neural Network Based Countermeasure for Replay Attack	Wenfeng Pang e QianHua He	2017	ACM Digital Library	SSD
S7	Investigating Raw Wave Deep Neural Networks for End-to-End Speaker Spoofing Detection	Heinrich Dinkel, Yanmin Qian e Kai Yu	2018	Engineering Village	SSD
S8	An unsupervised neural prediction framework for learning speaker embeddings using recurrent neural networks	Arindam Jati e Panayiotis Georgiou	2018	Engineering Village	SS
S9	The IBM speaker recognition system: Recent advances and error analysis	Seyed Omid Sadjadi, Jason W. Pelecanos e Sriram Ganapathy	2016	Scopus	SR
S10	Using Convolutional Neural Networks to Classify Audio Signal in Noisy Sound Scenes	M.V. Gubin	2018	Scopus	SV
S11	Advances in deep neural network approaches to speaker recognition	Mitchell McLaren, Yun Lei e Luciana Ferrer	2015	Engineering Village	SI
S12	Speaker Recognition for Robotic Control via an IoT Device	Zhanibek Kozhirbayev, Berat A. Ero, Altynbek Sharipbay e Mo Jamshidi	2018	Scopus	SR
S14	Speaker identification for the improvement of the security communication between law enforcement units	Jaromir Tovarek e Pavol Partila	2017	Engineering Village	SI
S15	Weaknesses of voice biometrics - Speaker verification spoofing using speech synthesis	Milan Rusko, Marian Trnka, Sakhia Darjaa e Marian Ritomský	2017	Scopus	SSD
S16	Speaker identification based on combination of MFCC and UMRT based features	Anett Antony e R. Gopikakumari	2018	Engineering Village	SI
S17	Audiovisual speaker identification based on lip and speech modalities	Fatma Chelali e Amar Djeradi	2017	Scopus	SI
S18	Speaker identification framework by peripheral and central auditory models	Masanori Morise e Kenji Ozawa	2015	Scopus	SI
S19	Real time implementation of speaker recognition system with MFCC and neural networks on FPGA	Bhanuprathap Kari e S. Muthulakshmi	2015	Scopus	SR
S30	Voxceleb2: Deep speaker recognition	Joon Son Chung, Arsha Nagrani, Andrew Zisserman	2018	Engineering Village	SV
S32	Speaker diarization system using HXLPS and deep neural network	V. Subba Ramaiah, R. Rajeswara Rao	2018	Science Direct	SD
S33	Multi-talker Speech Separation With Utterance-Level Permutation Invariant Training of Deep Recurrent Neural Networks	Morten Kolbæk, Dong Yu, Zheng-Hua Tan, Jesper Jensen	2017	Engineering Village	SD
S34	Deep Learning Backend for Single and Multisession i-Vector Speaker Recognition	Omid Ghahabi e Javier Hernando	2017	ACM Digital Library	SV

Tabela 2.7: List of selected studies focusing on SFE

Cod	Study	Authors	Year	Library	Profile
S11	Advances in deep neural network approaches to speaker recognition	Mitchell McLaren, Yun Lei e Luciana Ferrer	2015	Engineering Village	SI
S13	Speaker verification based on extraction of deep features	Evangelos Mitsianis, Evaggelos Spyrou e Theodore Giannakopoulos	201	Engineering Village	SV
S20	A complete end-to-end speaker verification system using deep neural networks: From raw signals to verification result	Jee-Weon Jung, Hee-Soo Heo, IL-Ho Yang, Hye-Jin Shim e Ha-Jin Yu	2018	Scopus	SV
S21	Employing phonetic information in DNN speaker embeddings to improve speaker recognition performance	Md Hafizur Rahman, Ivan Himawan, Mitchell Mclaren, Clinton Fookes e Sridha Sridharan	2018	Engineering Village	SR
S22	DNNs for unsupervised extraction of pseudo speaker-normalized features without explicit adaptation data	Neethu Mariam Joy, Murali Karthick Baskar e S. Umesh	2017	Science Direct	SA
S23	Text-independent speaker verification using convolutional deep belief network and Gaussian mixture model	Ivan Rakhmanenko e Roman Meshcheryakov	2017	Scopus	SV
S24	Deep feature for text-dependent speaker verification	Yuan Liu, Yanmin Qian, Nanxin Chen, Tianfan Fu, Ya Zhang e Kai Yu	2015	Science Direct	SV
S25	Deep neural network based i-vector mapping for speaker verification using short utterances	Jinxi Guo, Ning Xu, Kailun Qian, Yang Shi, Kaiyuan Xu, Yingnian Wu e Abeer Alwan	2018	Science Direct	SV
S26	Speaker verification based on the fusion of speech acoustics and inverted articulatory signals	Ming Lia, Jangwon Kimd, Adam Lammertd, Prasanta Kumar Ghoshe, Vikram Ramanarayanand e Shrikanth Narayanan	2016	Science Direct	SV
S27	Deep neural network-based bottleneck feature and denoising autoencoder-based dereverberation for distant-talking speaker identification	Zhaofeng Zhang, LongbiaoWang, Atsuhiko Kai, Takanori Yamada, Weifeng Li and Masahiro Iwahashi	2015	Engineering Village	SI
S28	Speaker2Vec: Unsupervised learning and adaptation of a speaker manifold using deep neural networks with an evaluation on speaker segmentation	Arindam Jati e Panayiotis Georgiou	2017	Scopus	SS
S29	A comparison of neural network feature transforms for speaker diarization	Sree Harsha Yella e Andreas Stolcke	2015	Engineering Village	SD
S31	Speaker diarization with LSTM	Quan Wang, Carlton Downey, Li Wan, Philip Andrew Mansfield, Ignacio Lopz Moreno	2018	Engineering Village	SD

presented by the study was devised. In the "ANN Architecture" column, the ANN architecture that presented the best performance in each study is represented in a very simplified way. In this representation, only the quantities and types of processing layers were recorded in the composition of the ANNs. The objective was to show in a simplified way the complexity and the size of the ANN of each study. Information, such as the number of neurons in each layer or the type of activation function used by ANN, has been suppressed. The following acronyms were used to represent information about the types of ANN and its architectures recorded in the Tables 2.8 and 2.9: 1D (1 Dimension), ANN (Artificial Neural Network), AP (Average Pooling), BN (Bottleneck), CDBN (Convolutional Deep Belief Network), CNN (Convolutional Neural Network), CONV (Convolution), DBN (Deep Belief Network), DNN (Deep Neural Network), FC (Fully Connected), GRU (Gated Recurrent Unit), LSTM (Long Short-Term Memory), MLP (Multi-Layer Perceptron), MP (Max Pooling), POOL (Pooling), ResNet (Residual Network) and RNN (Recurrent Neural Network). Table 2.8 presents the models used as baselines for comparison with the SM ANN solutions proposed by the studies and answers whether the ANN exceeds the performance of the baselines. In the Table 2.9 the same types of information are also presented but one more column is shown that informs the conventional feature extraction/conversion methods used on audios by the studies to feed the SFE ANN.

After reading all the papers, we firstly created a taxonomy for gathering all the tasks regarding to SFSP when the speaker audio is the focus of the application, which we present in next Chapter. Chapter 2.4 presents the answer to our RQs. For presenting an unified overview of these tasks and the technologies based on ANNs, Chapter 2.5 presents a discussion of our results.

2.2 Speaker-Focused Speech Processing (SFSP)

Speech is regarded as a speaker's biometric characteristic that can be analysed for some specific purpose [77]. Speech processing focusing on the speaker information have gained attention over the last years for many different solutions. However, as far as we know, there is a lack of an overview and a taxonomy that embraces different research areas with a focus on the speaker. Figure 2.2 was taken from article [21] and shows the representation of some areas that descend from speech processing. According to our SLR, we were able to identify 5 new areas descending from speech processing and we show them in Figure 2.3 : Speaker Adaptation, Speaker Diarization, Speaker Segmentation, Speaker Clustering and Speaker Spoofing Detection. We base on [21], registered in Figure 2.2, to propose Figure

Tabela 2.8: Information from selected studies focusing on SM

Cod	Area	Type of ANN	Architecture of SM ANN (proposed)	Baseline SM Solution (compares with SM ANN)	Proposed ANN performs better?
S1	SI	1D-CNN	1D-CNN(3 CONV+1 FC)	UBM-GMM, i-vector/PLDA	Yes
S2	SI	CNN	CNN(3 basic CONV +5 CONV/POOL +2 Attention-based+1 FC)	VGG-11, ResNet-18, ResNet-34, ResNet-50, i-vector	Yes
S3	SR	CNN(ResNet)	CNN(ResNet)(34 CONV +2 MP+3 FC)	i-vectors/SVM, i-vectors/PLDA/SVM, VoxCeleb CNN, VoxCeleb CNN(variations), ResNet-18	Yes
S4	SV	CNN	Deep CNN(8 CONV+4 MP +2 FC Attention)	i-vectors, Feedforward(mean), Feedforward(Attention), Convnet	In some tests.
S5	SR	DNN	DNN(6 hidden layers)	ANN Models of SITW 2016	Yes
S6	SSD	1D-CNN-RNN	1D-CNN-RNN(2 CONV-1D +1 GRU+1 FC)-model 2	1D-CNN-RNN-model 1	Yes
S7	SSD	CNN-RNN	CNN-RNN(3 CONV +1 LSTM+2 FC)	GMM(MFCC), DNN(CQCC8k-DD), GMM(CQCC4k), DNN, CNN, LSTM	In some tests.
S8	SS	RNN	Siamese RNN(2 GRU(3 layers) +1 FC)	algorithm Bayesian Information Criterion (BIC)	Yes
S9	SR	DNN	DNN(7 FC+1 BN)	GMM	Yes
S10	SV	CNN-RNN	CNN-RNN(3 CONV +1 FC / RNN)	CNN-RNNs(1 to 6 hidden layers)	Yes
S11	SI	DNN	DNN	UBM	No
S12	SR	ANN	ANN(2 hidden layers)	There is not	No comparisons with other solutions. Uses the same ANN in different scenarios.
S14	SI	ANN	ANN(1 hidden layer)	There is not	No comparisons with other solutions.
S15	SSD	DNN	DNN(6 hidden layers)	Unit Selection synthesizer, HMM synthesizer	Yes
S16	SI	MLP	MLP(1 hidden layer)	There is not	No comparisons with other solutions. Uses ANN but it is not its main research focus.
S17	SI	MLP	MLP(1 hidden layer)	There is not	No comparisons with other solutions.
S18	SI	ANN	ANN(2 hidden layers)	ANNs(1 to 4 hidden layers)	Yes
S19	SR	MLP	MLP(1 hidden layer)	There is not	No comparisons with other solutions.
S30	SV	CNN(ResNet)	CNN(ResNet)(50 CONV +1 MP+1 AP+2 FC)	i-vectors/PLDA, VGG-M, ResNet-34	Yes
S32	SD	DNN	DNN	There is not	No comparisons with other solutions. Uses ANN but it is not its main research focus.
S33	SD	DNN-CNN-RNN	DNN(3 hidden layers)/ CNN(11 CONV+6 MP+1 AP)/ Bi-directional LSTM (3 layers)	There is not	No comparisons with other solutions. Uses the same ANN in different scenarios.
S34	SV	DBN-DNN	DBN-Universal DBN-DBN-DNN(3 hidden layers)	DNN(1 hidden layer)	No

Tabela 2.9: Information from selected studies focusing on SFE

Cod	Area	Types of ANN	Architecture of SFE ANN (proposed)	Conventional Feature Extraction/ Conversion Method (feeds SFE ANN)	Baseline SFE Solution (compares with SFE ANN)	Proposed ANN performs better?
S11	SI	DNN(BN)	DNN(4 layers+BN)	filterbank	MFCC, pcaDCT	No
S13	SV	CNN	CNN(3 hidden layers +2 MP)	STFT	algorthm Speeded Up Robust Features (SURF), algorthm Scale-Invariant Feature Transform (SIFT)	Yes
S20	SV	CNN-RNN	CNN(5 CONV/POOL) +LSTM+1 FC	Pre-emphasis	MLP(7 layers) /d-vector /mel-filterbank energies	Yes
S21	SR	DNN(BN) -DNN(BN)	DNN(7 hidden layers +BN)+DNN(6 hidden layers+BN)+DNN /x-vector(7 hidden layers+statistic POOL)	MFCC	UBM/i-vector, DNN/i-vector and BN/i-vector	Yes
S22	SA	DNN	DNN(6 hidden layers)	filterbank with pitch information, MFCC, LDA	MFCC, LDA, filterbank, Conventional FMLLR, Basis FMLLR with and without VTLN	Yes
S23	SV	CDBN	CDBN(3 layers)	Not reported	MFCC, Greedy Add-del algorithm	No
S24	SV	DNN	DNN(2 hidden layers)	PLP	PLP	Yes
S25	SV	DNN(ResNet)	DNN(6 layer+ Residual Block(2 FC +BN))	mel-filterbank	MFCC	Yes
S26	SV	DBN	DBN	MFCC	MFCC	Only when combined with the baseline.
S27	SI	DNN-DNN(BN)	DNN(3 layers)- DNN(9 layers+BN)	MFCC	CMN+MFCC, MCLMS-SS, MSLP-SS, BF/MLP, BF/DNN, DAE	Yes
S28	SS	DNN	DNN(5 hidden layers)	MFCC	Some popular distance metrics: BIC with MFCC13, GD with MFCC13, KL2 with MFCC13, KL with MFCC40; and references from other authors with some state-of-the-art papers	Yes
S29	SD	DNN-ANN	DNN(2 hidden layers+BN) and ANN(1 hidden layer+BN)	MFCC	MFCC	In some tests.
S31	SD	RNN	LSTM(3 layers +final linear layer)	log-mel-filterbank energies	GMM-UBM/i-vector	Yes

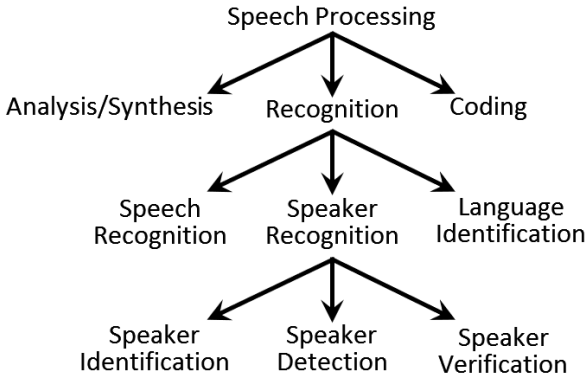


Figura 2.2: Speech processing and other descendant areas, according to [21].

2.3, considering that there was an evolution from 1997 to the actual days. Considering that our SLR focused on the speaker, we believe that other research focuses, also derived from Speech Processing, may have emerged from 1997 to the present day, which would contribute to other ramifications in Figure 2.3. Therefore, Figure 2.3 shows the areas derived from speech processing and in bold the areas classified as SFSP by our proposed taxonomy are highlighted. We use the "+" sign to show the cases in which the predecessor area needs to join the successor areas to be considered.

During the SLR execution, 35 solutions were selected representing 7 different speaker-focused research areas. Among the researched works we did not find any that presented the same approach profile as this SLR. In view of the identified research areas, we felt the need for a taxonomy that would represent the research originated from speech processing but with a specific research focus on the characteristics related to the speaker. So we propose the term Speaker-Focused Speech Processing (SFSP). We have not found any work in the bibliography using this term. In the next subsections we provide a brief explanation of the 8 research areas identified by SLR and classified by us as SFSP: SI, SV, SR, SS, SC, SD, SSD and SA. Also, we identified some concepts used in SFSP regarding to ANNs as well as the typical steps for training ANNs for SFSP. All of these items are described in what follows.

2.2.1 SFSP Areas

2.2.1.1 Speaker Verification (SV), Speaker Identification (SI) and Speaker Recognition (SR)

A SV system aims to use a speech sample to test whether a person who claims to have produced the speech has done so indeed [55] [64]. This is a binary classification problem.

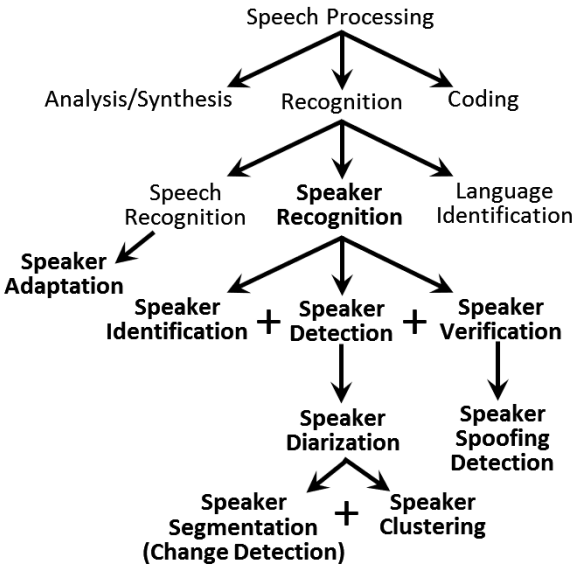


Figura 2.3: New branches of SFSP considering the evolution of research in speech processing from 1997 to the actual days.

In other words, the system checks whether the speaker is the person he claims to be and answers whether this statement is correct or wrong [14]. SV has been widely applied in telephone or network access control systems, such as telephone banking or apartment security [138].

A SI system uses a speech sample to select the identity of the person that produced the speech from among a population of speakers [55] [14]. It searches for the best matching speaker among the residents already known to the system but it may be that the unknown speaker is not enrolled in the system. For this reason, in many systems, SI is followed by SV [64]. It is possible to use this technique to verify speaker identity and control access to services such as voice dialing, banking by telephone, telephone shopping, database access services, information services, voice mail, security control for confidential information areas, and remote access to computers [55].

SR is a specific vocal interaction task and refers to the use of an automated system or machine to recognize persons from their voices [103]. It is a multi-disciplinary task that uses the speaker vocal features to deduce information about speaker identity. It is a branch of biometrics that may be used for identification, verification and classification of individual speakers [10]. There are two types of SR systems such as text-dependent or text-independent. The former system utilizes a fixed utterance for training and testing a person, whereas the later one does not employ a fixed phrase for any cases [77]. At first glance, the 3 topics, SR, SI and SV, seem to have the same meaning, but there

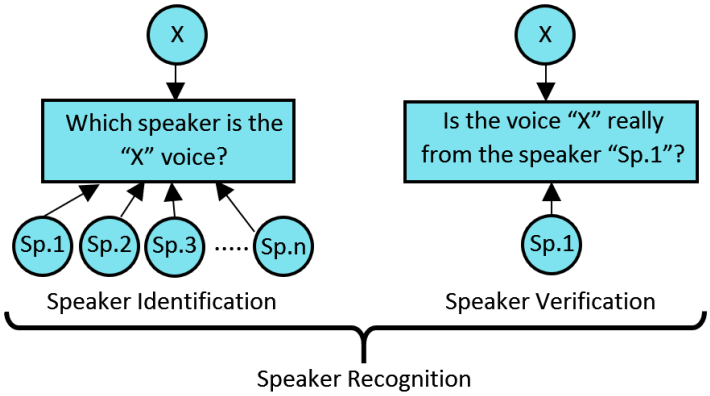


Figura 2.4: Representation of speaker identification, verification and recognition systems

are details that differentiate their types of performances. The SR concept includes both SI and SV tasks [99] [103] [7] [14]. Figure 2.4 shows a representation of the SI, SV and SR systems. SR usage is rapidly increasing and some applications include: access control, online transactions, law enforcement, speech data management, multimedia and personalization [77].

2.2.1.2 Speaker Segmentation (SS), Speaker Clustering (SC) and Speaker Diarization (SD)

SS, sometimes known as Speaker Change Points Detection, refers to the task of dividing an audio signal into multiple audio chunks such that each of them denotes a speaker homogeneous region, ideally containing only one speaker [76] [61]. It aims to detect all speaker change points [109]. SC refers to unsupervised classification of speech segments based on speaker voice features. SS followed by SC is called SD [76]. Therefore, a SD system consists of two main parts: segmentation and clustering [109]. It is the process of partitioning an input audio stream into homogeneous segments according to the speaker identity [131]. It answers the question "who spoke when?" in a multi-speaker environment. In particular, the speaker boundaries produced by diarization systems have the potential to significantly improve acoustic speech recognition accuracy. [131] further explain that a typical SD system usually consists of four components:

1. Speech segmentation, where the input audio is segmented into short sections that are assumed to have a single speaker, and the non-speech sections are filtered out;
2. Audio embedding extraction, where specific features such as MFCC, speaker factors or i-vectors are extracted from the segmented sections;

3. Clustering, where the number of speakers is determined, and the extracted audio embeddings are clustered into these speakers; and optionally
4. Resegmentation, where the clustering results are further refined to produce the final diarization results.

SS has numerous applications in the fields of SD, speaker tracking and automatic speech recognition [61]. SD has a wide variety of applications including multimedia information retrieval, speaker turn analysis, and audio processing [131]. The main SS and SD utilization is rich transcription applications. Rich transcription is a transcription of a recorded event. It can works to generate readable transcriptions of conversational speech in multiple languages, incorporating capitalization, punctuation, speaker change point markers (segmentation) and speaker clustering (diarization).

2.2.1.3 Speaker Spoofing Detection (SSD)

State-of-the-art SSD systems have achieved great performance in recent times. However, performance is usually measured in an ideal scenario where impostors do not try to disguise their voices to make them similar to the target speakers and where target speakers do not try to conceal their identities [88]. The progress of speech synthesis technology leaded to automatic SV systems encountering serious spoofing attacks challenges. Spoofing attacks is the technique where an imposter can easily steal the voiceprint information of a target speaker and use the stolen information to generate high quality speech signals similar to those of the target speaker, through voice conversion or speech synthesis techniques [138] [88]. The generated speech can then be used to attack SV systems [138] where an attacker masquerades as a target enrolled speaker in order to gain illegitimate access to the system [52]. That is, while the performance of automatic SV systems have considerably improved during recent years, as in the case for any other biometric person authentication systems, reliability of automatic SV systems against spoofing attacks (also known as presentation attacks) has become an important security concern [52]. Typical counterfeiting attacks performed on SV systems can be done by different techniques. [120] list such techniques and explain their meanings:

- Impersonation - It refers to spoofing attacks with human-altered voices and is one of the most obvious forms of spoofing;
- Replay attacks - It uses speech recordings of a genuine client or concatenation of shorter speech segments;

- Voice conversion - It is a technique that electronically converts one speaker's voice towards that of another;
- Speech synthesis - It uses a speech synthesizer system adapted to the voice of genuine clients;
- Artificial, non-speech-like tone signals - Certain short intervals of converted speech yield extremely high scores or likelihoods. Such intervals are not representative of intelligible speech but they are nonetheless effective in overcoming typical SV systems.

Speech synthesis and voice conversion attacks in turn, have gained more attention due to two reasons: first, both speech synthesis and voice conversion attacks techniques have improved significantly where high quality speech signals can be generated with limited amount of training data and the second, the availability of freely available open-source speech synthesis and voice conversion tool kits which can easily be used by non-expert attackers [52]. To prevent these attacks, combat methods known as SSD have been developed. SSD is a binary classification task and your goal is to discriminate spoofed speech from the genuine speech [52]. Research in SSD has been carried out to improve the security of SV systems and uses several techniques. DL has already been successfully introduced into the speaker falsification research community as can be seen in [75] and [25]. As another example, [52] highlights the main contribution of the paper presented by [116] that uses features extracted of audios in the training of DNN for SSD. According to [33], the spoofing detection community is focused on two directions: SFE and classifier optimization in the SM.

2.2.1.4 Speaker Adaptation (SA)

In speech recognition, SA refers to the range of techniques whereby a speech recognition system is adapted to the acoustic features of a specific user using a small sample of utterances from that user [123] [1]. The development of speaker-independent speech recognition systems has seen significant progress; however, the recognition performance of these systems has not yet reached that of speaker-dependent speech recognition systems in which a user's speech is registered before hand. Much hope has therefore been placed on the establishment of SA techniques that can bring performance of a speaker-independent system up to that of a speaker-dependent one using the smallest amounts of data [123]. In many cases it is undesirable to train an speaker-dependent system due to the large amount of training data needed and hence the required enrollment time. Therefore SA techniques

which tune an existing speech recognition system to a new speaker are of great interest. Adaptation methods require a sample of speech (adaptation data) from the new speaker so that the models can be updated. The amount of adaptation data needed depends on the way the SA technique uses the data and on the type of system to be adapted [84]. [81] explain SA can be formulated in a number of ways and mentions four different SA setups, as seen:

1. adaptive clustering, in which an existing set of speaker-independent speech models is updated using a new set of speaker-independent data;
2. speaker transformation or speaker conversion, in which a well-trained model for one speaker is converted into a model for a new speaker using a small amount of speaker-specific training data;
3. speaker adaptation, in which a speaker-independent (or multispeaker) model is adapted to a single speaker using speaker specific training data from the new talker;
4. sequential adaptation, in which speaker-specific training data are acquired over time, and the speaker-dependent model is adapted sequentially every time that new training data is acquired.

These implementations differ only in the ways in which the training data are utilized; the adaptation techniques involved are usually very similar. To [65], various SA/normalization approaches in DNN can be classified into three categories: network components adapted towards a particular speaker, speaker-normalized input features and input features appended with speaker-specific features. The first category adapts either the weight functions or the bias terms in DNN to a particular speaker. The second category provides speaker-normalized features as DNN inputs. In the third category, DNNs are made aware of the speaker in- formation during training by augmenting DNN input with auxiliary codes that carry speaker information [81].

2.3 Artificial Neural Networks (ANNs) for SFSP

In this Chapter we show the types of ANNs identified by our SLR and explain each one, based on broader concepts available in various works of literature. We also find it interesting to explain some concepts, which we found in the analyzed articles, that are used in the ANN training processes for SFSP. Finally, we show the steps that make up the ANNs training for SFSP after analyzing in the articles the explanations given by each one of them about the step by step used in the execution of their SFSP processes.

2.3.1 Types of ANNs

Generic ANN Model: According to [106], ANNs are computational processing systems heavily inspired by biological nervous systems (such as the human brain) operation. ANNs are mainly comprised by a high number of interconnected computational nodes (referred to as neurons), working entwine in a distributed fashion to collectively learn from the input in order to optimise its final output. The calculation within a neuron is generally done by two operators: one input operator and one activation function. The input operator can be, for example, the result of sum the neuron inputs and one bias. The activation function determines the output of the neuron from the result of the input operator. Neurons are connected to each other by synaptic connections, which relay the output of a neuron to the input of another, multiplying it by a coefficient called synaptic weight. Neurons are organized in layers, which are divided into three groups: one input layer, one or many hidden layers and one output layer. Too many hidden layers can lead to an overfitting phenomenon of the ANN; not enough layers can lack robustness for the ANN to learn the problem. Synaptic weights and biases are optimized by a training algorithm [16]. [106] state that the input data is usually loaded in the form of a multidimensional vector to the input layer, which will be distributed to the hidden layers. The hidden layers will then make decisions from the previous layer and weigh up through a stochastic change. Weights updating is referred to as the process of learning.

Multilayer Perceptron (MLP): MLP is a very simple model of ANN and is based on the principle of a feed-forward-flow of information, i.e. the network is structured in a hierarchical way [128]. It is important to mention that the MLP model is a variant form of the classical ANN model and this model has been widely used in the current era of big data analytics. The basic MLP model comprises of three layers: (I) input layer, (II) hidden layer and (III) output layer. Basically the input layer receives the set of input data, the processing of the features is performed in the hidden layer(s), and the output layer is used to reveal the predicted results[117]. The input units play no active role in processing the information flow, because they just distribute the signals to the units of the first hidden layer. All hidden units work in an identical way and the output unit is a simpler version of a hidden unit. In an MLP, each hidden unit transforms the signals from the former layer to one output signal, which is distributed to the next layer. Each hidden unit has an, in general nonlinear, activation function. The activation function is modulo a translation via

an individual bias, the same for all hidden units. The output of a hidden unit is determined by the weighted sum of the signals from the former layer, which is then transformed by the activation function. In the output unit the activation function is the identity function[128].

Deep Neural Network (DNN): On the other hand, an ANN having many multiple hidden layers, stacked upon each-other, is commonly called DNN. DNN architectures are characterized by one or more hidden layers consisting of hidden nodes, with each hidden node representing a nonlinear activation function [41]. ANNs differ by their architectures composition, represented by different amounts and types of layers and neurons. The deep architecture in DNNs is a set of non-linear activation functions that enables the network to effectively model complex non-linear mappings from input to output [134]. DNNs have been proved to present an excellent ability to automatically learn important representations of characteristics of the data inserted in the network. So, DNNs have recently attracted much attention due to their excellent performance in phone recognition, handwritten digits recognition, face recognition, etc. Researchers began to study how to incorporate DNN in SR [142]. The appropriate numbers of layers and neuron units per layer that allow the best performance of a DNN are identified during training. In theory, more units in each layer achieve better performance in recognition tasks, according to [141]. Conversely, a too large number of units may lead to overlearning, which causes diminished performance.

Algorithms for learning such as DNN are called DL. DL refers to a branch of machine learning techniques which attempts to learn high level features from data. Since 2006, DL has become a new area of research in many applications of machine learning and signal processing. Various DL architectures have been used in speech processing [45]. In recent years, machine learning research has seen a marked switch from hand-crafted features to those that are learned from raw data, mainly due to the success of DL. DL models have become increasingly important in speech recognition, object recognition/detection, and more recently in natural language processing. Recent advances in DL have benefited from a confluence of factors, such as the availability of large-scale datasets, computational resources, and advances in both unsupervised and supervised training algorithms [23].

Convolutional Neural Network (CNN): One of the most popular DNNs is CNN. It take this name from mathematical linear operation between matrixes called con-

volution. CNN has an excellent performance in machine learning problems [5]. It has been popular in pattern recognition for non-relational data, such as images and sound processing [134]. A traditional CNN architecture consists of a sequence of layers which may include convolutional layer, non-linearity layer, pooling layer and fully-connected layer [5] [26]. The convolutional and fully-connected layers have parameters but pooling and non-linearity layers don't have parameters [5]. Basically a CNN consists of interleaved convolutional layers and pooling layers. The former layers utilize locally connected filters to share weights across the input, which enables translation invariance of the input, and the latter layers are designed to reduce the dimensionality of the data. These convolutional filters also have interpretable time and frequency meanings for audio spectrograms and are able to learn time-frequency feature representations from two dimensions [134]. According to [26] the convolutional layer in a CNN is where majority of the learning process takes place. Design and placement of the filters along the various layers of a CNN determine the "concepts" that are learned at each layer. [26] still claim that "deciding the shape of filters in CNNs is crucial to effectively learning the target concept from the input data". [85] believe that multiple layers in a deep network do not have to use the same filter shapes since they may capture different types of information. [134] highlight the importance of CNN contributions when they say that CNNs have been broadly applied in pattern recognition using many typical architectures such as VGG nets or ResNets. CNN has had ground breaking results over the past decade in a variety of fields related to pattern recognition; from image processing to voice recognition. The most beneficial aspect of CNNs is reducing the number of parameters in ANN . This achievement has prompted both researchers and developers to approach larger models in order to solve complex tasks, which was not possible with classic ANNs [5].

Residual Network (ResNet): Due to the difficulty in training DNNs, [56] presented a residual learning framework to ease the training of substantially deeper networks, called ResNet. The performance results with ResNet presented by [56] winned the 1st place on the ILSVRC 2015 classification task. It is a new CNN model called deep ResNet for image classification. The main difference between a ResNet and a typical CNN is that the second organizes the architecture in blocks combining basic units such as convolution, nonlinear mapping, pooling or batch normalization in a cascade manner. A ResNet has a shortcut pathway directly connecting the input and output of these blocks [133]. The layers of the neural network were explicitly reformulated

by [56] learning residual functions, with reference to layer inputs, instead of learning unreferenced functions. According to [56], these ResNets are easier to optimize and can gain accuracy from considerably increased depth.

Recurrent Neural Network (RNN): Another type of DNN largely used in signal processing is the RNN. RNN are composed by powerful dynamical systems that incorporate an internal memory, or hidden state, represented by a self-connected layer of neurons [18]. This property makes them well suited to model temporal sequences, such as frames in a magnitude spectrogram or chord labels in a harmonic progression, by being trained to predict the output at the next time step given the previous ones. RNNs are completely general in that in principle they can describe arbitrarily complex long-term temporal dependencies, which made them very successful in music applications. Some works perform the combination of different ANNs, such as [107] who idealized an architecture combining CNN and RNN in their ANN. According to [107] the combination occurred, because CNN is good at extracting and subsampling globe features while RNN is good at capturing the long term dependencies along a sequence, that is, capturing the past message and the current one to find its dependencies.

Deep Belief Network (DBN): Finally, other type of DNN largely used in signal processing is DBN. DBN is a generative probabilistic model composed of one visible (observed) layer and many hidden layers. Each hidden layer unit learns a statistical relationship between the units in the lower layer. The higher layer representations tend to become more complex [82]. The building block of a DBN is a probabilistic model called a Restricted Boltzmann Machine (RBM), used to represent one layer of the model [80]. RBM is a generative stochastic ANN that can learn a probability distribution over its set of inputs [2]. RBM real power emerges when RBMs are stacked to form a DBN, a generative model consisting of many layers. In a DBN, each layer comprises a set of binary or real-valued units. Two adjacent layers have a full set of connections between them, but no two units in the same layer are connected.

2.3.2 Some Concepts Used in ANN Training for SFSP

Extracting speaker's specific features of the is not an easy task. Audio-only SR systems are far from being perfect, especially under noisy conditions [24]. The presence of various environmental noises and reverberation in the input speech signal has a significant negative impact on the performance of most applications that deal with speech [112]. According

to [89], even the session variability between training and test recordings of the same speaker can heavily degrade the system performances in SV. This type of variability is usually attributed to the audio channel effects, although it also includes phonetic and intra-speaker variations such as changes of speaker's emotion, health and others. In this Chapter we discuss a few methods and techniques to feature extraction and other characteristics regarding to ANN training for SFSP.

Spectrograms: Spectrograms are images that result from the spectral content of audios [98]. They are used for training neural networks through visual feature extraction from the spectral content. A spectrogram is a bidimensional image that displays the change of frequency along the vertical axis and time along the horizontal axis. It is calculated using the Short-Time Fourier Transform (STFT) on windowed audio frames [134]. The brighter a "pixel" is, the higher the energy at this time and frequency. Spectrograms are commonly used for sound classification [85]. In spectrograms, local filters tend to capture variations within one frequency region, while global filters could capture the relationships between different harmonics and syllables [85].

Usually, a whole spectrogram is used as CNN input to obtain the global feature representation. To learn more salient features, the spectrogram is split into segments, which contain small frames along the time axis, being called a time-distributed spectrogram [134]. Each segment is a part of the spectrogram in a short interval which contains some vocalization and is labeled with one class [85]. Using these small time-distributed segments of the spectrogram as CNN input, different local features at different time steps are learned. However, spectrograms also represent the distribution of energy along the change of frequency. In many approaches, frequency-domain only features (e.g., MFCCs) also obtain good results in audio classification tasks, which have proved the importance of frequencies information [134].

Data Embeddings: In the context of ANN, embeddings are low-dimensional, learned continuous vector representations of discrete variables [58]. Data embedding is used in many machine learning applications to create low-dimensional feature representations, which preserves the structure of data points in their original space [23]. The embeddings are established by a ANN whose particular architecture allows to integrate the original data structure within the learnt representations. More precisely, considering that a Knowledge Base is defined by a set of entities and a set of relations between them, a model can learn one embedding for each entity, that is, one low dimensional vector, and one operator for each relation, that is, a matrix

[17]. It is a usual practice of reduction of dimensionality, that is to say, to project data of high dimension in a representation of low dimension, reflecting the intrinsic structure of the data and achieving a better performance in future processing.

ANN based audio embeddings (d-vectors) have seen wide-spread use in SV applications, often significantly outperforming previously state-of-the-art techniques based on i-vectors [131]. Recent advancements in DNN research have attracted speech scientists to utilize the distinctive ability of DNNs to learn and extract speaker-specific features from audio. The most common trend is to use some loss function that discriminates between speakers and extract one or more meaningful hidden layer representations, generally known as "speaker embeddings", which are then used as speaker-specific features [62]. Following steps equivalent to those described by [58], neural network embeddings can be used:

- Finding nearest neighbors in the embedding space. These can be used to make recommendations based on user interests or cluster categories;
- As input to a machine learning model for a supervised task;
- For visualization of concepts and relations between categories.

Bottleneck Features (BF): BF or simply Bottleneck (BN) are generated from a MLP or DNN in which one of the internal layers has a small number of hidden units, relative to the size of the its other layers. This special small hidden layer creates a constriction in the network to compress the task-related (classification or regression) information into a low dimensional representation. Therefore, BFs can be considered as nonlinear transformation and dimensionality reduction of the input features to a DNN [137] [83]. BFs can be derived using both unsupervised and supervised method. In unsupervised approach, classically, an autoencoder with one hidden layer trained to predicts input features themselves. In the supervised approach, works as BFs are created by an MLP or DNN trained to predict the class label, for example, phonemes or phoneme states [83].

BFs have been used in many studies for SR [118] [105] [95], speech recognition [137], language identification [93] and acustic event recognition [101]. BFs in many studies are obtained from traditional feature extraction methods, such as MFCC or PLP [43] [114], but have also been combined with features from these traditional methods to be used as DNNs input, showing improvements in its performance in the tasks of recognition using audios [137] [95] [83]. [118] explore methods to further improve BFs and obtain experimental results that show that the exploration of phonetic

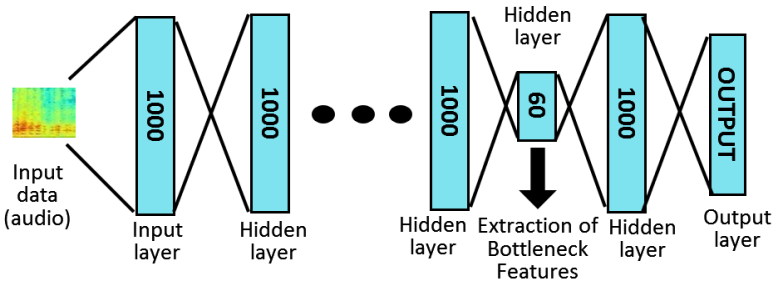


Figura 2.5: Representation of a DNN with BFs extraction layer

information encoded in BFs obtains further improvements to speaker embeddings systems. It is believed that because the BFs capture information complementary to the conventional features derived from the audio spectra [137], that such information is more effective in the recognition tasks, however it is necessary to preserve the original input features [83]. Figure 2.5 shows a representation of a DNN that extracts BFs through one of its hidden layers.

2.3.3 Steps to training ANNs for SFSP

After becoming familiar with these SFSP tasks, we are going forward to understand how they are executed. We observed that SFSP systems follow a very similar execution process. Some differences can be considered as peculiarities within the execution process of each of them. We also note that there is a lack of standardization in the nomenclatures used by the authors. In order to clarify the basic steps that make up the SFSPs executions, we represent these steps in Figure 2.6, which shows a generic SFSP process execution model. To create this SFSP process execution generic model, we analyzed the works that explain the step by step they used to execute their SFSP process. According to [66], conventional SV systems are normally composed by following four stages: pre-processing, acoustic feature extraction, speaker feature extraction and binary classification. [127] state that biometric identification based on the human voice consists of three main steps. The process begins with the biometric sample (recording of speech), followed by a speech processing (features extraction). The final step is the classification of the speaker. For [64], both the SV and SI system consist of three essential elements: SFE, SM and speaker matching. SFE concerns to extracting essential features from an input speech for SR; and SM concerns to probabilistically modeling the feature of the enrolled speakers. [77] draws attention to the need for the existence of a step before the ANN modeling so that a treatment is performed on the input data to highlight the information of interest to the

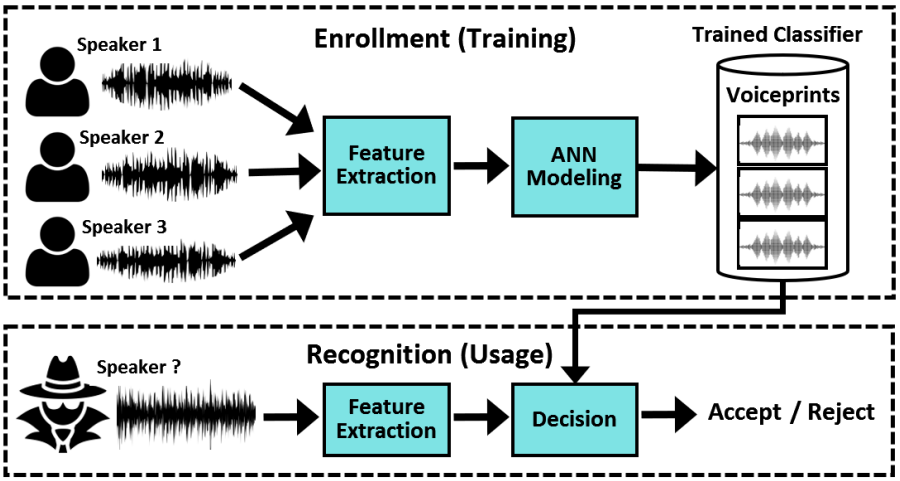


Figura 2.6: Generic SFSP Process Execution Model

system when it says that "the training process requires a large number of samples and the characteristic specificity is not obvious, which results the DL cannot work directly". Reinforcing [64] point of view, [103] mentions that there are three basic steps involved in SI:

- SFE: This step converts raw speech signals into a set of feature vectors. There are different types of features that can be extracted from the speech and depending on the choice, the accuracy of recognition varies. Some feature extraction algorithms available are MFCC, PNCC (Power Normalized Cepstral Coefficients), LPC (Linear Prediction Coefficients), PLP (Perceptual Linear Prediction), RASTA PLP (Relative Spectral PLP), IHC (Inner Haircell Coefficients) etc.
- SM: The extracted features are used to generate models corresponding to each speaker and stored in specific representations to perform comparison during testing. Different methods available are GMM (Gaussian Mixture Models), DNN, HMM (Hidden Markov Models), i-vector method, among others.
- SI: The final stage involves the classification of the test speech signal. Relative scores are computed for each of the speaker models and the one with the highest score is identified to be the test speaker.

Based on these definitions, we developed Figure 2.6, which shows the basic steps for training and using an ANN for SFSP. We can divide the process in two steps: Enrollment and Recognition. In Enrollment, an audio dataset containing many speakers samples is utilized in the ANN training. First, features are extracted from this samples. This features are then utilized to ANN training during the ANN Modeling execution. Finally, a ANN

Tabela 2.10: Result of QA in studies focusing on SM

Pos	Cod	Prof.	QA1	QA2	QA3	QA4	QA5	QA6	QA7	QA8	QA9	QA10	Final Grade	Final Grade Percent.
1	S2	SI	9	9	4	12	12	6	4	2	2	1	61	96.83%
2	S3	SR	6	9	4	12	12	6	4	2	2	1	56	88.89%
3	S1	SI	9	9	4	12	6	3	4	2	2	1	52	82.54%
3	S8	SS	9	9	4	12	3	6	2	2	4	1	52	82.54%
4	S7	SSD	3	9	4	12	12	3	4	2	0	1	50	79.37%
5	S30	SV	3	9	4	12	6	6	4	2	2	1	49	77.78%
6	S4	SV	6	9	4	8	9	3	2	2	2	1	46	73.02%
6	S5	SR	6	6	4	8	9	6	4	2	0	1	46	73.02%
7	S9	SR	9	6	4	8	6	6	4	0	0	1	44	69.84%
8	S10	SV	6	9	2	8	3	6	4	2	2	1	43	68.25%
8	S33	SD	6	6	4	8	6	6	4	2	0	1	43	68.25%
9	S34	SV	6	6	4	4	9	6	4	2	0	1	42	66.67%
9	S6	SSD	0	9	4	8	3	6	4	2	0	1	37	58.73%
10	S32	SD	9	3	4	4	6	3	4	2	0	1	36	57.14%
11	S11	SI	6	6	2	0	6	6	4	2	2	1	35	55.56%
11	S12	SR	6	9	4	0	3	6	4	0	2	1	35	55.56%
12	S15	SSD	0	6	4	0	6	6	2	0	2	1	27	42.86%
12	S17	SI	9	6	4	0	3	0	2	2	0	1	27	42.86%
13	S16	SI	3	3	4	4	6	0	2	2	0	0	24	38.10%
14	S14	SI	0	9	4	0	3	0	2	2	0	1	21	33.33%
15	S18	SI	0	9	4	0	3	0	2	0	0	1	19	30.16%
15	S19	SR	3	9	2	0	3	0	2	0	0	0	19	30.16%

model is obtained where the speakers voiceprints are registred to usage in recognition process. Recognition step represents the ANN model usage, where unknowned speakers will be evaluated by the ANN model trained. First, features has to be extracted like in Enrollment step and the recognition Decision is evaluated by the ANN model.

2.4 Answering Our Research Questions

This Chapter presents the results of the literature analysis carried out using the data extracted from the selected studies and the results of the QA. The QA was applied to each of the 34 selected studies and produced a total of 35 results. After assigning grades for each of the 10 QA questions, a final grade was calculated for each study. Tables 2.10 and 2.11 list the results of the QA, showing the studies classified by their final grades in descending order. Studies that obtained a valid grade in the QA9 presented a justification, as requested. Table 2.12 shows the justifications for these studies. In what follows, we describe the answers for our RQs for the selected studies.

Tabela 2.11: Result of QA in studies focusing on SFE

Pos	Cod	Prof.	QA1	QA2	QA3	QA4	QA5	QA6	QA7	QA8	QA9	QA10	Final Grade	Final Grade Percent.
1	S28	SS	6	9	4	8	12	6	4	2	2	1	54	85.71%
2	S25	SV	6	9	4	4	12	6	4	2	2	1	50	79.37%
3	S27	SI	6	9	4	8	9	6	4	0	2	1	49	77.78%
4	S31	SD	6	9	4	8	9	3	4	2	2	1	48	76.19%
5	S20	SV	9	9	4	8	6	6	4	0	0	1	47	74.60%
5	S21	SR	6	9	4	12	6	3	2	2	2	1	47	74.60%
5	S22	SA	9	3	4	8	6	6	4	2	4	1	47	74.60%
5	S29	SD	6	9	4	12	6	3	4	2	0	1	47	74.60%
6	S24	SV	9	6	4	8	6	6	4	0	2	1	46	73.02%
7	S11	SI	6	9	4	4	6	6	4	2	2	1	44	69.84%
8	S13	SV	3	9	4	12	6	0	0	0	0	0	34	53.97%
9	S23	SV	3	9	4	4	6	0	2	2	0	1	31	49.21%
10	S26	SV	6	3	0	0	6	3	2	2	2	1	27	42.86%

2.4.1 RQ1: What are the SFSP areas found and the percentages of the selected studies that work in each one of them?

Table 2.13 shows the SFSP areas identified in the SM and SFE studies. The area with the largest number of studies was SV, followed by SI and SR. Analyzing each focus of approach, SV appears with the greatest number of studies of SFE, with a difference much greater than the other SFSP areas that have practically equal amounts. With regard to the SM focus, SI area appears in first place. The other SFSP areas have smaller amounts of studies, but there is no big difference between them. No SM works were found in the SA area, nor were there SFE works in the SSD area. Finally, it can be seen that there is a preference in research with SV and SI areas, since they correspond to 28% of the SFSP areas (in total 7), but they represent more than half of the works analyzed with 54.29%.

2.4.2 RQ2: What types of ANN architectures with the best performance have been used by studies focusing on SM and SFE?

Table 2.14 shows 18 types of identified ANN architetures. Of these 18, 11 were found in the SM studies and also 11 in the SFE studies. Only 4 types of ANN architectures were common to the SM and SFE approaches: RNN, DNN, CNN and CNN-RNN. Traditional DNN appears with the highest number in both SM and SFE approaches and in total it was used in 22.86% of the works. If we consider the use of DNN also with the participation of other networks, we count 8 types of architectures using DNN and 42.88% of the analyzed

Tabela 2.12: Justifications for QA9

Cod	Profile	AQ9 Grade	Justifications for QA9
S1	SI	2	Specializing in noisy audios and sub and speaker supra-glottal features.
S2	SI	2	The network specializes in SI and an attention mechanism is introduced to systematically improve the features of certain frequency bands: the FreqCNN.
S4	SV	2	Neural network specialized in short duration audio (5 seconds).
S8	SS	4	a) It involves unsupervised training and, therefore, can be trained on real life data, in which the audio streams have multiple speakers with change points from unknown speakers. This makes it highly scalable. b) Learn embeddings from short-term speakers, which can be useful for applications like SS. c) GRUs exploit sequential information in MFCC frames and help to learn speaker links faster than CNNs, even with fewer parameters. d) The implementation of a Siamese network reduces the number of parameters compared to the dense architecture.
S10	SV	2	SV in noisy audios. It takes a new approach to solve the problem of separating audio signals from speech that are mixed with other sounds in the original audio. In the first stage, an CNN determines the presence or absence of the speech signal in a noisy scene. In the second stage, other neural networks filter the speech signal determined in the first stage.
S11	SI	2	It highlights a problem of incompatibility between microphone and telephone channels affecting SI systems based on DNN and proposes the method called Average and Normalization of Variance showing improvement with its application. The method is not shown in detail.
S12	SR	2	Study of application on Human-Robot Interactions and Internet of Things devices using dialect regions in America and English, French and German's multilingual speeches.
S15	SSD	2	Utterances were used as input for three synthesizers to create the spoofing utterances. The study create spoofing utterances to be tested on SV systems to verify the ability to break the protection of these systems through the SV functionality.
S21	SR	2	Application of Stacked BFs in DNNs as a technique to improve the features extraction from the speaker. It explored the possibilities of improving the performance of the speaker embeddings system by employing phonetically rich BF for training RNPs.
S22	SA	4	Proposes a feature extractor based on DNN. An DNN is trained to produce normalized pseudo-features of the speakers (FMLLR) from 3 types of extractors with non-normalized features (filterbank, MFCC and LDA). In order to achieve this, in its training, DNN receives non-normalized features at its entrance, having as target, at its exit, the corresponding normalized features. In the end it learns to produce the normalized pseudo-features. These features are later used for acoustic modeling in another DNN (the study does not show details of this DNN). The best performance was achieved using Cepstral Mean Normalization (CMN) for normalization and the combination of FMLLR + VTLN (Vocal Tract Length Normalization) features as a target for learning the network.
S24	SV	2	Proposes obtaining deep features through a multi-task neural network to absorb features of the speaker and speech (phoneme or phrase) and to execute some method on these features to generate the acoustic model to identify the speaker and the speech.
S25	SV	2	Proposes a DNN-based technique to train non-linear mapping of i-vectors representing short utterances from a long audio version, in order to improve the performance of the evaluation on the short utterances.
S26	SV	4	Fusion approach at the level of speech and physiological features, combining acoustic (speech) and articulatory (estimated) information from the speakers for verification tasks of independent and text-dependent speakers. The tract variables include nine articulatory parameters such as lip aperture, lip protrusion, jaw opening, the constriction degree and constriction location of tongue tip, tongue blade and tongue dorsum.
S27	SI	2	Network specializing in long-distance speech and provides a mechanism for speech dereverberation with noise.
S28	SS	2	Presents a neural network to prove the assumption that two audio clips close to the audio refer to the same speaker.
S30	SV	2	The study presents a new dataset containing more than 6,000 celebrities worldwide, of different ethnicities and languages, and a large number of videos and audios.
S31	SD	2	Telephone conversations in six languages: Arabic, English, German, Japanese, Mandarin and Spanish. It also implements a representation of speaker embeddings, called d-vector, which the study shows to perform better than i-vector.

Tabela 2.13: Number of studies selected by SFSP Areas

SFSP Areas	SM		SFE		Total	
	Amount	Percent.	Amount	Percent.	Amount	Percent.
SV	4	18.18%	6	46.15%	10	28.57%
SI	7	31.82%	2	15.39%	9	25.72%
SR	5	22.73%	1	7.69%	6	17.14%
SD	2	9.09%	2	15.39%	4	11.43%
SSD	3	13.64%	0		3	8.57%
SS	1	4.54%	1	7.69%	2	5.71%
SA	0		1	7.69%	1	2.86%
TOTAL	22	100.00%	13	100.00%	35	**100.00%**

works. Of the 18 types of networks, 7 are architectures made up of more than one neural network. The wide variety of ANN architectures can perhaps be explained by the variety of types of research in the studies. As each type of ANN presents specific actions on different types of approaches, the combination between them further expands the possibilities of specialized actions. 3 types of ANN used the BF technique. All 3 were from studies focused on SFE. The use of BN assists in the specific selection of the speaker features in order to improve the performance of the speaker modeling neural networks. Proportionally, there is a greater variety of architectures in the SFE works because the 13 works used 11 different types of architectures, corresponding to 84.62% difference in architectures, while in the SM focus the 22 works also used 11 different types of architectures, but corresponding to 50.00%. We think that a justification for the greater variety of types of neural networks related to SFE works is the great diversity of audio scenarios analyzed by these works. Each of these audio scenarios explores very different conditions in their datasets and possibly the need to explore specific situations when extracting speaker features is the cause of the large number of different neural network architectures.

2.4.3 RQ3: About conventional feature extraction methods utilization:

- **RQ3.1: What conventional feature extraction methods were used feeding SM ANN?** In the 22 SM studies, 12 different types of conventional feature extraction methods were identified, totaling 27 uses. Table 2.15 lists the methods identified and the number of uses in the studies. The purpose of these methods is to read and extract the speaker features existing in the audios so that they feed the SM ANNs. It is possible to observe a very large use of the traditional MFCC method, corresponding to 40.74% of the 27 uses. 3 studies did not inform the extraction

Tabela 2.14: Number of studies selected by types of ANN Architectures

Types of ANN Architectures	SM		SFE		Total	
	Amount	**Percent.**	**Amount**	**Percent.**	**Amount**	**Percent.**
DNN	5	22.73%	3	23.08%	8	22.86%
ANN	3	13.64%	0		3	8.57%
MLP	3	13.64%	0		3	8.57%
CNN	2	9.09%	1	7.69%	3	8.57%
CNN-RNN	2	9.09%	1	7.69%	3	8.57%
CNN(ResNet)	2	9.09%	0		2	5.71%
RNN	1	4.55%	1	7.69%	2	5.71%
1-D CNN-RNN	1	4.55%	0		1	2.86%
1-D CNN	1	4.55%	0		1	2.86%
DNN-CNN-RNN	1	4.55%	0		1	2.86%
DBN-DNN	1	4.55%	0		1	2.86%
DNN(ResNet)	0		1	7.69%	1	2.86%
DNN(BN)-DNN(BN)	0		1	7.69%	1	2.86%
DNN-DNN(BN)	0		1	7.69%	1	2.86%
DNN(BN)	0		1	7.69%	1	2.86%
CDBN	0		1	7.69%	1	2.86%
DBN	0		1	7.69%	1	2.86%
DNN-ANN	0		1	7.69%	1	2,86%
TOTAL	22	100.00%	13	100.00%	35	**100.00%**

methods but reported that the ANN was fed with spectrograms. The second most used method was the STFT, 3 times, quite distant from the MFCC that was used by 11 studies. The other methods had only 1 use.

- **RQ3.2: What conventional feature extraction methods were used as a baseline for comparison with SFE ANN or feeding SFE ANN?** For studies focusing on SFE, the conventional feature extraction methods were used in two ways, acting as: baseline for performance comparison with the SFE ANNs proposed by the studies and as a first step in the feature extraction, feeding the SFE ANNs. Table 16 lists the methods identified and the number of uses in these two scenarios. There is an absolute preference in the use of the MFCC method in these two scenarios in relation to the other 7 identified methods. Its use was accounted for in 60% of studies as a baseline for comparison with ANN and 46.67% as a method of feeding ANN. Another 4 methods are also used in the two scenarios, but with much less use than the MFCC. The result of this analysis carried out by RQ3 shows that the MFCC method had an absolute preference in surveys related to SM and SFE.

Tabela 2.15: Conventional Feature Extractors that feed SM ANNs.

Conventional Feature Extraction Method	**Amount**	**Percent.**
MFCC	11	40.74%
Short Time Fourier Transform (STFT)	3	11.11%
Not informed	3	11.11%
log filterbank energies	1	3.70%
pcaDCT	1	3.70%
raw-waveform feature	1	3.70%
RNP(BN)	1	3.70%
Unique mapped real transform (UMRT)	1	3.70%
Feature space Maximum Likelihood Linear Regression (fMLLR)	1	3.70%
Perceptual Linear Predictive (PLP)	1	3.70%
Holoentropy with the eXtended Linear Prediction using autocorrelation Snapshot (HXLPS) + i-vector	1	3.70%
Frequency Filtering (FF)	1	3.70%
Fast Fourier Transform	1	3.70%
TOTAL	27	100.00%

Tabela 2.16: Feature Extractors acting as: baseline for performance comparison with proposed SFE ANN and conventional method of extraction to feed the proposed SFE ANN.

Conventional Feature Extraction Method	**Comparing performance with SFE ANN (baseline)**		**Feeding the SFE ANN**	
	Amount	**Percent.**	**Amount**	**Percent.**
MFCC	9	60.00%	7	46.67%
pre-emphasis	0		1	6.67%
log mel filterbank energies	2	13.33%	2	13.33%
log filterbank energies	0		1	6.67%
LDA	1	6.67%	1	6.67%
PLP	1	6.67%	1	6.67%
STFT	1	6.67%	1	6.67%
pcaDCT	1	6.67%	0	
Not informed	0		1	6.67%
TOTAL	15	100.00%	15	100.00%

Tabela 2.17: Main metrics used by SFSP areas.

Metric	SI	SR	SV	SD	SS	SA	SSD	Total
Equal Error Rate (EER)	2	4	8		1		3	18
Accuracy	4	2	2				1	9
minimum Detection Cost Function (minDCF)		1	4					5
Cost primary	2		1					3
Detection Error Tradeoff (DET)		1	2					3
minimun Cost Detection Error Tradeoff (minCDET)		1	1					2
Recognition Rate (RR)	2							2
Diarization Error Rates (DER)				2				2
Word Error Rate						1		1
Phone Error Rate						1		1
Other metrics (Total: 21)	2	4	2	7	3	1	5	24
TOTAL	12	13	20	9	4	3	9	70

2.4.4 RQ4: What were the main metrics used by each SFSP areas?

Table 2.17 shows the accounting for all 31 metrics used by the studies. The 10 main metrics, most used or which had the most highlights in the studies, were listed, and the other 21 least used were grouped. RQ4 was developed because we assume that there would be a trend in the use of certain metrics for each SFSP area. But this trend was not identified due to the wide variety of metrics used by each one. The SR area was the one with the highest number of different metric types, 9 in total, followed by: SV with 8, SD with 8, SSD with 7, SI with 6, SS with 4 and SA with 3. The SS and SA areas show a smaller number of types of metrics that appear to be a trend. But this is probably not true because both have a very small number of selected studies, respectively 2 and 1 studies, if compared to the number of studies in the other SFSP areas. What can also be seen in Table 2.17 is that the EER metric is the one with the greatest number of uses, being used in 18 studies, corresponding to more than 25%. It is also the one with the highest usage per SFSP area, being used in 5 of the 7 areas. It comes followed by Accuracy, which has 9 uses in studies. The 5 most used metrics together account for 38 uses. They correspond to 16% of the metrics and 54% of the uses in the studies.

2.4.5 RQ5: What were the characteristics explored by the studies in the audio datasets?

A wide variety of audio characteristics explored by the studies were identified, 21 in total. The list of these characteristics is shown below:

1. Speaker physiological features;

2. Degraded audios;
3. Noise audios;
4. Different types of speech;
5. Speakers of different ethnicities or nationalities;
6. Speakers with different accents of the same language;
7. Speakers from different professions;
8. Speakers of different ages;
9. Bilingual or trilingual speakers;
10. Short speeches;
11. Speech from long distances;
12. Unique speakers;
13. Multi-speakers;
14. Real life conditions;
15. Audio falsification categories;
16. Telephone or microphone conversations;
17. Speakers of the same language, without any specific approach;
18. Combined with other media or other representations of the individual's features;
19. Phonemes, letters or words;
20. Related to stories or movies;
21. Closed places.

This large and distinct number of characteristics that the datasets address shows a very interesting current situation of research with ANNs for SFSP. This list helps to show how complex the process of audio feature extraction is so that later the ANN modeling can happen. The distinct features extracted from such audio characteristics show the need for extractions of specific features according to the objective of the study. We can conclude that the generalization of an ANN model proves to be very complex today due to the varied situations, audio characteristics and speaker individual features, which can be represented in each audio.

Tables 2.18 and 2.19 show, respectively, in which SM and SFE studies the 21 audio characteristics were covered. It can be seen that the study focusing on SM that explored different audio characteristics the most was S3, with 5. Second is S17, which addresses 4 types of audio characteristics. The most explored characteristic in studies focusing on SM was number 12, used by 6 studies and which presents the simplest approach with the use of unique speakers. In second place are characteristics 17 and 19, with 4 uses and that

Tabela 2.18: The various audio characteristics explored by studies focusing on SM.

Cod	Audio Characteristics Explored by SM Studies																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
S1	X	X																			
S2				X																	
S3					X	X	X	X	X												
S4										X											
S5												X	X	X							
S6															X						
S7															X						
S8													X	X							
S9																X					
S10			X																		
S11																X					
S12						X			X												
S14												X					X				
S15													X		X						
S16												X							X		
S17	X											X						X	X		
S18												X							X		
S19												X							X		
S30					X																
S32																	X				
S33																	X				
S34																X	X				

address, respectively, speakers of the same language and learning about phonemes, letters or words. For the SFE focus, the S28 study was the one that addressed the largest number of different characteristics, with 5. Next are studies S13, S27 and S31, which each used 3 types of audio characteristics. And the characteristics most used by the SFE studies were number 3, 12, 16 and 17, used in 3 studies each. Next, the most used characteristics were: 2, 5, 14, 19 and 20, used in 2 studies each.

2.4.6 RQ6: Which studies had QA with a grade higher than 75% of the maximum grade?

Given the RQ6, in this SLR phase, the studies that had the best QA (grade greater than 75% of the maximum grade) are registered in the tables 2.20 and 2.21. Six SM studies and four SFE studies were selected, which should have a summary report on the ANN architecture, audio aspects and performance measurements covered by their research.

A brief analysis of the studies with the best QAs are described in what follows. They are divided in SM and SFE studies.

Tabela 2.19: The various audio characteristics explored by studies focusing on SFE.

Cod	Audio Characteristics Explored by SFE Studies																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
S11																X					
S13												X					X		X		
S20										X											
S21		X	X																		
S22						X										X					
S23												X								X	
S24																	X		X		
S25														X							
S26	X																	X			
S27		X	X								X										
S28			X		X							X	X							X	
S29														X							X
S31					X											X	X				

Tabela 2.20: SM Studies selected by RQ6

Pos	Cod	Study	Profile	Library	Grade	Percent
1	S2	Audio classification using attention-augmented convolutional neural network	SI	Scopus	61	96.83%
2	S3	Discriminative deep audio feature embedding for speaker recognition in the wild	SR	Engineering Village	56	88.89%
3	S1	Extracting sub-glottal and Supra-glottal features from MFCC using convolutional neural networks for speaker identification in degraded audio signals	SI	Engineering Village	52	82.54%
3	S8	An unsupervised neural prediction framework for learning speaker embeddings using recurrent neural networks	SS	Engineering Village	52	82.54%
4	S7	Investigating Raw Wave Deep Neural Networks for End-to-End Speaker Spoofing Detection	SSD	Engineering Village	50	79.37%
5	S30	VoxceleB2: Deep speaker recognition	SV	Engineering Village	49	77.78%

Tabela 2.21: SFE Studies selected by RQ6

Pos	Cod	Study	Profile	Library	Grade	Percent
1	S28	Speaker2Vec: Unsupervised learning and adaptation of a speaker manifold using deep neural networks with an evaluation on speaker segmentation	SS	Scopus	54	85.71%
2	S25	Deep neural network based i-vector mapping for speaker verification using short utterances	SV	ScienceDirect	50	79.37%
3	S27	Deep neural network-based bottleneck feature and denoising autoencoder-based dereverberation for distant-talking speaker identification	SI	Engineering Village	49	77.78%
4	S31	Speaker diarization with LSTM	SD	Engineering Village	48	76.19%

SM – S2: [134] present an CNN architecture containing 3 layers called “Basic convolutional block” followed each by 1 convolution block and a function max pooling 2x2, stride 2. After that, 2 layers called “Attention-based block” are executed, which are also followed, each one, by 1 convolution block and max pooling 2x2, stride 2. The CNN ends with a fully connected layer followed by a normalization layer softmax. Each “Basic convolutional block"layer has a grouping that brings together other layers of convolution and concatenation and addition operations. Each “Attention-based block” layer has convolution layers, layers containing attention method and addition operation. The audio signal is represented by spectrograms to feed the neural network. The study presents a versatility neural network application, because in addition to the use for SI, it is also used for the speakers’ accents classification and emotion recognition in speech. For training, the CHAINS dataset was used, containing the following speech audios: solo (one person speaking), synchronous, retell, RSI (Repetitive Synchronous Imitation), whisper and fast. The results obtained by the proposed neural network demonstrate superior performance to the state-of-the-art. Using the Accuracy and the Unweighted Average Recall (UAR) as metrics, the CNN, called FreqCNN, had an average performance of 98.04% and 98.05%, respectively. The performances of the other neural networks for Accuracy and UAR were: VGG-11 neural networks with 75.21% and 75.11%, ResNet-18 with 75.04% and 74.99%, ResNet-34 with 66.05% and 66.00%, and ResNet-50 with 66.95% and 66.75%.

SM – S3: [14] present a 34 layers CNN(ResNet) architecture of residual convolution (ResNet-34), inspired by the neural network VoxCeleb VGG-M [102], containing 3 layers max pooling and 3 fully connected layers. At the end, normalization is performed by softmax. The audio signals are extracted by the conventional STFT method and represented in the form of a spectrogram for feeding the neural network. The study aims to act in a multilingual scenario. The focus of the study was on the SR problem in real life. For this purpose, the following datasets were used: VoxCeleb, containing celebrities of various ethnicities with use of English, with accents and in different conditions of the real world; and SIWIS, presenting a cross-language scenario, that is, containing bilingual and trilingual speakers using English, French, German and Italian. As it presents an SR research area, the study performs tests on SI and SV and experimental results show the effectiveness of the proposed solution in relation to the state-of-the-art. In SI tests, ResNet-34 performed better than the other solutions using the VoxCeleb dataset, obtaining the accuracy percentages of

85.2% and 93% in top 1 and top 5, surpassing the other version proposed by the study, ResNet-18, and the other compared methods: VoxCeleb RNC, i-vector / Probabilistic Linear Discriminant Analysis(PLDA) / Support Vector Machine (SVM) and i-vector / SVM. In the SV tests, using the EER metric, ResNet-34 obtained a value of 5.2%, surpassing all other competitors with the dataset SIWIS, but in the VoxCeleb dataset, it remained with 8.5% , being surpassed only by the VoxCeleb-256D embedding method which obtained 7.8%.

SM – S1: [26] created a CNN architecture based on 1D filters. The network contains 3 convolution layers and 1 fully connected layer. In conjunction with the first 2 convolution layers, the network presents a layer containing an activation function Rectified Linear Unit (ReLU) followed by a function max pooling 2x1, stride 2x1. After the convolution layers, it also has 2 layers of dropout containing a ReLU layer between them. The CNN is finished with a normalization layer softmax for 168 classes. The MFCC method was used to represent the audio signals that feed the network with 1D representation. The study proposes a specialization in SI based on degraded audios. The learning focus is directed to the sub-global and supra-global features of each speaker. Such features belong to the human speech production apparatus. The research used training with the following datasets: TIMIT, containing clean speech recordings of the eight main American English dialects with increased noise; NTIMIT, containing TIMIT data retransmitted and captured by telephone; SITW, containing speech samples collected from open source media; and Fisher, containing telephone conversations between pairs of people. [26] say that their proposed solution is compared with the existing baseline schemes with regard to research on synthetic and naturally degenerate speech data. A comparative test was performed between the 1D-CNN and the Universal Background Model with Gaussian Mixture Model (UBM-GMM) and i-vector / PLDA models using the TIMIT dataset. This dataset contained the voices of 168 speakers incorporating noise from: conversations, F-16 fighters, cars and factories. Later, the voices of 1052 people from the Fisher English Training Speech Part 1 dataset were added to the training. When evaluating the performance of the 3 methods, it was found that the 1-D CNN far exceeded the other 2 methods. The 1-D CNN presented 13.78% and 51.98% representing the worst and the best accuracy percentage in top 1 and 35.31% and 72.42% in top 5. The numbers reached by the other methods as worst and best results were 0% and 16.56% for UBM-GMM, and 0.19% and 6.54% for i-vector-PLDA at top 1; and 0% and 26.19% for UBM-GMM, and 0.39% and 19.14%

for i-vector-PLDA at top 5.

SM – S8: [62] study presents an Siamese networks architecture having two identical RNNs, whose weight values are shared. Each Siamese twin network contains 3 RNN-GRU layers with 200 hidden units in each layer. The authors report that the preference for GRU over LSTM is due to the smaller number of parameters. Each Siamese network produces a embedding of dimension 512. A calculation is performed with the embeddings provided by the two GRU networks producing a result of the same dimension 512, which will be the entrance of a fully connected layer. The network single output is made through a non-linear sigmoidal function that predicts if the pair of audio segments used as network input, is genuine (0) or imposter (1). The Feature Extraction is done with 40-dimensional high definition MFCC method, calculated with a sliding window of 25ms frame length and 10ms frame shift, using the Kaldi toolbox. The study aim is to present an unsupervised training framework for learning speaker-specific embeddings using a Neural Predictive Coding technique. The network is trained in unlabeled audio with multiple and unknown speaker exchange points. The speakers' short-term stationarity is assumed, that is, the speech frames with temporal proximity come from a single speaker. On the other hand, it is assumed that two random segments of short speech from different audio streams originate from two different speakers. Based on this hypothesis, a binary classification scenario is developed to predict whether a pair of short speech segments comes from the same speaker or not. A deep Siamese network based on RNN is trained and the resulting embeddings, extracted from a hidden layer representation of the network, are employed as the speaker embeddings. The datasets used were YoUSC-Tube and TED-LIUM for training and tests, respectively. In the validation, audio segments with duration between 1s and 3s were randomly created. As baseline, a speaker change detection algorithm based on the BIC metric was used. Two types of metrics were used to evaluate the results: F1 score, which is based on the harmonic mean of precision and recall; and the coverage and purity metrics. The proposed solution exceeds the baseline by scoring 0.86 in F1 score against 0.74 and (0.88, 0.86) in (coverage, purity), against (0.92, 0.75).

SM – S7: [33] propose a neural network architecture in study S7 that combines CNN, RNN (LSTM) and DNN and is therefore called CLDNN. There are 3 convolution blocks in the 3 initial layers, each composed of a convolution layer, a normalization batch layer and a ReLU activation layer. Then there is a RNN-LSTM layer and at the end there are 2 fully connected DNN layers. The last layer is a linear activation.

The audio signals are represented in raw waveform features form to feed the neural network. The study is concerned with the security of SV systems in real-world applications and proposes a neural network specialized in SSD that surpasses previous attempts that used the BTAS2016 dataset (0.19% Human-targeted Translation Edit Rate), positioning itself as the current state-of-the-art model for such a dataset. The proposed model is able to distinguish spoofing attempts regardless of the device, exploring the spoofing methods: repetition, speech synthesis and voice conversation. Regarding the ASVspoof2015 dataset, the proposed end-to-end solution achieves an error rate of 0.00% using the Equal Error Rate (EER) metric for the S1 to S5 spoofing speech detection. BTAS2016 and AVSpoof2015 datasets were used for training the neural network. The proposed architecture achieves the best results on the BTAS2016 basis, with 0.189 under the HTER metric (Human-targeted Translation Edit Rate) and 0.171 under ERR.

SM – S30: [28] draws attention to the two main contributions of the S30 study: first, the creation of a new dataset called "VoxCeleb2"that gathers large-scale audiovisual data collected from YouTube. It contains over a million statements from over 6,000 celebrities. Secondly, the development and comparison of CNN models for recognizing voice identity under various conditions, using the VoxCeleb2 dataset. The architecture presented for best performance is an CNN(ResNet), called by the authors ResNet-50, which contains 50 convolution layers and has in its final block 1 fully connected layer, followed by a average pool layer and ends with a fully connected layer, rated for 5,994 classes. The network is fed by audio signals in the spectrogram form. The VoxCeleb1 dataset was used for the baseline methods and VoxCeleb2 for the methods proposed by the study. The performance analysis uses two metrics for the evaluation: EER and a cost function. The ResNet-50 method surpassed all methods obtaining 0.429 in the cost function and 3.95% in EER against the following respective values for other methods: 0.549 and 4.83% of ResNet-34, 0.609 and 5.94% of VGG-M and 0.73 and 8.8% of i-vector + PLDA.

SFE – S28: [61] constructed a neural network to act as an automatic encoder with *(2k + 1)* hidden layers, where the *(k + 1)* layer is the one that provides, through BN, the speaker feature representations. The best performing network operated with 5 layers *(k = 2)*. At the entrance of the neural network, audio vectors containing MFCC features were used. Sequential vector segments of audio features, *w1* and *w2*, are created with the size of *d* frames each. The construction of each pair (*w1*

and *w2*) is separated by Δ frames. This Δ distance is measured from the end of segment *w2*, of the previous pair, to the end of segment *w2*, of the next pair. In this experiment, the Δ distance is not so great, allowing an intersection between the pairs, that is, there are enough repetitions of the speaker features between sequential pairs. Each pair becomes a training sample (input and output) for the automatic encoder. For each pair, *w1* goes to the autoencoder input layer and *w2* goes to the output. The automatic encoder tries to reconstruct *w2* using *w1*, minimizing a loss function through training. According to [61], this structure allows you to explore a longer context to capture speaker features and compress them into a smaller vector representation. The conventional method of Feature Extraction used for reading the audio signal is the MFCC, which feeds the neural network. A proposal for a new method to learn how to collect speaker features in an unsupervised manner is presented. They begin from the assumption of active speaker stationarity in a audio short time and then derive a speaker representation using DNN. They assume that temporally close speech segments belong to the same speaker and, as such, a joint representation, connecting these close segments, can encode their common information. Thus, this BN representation will mainly be capturing specific speaker information. The authors promote the method saying that it does not need to be supervised, does not need to create labels for the training samples and does not need to use VAD (Voice Activity Detection) resources. The proposed representation is presented as having the possibility of being used in different applications, such as SD and SI, but the authors present only their evaluation tests using on SS (which is the identification of different speaker sections in an audio). [61] used the following datasets for training: TED-LIUM, mostly with audio from just one speaker; and Youtube, containing monologues by a single speaker, discussions between several speakers, films, speeches with background music inside and outside studios, audios with different languages including English, Spanish, Hindi and Telugu. During the evaluation, the datasets used were TED-LIUM, NIST RT-06 (data from conference meetings) and Couples Therapy Corpus (which present spontaneous conversations between husband and wife). They demonstrated in the study that the proposed method surpasses state-of-the-art SS algorithms and the baseline methods based on the MFCC. Its performance, using the F1 score measurement method, registered 0.86 (TED-LIUM), 0.85 (YouTube) and 0.85 (YouTube large), compared to 0.73, 0.78, 0.74 and 0.79, which are performances of 4 works considered until then to be the state-of-the-art. Such works used in their performance evaluations artificial

dialogs created from the TIMIT dataset.

SFE – S25: [50] analyzed the problem of performance drop in SV systems when processing short audios. They show how the performance of solutions based i-vector SV systems degrade rapidly as the duration of the evaluation utterances decreases. To address this issue, they propose two novel nonlinear mapping methods which train DNN models to map the i-vectors extracted from short utterances to their corresponding long-utterance i-vectors, in order to improve the short-utterance evaluation performance. Both proposed solutions model the joint representation of short and long utterance i-vectors by using an autoencoder. The mapped i-vector can restore missing information and reduce the variance of the original short-utterance i-vectors. After tests with other methods considered state-of-the-art, all using i-vectors, the proposed DNNs achieves best results. It is a DNN with 6 hidden layers using residual blocks. Each residual block consists of two fully-connected layers and a short-cut connection that performs a summation between the entrance of the first layer to the end of the second layer. The authors also carry out further training with concatenation of phonemes to short audios. The audio signals are represented by 40 mel-filterbank features with an audio frame length of 25ms. The datasets used were NIST SRE 2010 and Speaker In The Wild (SITW). Both DNNs proposed provide significant improvement and result in a 24.51% relative improvement in EER from a baseline system. The addition of residual blocks improved performance to 26.47% compared to baseline. And the addition of phonemes improved even more, reaching 28.43% in relation to baseline.

SFE – S27: [141] carry out a research to SI by applying methods for the dereverberation of distant speech audios. For this, a solution is assembled using BN derived from an DNN and an automatic coding method for dereverberation. The audio signal representation for network input is done by the 25 dimensional MFCC method with audio frame length of 25ms, frame change of 10ms and sampling frequency of 16kHz. The solution architecture has two inputs for the MFCC audio signal: one has the objective of performing a dereverberation of the audio signal, where it executes an 5 layer DNN entitled as an automatic encoder of cepstral domain for elimination noise; the other has the objective of transforming the original features of the audio signals into discriminative features for the speeches with reverberation, where a 5 layer DNN is performed with BN being extracted from layer 3. The two networks are conceived as DBN but, according to the authors, after configuring DBN using Restricted Boltzmann Machines (RBMs), it is trained discriminatively using the

backpropagation algorithm to maximize the probability of class labels and in general, after this discriminative training type, a DBN is called an DNN. The Japanese Newspaper Article Sentence (JNAS) corpus, which obviously contains articles from Japanese newspapers, was used as dataset of clean speeches. The audio datasets Real World Computing Partnership (RWCP) and CENSREC-4, that contains Japanese sound scene bases, were used to simulate the reverberation over clean audios. Eight types of reverb were used in the tests and training of the network, which normally occur in everyday situations, among them: Japanese style bedroom and bathroom, small and large rooms with tatami flooring, conference room, elevator room, echo room in panel and cylindrical formats. The performance results that the proposed method surpassed some approaches to dereverberation, considered state-of-the-art, such as multichannel least mean squares (MCLMS). Compared with the MCLMS, a reduction in the relative error rates of 21.4% was obtained with the use of DNN for discriminative transformation, which performs the extraction of BF and 47.0% using the DNN that works as an feature automatic encoder, performing the dereverberation. In addition, the use of both DNNs has further improved performance.

SFE – S31: [131] propose a 3 layers RNN-LSTM architecture, each layer having 768 nodes, with a projection of 256 nodes. The solution is based on audio embeddings using d-vector, which according to the authors had already submitted contributions to studies such about SV. The research is directed to the use of d-vectors for SD, working in performance comparisons with the traditional i-vector. The audio signals are represented with log-mel-filterbank energies. Evaluations are carried out on 3 public datasets: CALLHOME American English (LDC97S42 + LDC97T14); 2003 NIST Rich Transcription (LDC2007S10), the English conversational telephone speech (CTS) part; and 2000 NIST Speaker Recognition Evaluation (LDC2001S97), Disk-8, known in the literature as CALLHOME, which contains 500 statements in six languages: Arabic, English, German, Japanese, Mandarin and Spanish. Diarization was performed using 4 clustering algorithms: Naive, Links, K-Means and Spectral. The performance comparison is made between the d-vector and i-vector model using the Diarization Error Rates (DER) metric. The proposed d-vector model shows superiority over i-vector in all evaluations using the 4 clustering algorithms.

2.5 Results Discussion

The analysis of the 34 studies selected by this SLR showed a wide research variety. The results presented by the RQs and the QA were very enriching, helping to learn about the topic addressed. Carrying out a general analysis on the topic, which deals with the use of ANN for SFSP, interesting issues related to this type of research can be highlighted. Beginning on audio processing, the identification of the 21 distinct audio characteristics covered by the research shows the wide variety of situations that were explored by the studies. This result was recorded in Chapter 2.4.5 in response to RQ5, in Tables 2.18 and 2.19. Most of the studies presented 2 types of audios characteristics but there were studies that arrived to present 5 audio characteristics. Particularities existing in each audio characteristic make us reflect on the complexity existing in research of this nature. Although the interest of this SLR was to explore SFSP-related research, while reading the papers it was possible to identify other research focuses that also extract audio features, such as emotion recognition, speech recognition and language identification. This helped to show the variety of possibilities for research on audio. Analyzing speaker-focused studies, the following SFSP areas were identified in the selected studies: SA, SD, SI, SR, SS, SSD and SV. This result responds to RQ1 as recorded in Chapter 2.4.1, listed in Table 2.13. This further capillarizes the possibilities of research on audios and shows that, proportionally, considering the number of 34 studies analyzed, the 7 SFSP areas identified represent a wide variety showing that research of this nature is well diversified.

This SLR has specialized in analyzing ANN solutions to address SFSP. During the analysis, we concluded that most articles have two main phases in the ANN system for SFSP: SFE and SM. SFE is the solution for extraction of audio speaker features and SM is the solution modeling focused on the speaker using as input the data structure created in SFE. ANN solutions were presented for both SFE and SM and therefore we decided to segment the QA results in two lists: studies focused on SM (Table 2.10) and studies focused on SFE (Table 2.11). The ANN architectures presented by the 35 solutions analyzed were obviously completely different. We try to group the solutions in similar types of ANN architectures, creating a typification and quantifying it with the objective of identifying trends. Even so, the number of different types of ANNs architectures was considered large. We found 18 different types of ANN architectures in the 35 solutions. The result of this typification served as a response to RQ2 and was recorded in Chapter 2.4.2, listed in Table 2.14. The 22 SM solutions and the 13 SFE solutions were both related to 11 different types of ANN architectures. Taking into account the proportionality of

these numbers, they show that apparently the SFE researches present a greater variety of solution types than the SM researches. Of the 18 types of architectures identified, 11 used a single type of ANN in their architectures, while the other 7 used two or three types.

Comparing SFE with SM studies, in general, SFE solutions seem to be more complex. This complexity is justified by the wide variety of situations that the audio datasets can represent. Each SFE solution extracts from the audios features that are related to the specific situations treated as objectives by the studies. Therefore, for each situation analyzed by the studies, the ANNs solutions combined with feature extraction methods are quite diverse, appearing to be more complex solutions. Other important issues addressed by the SFE solutions were related to the moment for extracting the audio features and their better vector representations. BF and Embeddings were examples of solutions that addressed these issues. Vector representations with speaker features extracted from the SFE solution are used as SM input solutions for modeling the ANN solution. SLR selected 13 studies that presented SFE solutions. Instead of directly reading audios, these SFE solutions are powered by conventional features extraction methods. But these methods were also used in the studies as a baseline to compare their results with the results obtained by SFE ANNs. In the SM solutions, the conventional features extraction methods used to produce ANN input were also identified. RQ3 asks about the conventional features extraction methods identified in the analyzed studies. Chapter 2.4.3 presents the answers in Tables 2.15 and 2.16 for the SM and SFE studies, respectively. The conventional MFCC method was widely used by SM and SFE studies. In SM solutions, where a total of 12 methods were identified, the MFCC was used as ANN input in 40.74% of cases. In SFE solutions, where a total of 8 methods were identified, it was used as a baseline in 60% of comparisons and as ANN input in 46.67% of cases. Another issue related to the analysis was the types of metrics practiced by SFSP areas. There was curiosity about what types of metrics existed and if there was any trend in their use for each SFSP area. Many metrics were identified, 31 in total, and they were used 70 times in the 34 studies. The most widely used metric was the EER, being used 18 times, corresponding to 25.71% of all uses, and mentioned in 5 of 7 SFSP areas. Secondly, the most used metric was Accuracy with 9 uses (12.86%). The other metrics had a maximum of 5 uses. The main metrics identified in their accounting are shown in Chapter 2.4.4, in Table 2.17, in response to RQ4. It was not possible to identify a metric standard for each SFSP area because studies in the same SFSP area used different metrics.

A QA was performed on the studies selected by the SLR. Ten questions were created, each containing a grade and a weight. The objective was to verify research qualities and

analyze informations related to the theme SFSP for ANN. QA results were organized in two lists for SM and SFE studies. SM and SFE study results were ordered by their final grades and presented in Tables 2.10 and 2.11 respectively. RQ6 asked "Which studies had QA with a score higher than 75% of the maximum score?". In Chapter 2.4.6, in Tables 2.20 and 2.21, 6 SM and 4 SFE studies were presented in response to the RQ6. A brief report on each of these 10 solutions was also requested in RQ6. For each one, it was recorded in Chapter 2.4.6: its architecture, the research objective, a brief summary of the idealized solution and the performance result of the solution in comparison with other solutions. During the analysis of the studies, solutions were identified that made comparisons with the state-of-the-art. Of the 22 SM studies, 3 compared their solutions with the state-of-the-art and managed to overcome them. As for the other studies, 11 made comparisons with other solutions considered baselines and 8 made comparisons with variations of the proposed solutions themselves or with no other solution. The 3 studies that overcame the state-of-the-art were S2, S3 and S5. They were among the 8 studies best evaluated by QA but only S2 and S3 were among the 6 selected by RQ6. In relation to the 13 SFE studies, 3 exceeded the state-of-the-art. The other 10 studies made comparisons with solutions considered baseline. The 3 studies that overcame the state-of-the-art were S27, S28 and S31. They were among the 4 studies best evaluated by QA and all were selected by RQ6.

Briefly mentioning each of these studies that overcame the state-of-the-art, we will emphasize the aspects highlighted by their solutions that made a difference during the comparison of results. In the study S2, [134] propose a task-independent model, called FreqCNN, to automatically extracts distinctive features from each frequency band using convolution kernels from a ANN. The authors structured on the network two types of blocks called "Basic Convolutional Block" and "Attention-Based Block". The study uses the spectrogram as a representation of the audio signals and as ANN input. The spectrogram represents time on the abscissa axis (x) and frequency on the ordinate axis (y). The authors explain that normally the spectrogram is divided into frames within the same time domain that are called a time-distributed spectrogram along the x-axis. But in this study they practice a division of the spectrogram distributed in frequency, by the y-axis. The idea is to pay attention to the energy distribution in different frequency intervals over the entire time window. In the Basic Convolutional Block, with this representation distributed in the frequency, using multiple convolution layers, global and local information of the frequencies are extracted. In the Attention-Based Block, an attention mechanism is used where the model learns to reorganize the global feature representation. Using different local features, the model reorganizes them to form a new global feature

representation. By aggregating all the attention-based global features and the original global representation, the final output of the block is obtained. The proposed CNN has overcome the state-of-the-art. The proposed CNN surpassed the state-of-the-art and its results showed better performance than VGG-11, ResNet-18, ResNet-34, ResNet-50 and i-vector solutions, running on 3 relevant data sets and using Accuracy and Unweighed Average Recall (UAR) metrics.

Bianco et al., in study S3 [14], inspired by the neural network VoxCeleb VGG-M [102], proposed a modification in it and present two CNNs(ResNet): ResNet-18 and ResNet-34. ResNets were used in the SR problem in the wild, where utterances maybe of variable length and also contain irrelevant signals [135]. The problems of SI and SV are tested and compared to other methods considered state-of-the-art. The networks are fed with audio data spectrograms. These ResNets were originally designed for image classification. The proposed architecture is trained using a linear combination of two loss functions: contrasting center loss and crossed entropy softmax. This allowed the construction of a trained network for SI that incorporates discriminative features. In addition, these features can be applied directly to the SV using cosine similarity, without adding complexity to the training process. The experimental results show the effectiveness of the proposed solution in relation to the state-of-the-art. The proposed network shows to be robust in unrestricted conditions and, more important, it shows to be quite robust in a multilingual characteristic. In SI tests the best top-1 accuracy is obtained by the proposed ResNet-34 architecture with contrastive-center loss, with an improvement of 4.7% with respect to the state-of-the-art. In the study S5, [44] presents the QUT SR system as a competitor in the SR challenge Speakers In The Wild (SITW) 2016. The proposed system achieved a ranking of second place, out of all participating teams, in the main core-core condition evaluations. In this condition, a segment of audio with speech from a single speaker (but including potential non-speech and noise portions) is compared to another segment of speech from a claimant (also possibly including non-speech and noise portions). This system uses an i-vector/PLDA approach, with domain adaptation and a DNN trained to provide speaker feature statistics.

In the study S27, from [33], the proposed solution mentions that the research stimulus was generated due to the few studies already carried out with DNNs to recognize speakers who speak at a distance. In this study, BF derived from a DNN and a cepstral domain Denoising AutoEncoder (DAE)-based dereverberation are presented for distant-talking SI, and a combination of these two approaches is proposed. The proposed method shows superior results to the methods MultiChannel Least Mean Squares with Spectral Subtraction

(MCLMS-SS), MultiStep Linear Prediction with Spectral Subtraction (MSLP-SS), BF extracted from MultiLayer Perceptron (BF-MLP) and MFCC with Cepstral Mean Normalization (MFCC-CMN). Compared with the MCLMS, authors obtained a reduction in relative error rates of 21.4% for the BF and 47.0% for the autoencoder feature. Moreover, the combination of likelihoods of the DNN-based BF and DAE-based dereverberation further improved the performance. [61] present in the study S28 a novel solution termed Speaker2Vec to derive a speaker-features manifold learned in an unsupervised manner. The DNN is employed for SS, but the authors mention that it could also be employed for SD and SI. The assumption of short-term active-speaker stationarity is analyzed, that is, audio parts with temporally-near speech segments belong to the same speaker. And for this, embeddings are obtained from the speaker using DNN. During the DNN training, two sequential audio segments are extracted from speakers' speeches. The first segment feeds the DNN and the next segment goes to the DNN's end to obtain a comparison with the first segment processing result. The trained model generates the embeddings for the test audio and applies a simple distance metric to detect speaker-change points. The proposed method outperforms 5 SS algorithms considered state-of-the-art and MFCC based baseline methods on four evaluation datasets. In the study S31 [131] report on the domain of i-vector-based audio embedding techniques in SV and SD applications. They mention about the rise of DL in various domains and that ANN based audio embeddings, also known as d-vectors, have consistently demonstrated SV performance. Based on these observations, the study proposes the development of a new d-vector based RNN(LSTM) solution to SD. The system is evaluated using three public datasets and is carried out in conjunction with 4 clustering methods. RNN(LSTM) results are compared to those of 6 publications taken as a reference, managing to overcome all of them.

Based on the content analyzed in the 34 studies, it was possible to perceive that SFSP research using ANN is in the process of evolution and has overcome methods considered state-of-the-art. Many studies have shown great complexity in their solution architectures, mainly the SFE studies. SFE solutions need to perform the extraction of specific audio features. It is at this moment that the speaker speech features are selected and the other sounds and noises are filtered. To extract the speaker features, several methods were observed in the analyzed solutions, the MFCC being the most used. In an SFE solution, the speaker features are inputed in the ANNs to improve the speakers data and to produce feature vectors, which have higher quality in their representation. These feature vectors are used as input to another speaker classification or clustering solution, which can be another ANN of an SM solution. The SM studies also presented very interesting solutions

using the training and modeling of ANNs for the speakers classification or clustering. During the execution of the SLR, 7 SFSP areas were identified: SA, SD, SI, SR, SS, SSD and SV, a number of segmentation areas considered large for the 35 analyzed solutions. This shows the variety of areas in SPSF research. The largest amount of research was directed to SV, SI and SR, corresponding to 10, 9 and 6 solutions, that is, 71.43% of the total. Another important topic, which caught our attention, was the great diversity of acoustic characteristics identified in the datasets used by the solutions. We identified 21 different acoustic characteristics and presented in tables 2.18 and 2.19 which of these were used by each SM and SFE study. Most datasets used more than one acoustic characteristic, the S3 study being the one that used the most, in total 5. Some of the 21 audio characteristics identified were: degraded audios, speakers from different professions, bilingual or trilingual speakers, short speeches, multi-speakers, real life conditions, audio falsification categories and telephone or microphone conversations. We also adopted a representation for the identified ANN architectures in order to quantify and identify a possible trend of use in this type of research. The variety of architectures was very large because many studies used different types of ANNs in the same solution. We recorded 18 types of ANN architectures found and we could not say that there is a trend in the type of ANN used for SFSP.

Although the focus of this SLR was on the speaker, we identified in these same 34 studies the conduction of research directed to the identification of emotions, speech and language using also the extraction of characteristics from the audios. Taking into account that ANNs are computer systems inspired by the functioning of the animal brain [7], if we associate the solutions analyzed by this SLR with the functioning of the brain and other systems of the human body, we can say that the SFE and SM solutions resemble the role played by the human auditory and brain systems in identifying people speaking. But despite the excellent results presented by the studies analyzed, it is clear that research still needs to go a long way towards improvement. In general, the studies analyzed showed excellent results and a perspective of evolution in future research.

Capítulo 3

Technologies and Theoretical Reference Used for the Experiment

This Chapter presents the technologies used to create the experimental plan, used to evaluate the robustness of the CNN architecture. Section 3.1 presents the CNN architecture adapted and the list of technologies and tools used. The MFCC method used to extract features from audio files is explained in section 3.2. Also explained in the following sections are the data input matrix used to CNN training and the criteria established for CNN performance evaluations.

3.1 CNN Architecture Adapted

CNN architecture adapted in this work was inspired on the CNN proposed by [26]. This CNN features 1-dimensional convolutional filters and was used by [26] as a specialization proposal for the SI task based on degraded audios. The learning focus is directed to the sub-global and supra-global features of each speaker. Such features belong to the human speech production apparatus. In its training, the data input was made with the features extracted from the audio spectrogram images, by the MFCC method.

Article [26] presents an interesting explanation about the use of 1-dimensional CNN (1D-CNN) in Speech Processing (SP) solutions. The authors mention that small square-shaped filters are especially good for learning local patterns in image data, such as edges and corners, due to the high correlation between pixels in a small local neighborhood [26]. But this is not the case when a matrix with MFCC features is inputed in CNN, since, as far as we knows, a local semantic structure cannot be captured by a two-dimensional filter and therefore a 1D filter becomes the best solution for learning speaker-dependent

features stored in the MFCC feature matrix. According to [26] the time variable is not such a relevant characteristic for reading audio data in SR and SI tasks. They says that, in the field of speech recognition, 1D filters across the time variable have shown promising results by effectively learning temporal features in the data. However, in the context of text-independent SR, the temporal relevance of speaker-related features is greatly reduced (but not eliminated), as the content of speech is generally unrelated to the speaker's identity, especially in cases where data is collected for experimental research purposes and not in a natural conversational mode. The use of MFCC method by [26] is due to its ability to capture acoustics features of supra-glottic and sub-glottic vocal tracts, reported by the authors as more beneficial for the SI task. Such features register the acoustics of the trachea-bronchial airways and are known to be robust to noise in SI task. Therefore, such MFCC ability, indicates its potential to contribute to CNN learning. Comparative tests were performed by [26] between the 1D-CNN, UBM-GMM and i-vector/PLDA models and 1D-CNN outperformed the other two solutions. Details of the results presented by [26] can be found in Chapter 2.4.6.

CNN from [26] was analyzed and adapted by us. In our CNN version, we kept the same types and amounts of layers, and the same amount of feature maps produced by the convolution and maximum pooling layers. The following changes were made: maximum pooling strides changed from 2 to 1, in kernel size of the 3 convolution layers and in number of speaker classes. The purpose of our work is also a specialization in SI task but based on multilingual speakers, using the dataset SIWIS proposed by [14]. MFCC method was also used to represent the audio signals that feed the CNN. Through a 26-dimensional MFCC the audio files features are extracted, with the objective of learning by CNN the "voiceprint" of each speaker. Figure 3.1 shows the 1D-CNN architecture adapted and used in the experiments of this work.

3.1.1 Technologies and Tools Used for CNN Adaptation and Training

For the adaptation and training of the CNN architecture, the following technological resources were used:

- Python Programming Language version 3.6.4[1];
- Anaconda version 5.1.0[2], which is a python distribution platforms;

[1] `https://www.python.org/`
[2] `https://www.anaconda.com/`

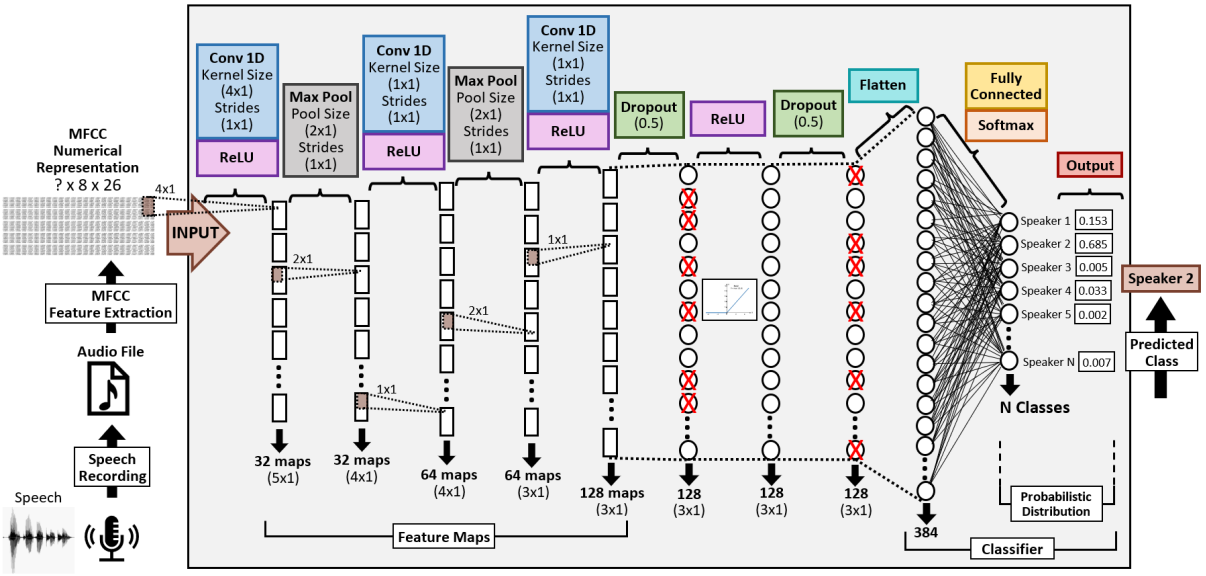


Figura 3.1: CNN architecture adapted from [26], being used by us in a SI task.

- Tensorflow version 2.0.0[3], which is an open source library used for numerical computing and large-scale machine learning;
- Keras version 2.2.4[4], which is an open source software library that provides a Python interface for ANN;
- Spyder[5], which is an Integrated Development Environment (IDE), a free and open source scientific environment for Python;
- Notebook Dell Inspiron 5570[6], 15-inch screen; 24 GB of RAM; Intel® Core™ processor i7-8550U CPU @ 1.80GHz [4 Cores] [8 Logical processors]; Microsoft Windows 10 Home Single Language operating system, 64 bits; AMD Radeon™ 530 graphics card.

CNN training was conducted over 200 epochs. A total of 14,744 audio files from the dataset SIWIS were used. 13,270 files (90% of the total) were used for training and validation, 80% of which were used for training and 20% for validation. 1,474 files (10% of the total) were used for CNN testing. Before starting the training of CNN, the matrix containing the data for training and validation was randomly ordered. Figure 3.2 shows the CNN architecture summary, built for training. A total of 27,652 trainable parameters were recorded. This CNN architecture summary is the result of building the CNN architecture design shown in Figure 3.1.

[3] https://www.tensorflow.org/
[4] https://keras.io/
[5] https://www.spyder-ide.org/
[6] https://www.dell.com/

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv_n1 (Conv1D)             (None, 5, 32)             3360
_________________________________________________________________
max_pooling_n1 (MaxPooling1D (None, 4, 32)             0
_________________________________________________________________
conv_n2 (Conv1D)             (None, 4, 64)             2112
_________________________________________________________________
max_pooling_n2 (MaxPooling1D (None, 3, 64)             0
_________________________________________________________________
conv_n3 (Conv1D)             (None, 3, 128)            8320
_________________________________________________________________
dropout_n1 (Dropout)         (None, 3, 128)            0
_________________________________________________________________
relu_n1 (ReLU)               (None, 3, 128)            0
_________________________________________________________________
dropout_n2 (Dropout)         (None, 3, 128)            0
_________________________________________________________________
flatten_n1 (Flatten)         (None, 384)               0
_________________________________________________________________
dense_n1 (Dense)             (None, 36)                13860
=================================================================
Total params: 27,652
Trainable params: 27,652
Non-trainable params: 0
_________________________________________________________________
```

Figura 3.2: The CNN architecture summary.

3.2 Audio Feature Extraction Method Used

Audio feature extraction is the method used to filter out the features of interest that are embedded in the audio. They are widely used in SFSP solutions, as we could see through SLR, where MFCC was identified as the feature extraction method most used by the analyzed studies, as recorded in Chapter 2.4.3. In this studies, after performing the features extraction, data are used as input for ANN training. MFCC was used by [26] to train the CNN architecture selected for this experiment. MFCC analysis is a audio feature extraction method widely used in nonstationary signals study, including the recognition and speech intelligibility studies [72]. MFCC representation was created by Paul Mermelstein in 1976 [97]. In his article, Mermelstein emphasized that mel-based cepstral parameters have the advantage that generally fewer parameters suffice for an adequate representation of the power spectrum than other coefficients representations [97]. Detailing a little more, MFCC method initially receives the speech as input, converts the voice signal that has its base in time domain to frequency through the Fourier Transform. The signal in frequency is then processed by the Mel Filter Bank and after that Discrete Cosine Transform is done for transforming the mel coefficients back to time domain. The result of this method produces the MFCCs. Figure 3.3 represents the MFCC method steps, according to the content presented by [97] [100] [125].

MFCC is based on the “mel” scale, which is a theory inspired on the characteristics and perceptions of human hearing [71]. Mel scale was created by Stevens, Volkmann and

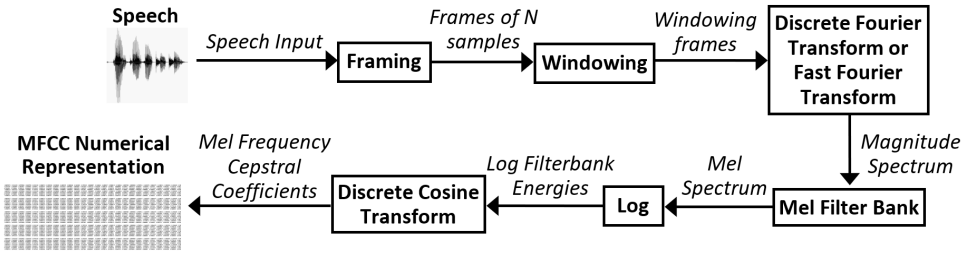


Figura 3.3: MFCC method steps.

Newmann in 1937 [124], which define it as follows: "The mel scale equates the magnitude of perceived differences in pitch at different frequencies". That is, the mel scale reflects how people hear musical tones. Human perception of frequency contents of sounds for speech signal does not follow a linear scale. Thus for each tone with an actual frequency, measured in Hertz (Hz), a subjective pitch is measured on mel scale [125]. The reference relationship between the mel scale and frequency is as follows: the pitch of 1 KHz, tone 40 dB (decibel) above the perceptual hearing threshold, is defined as 1,000 mels [125]. The approximate formula used to calculate the mels for a certain frequency f in Hz is::

$$Mel(f) = 2595 * \log_{10}(1 + \frac{f}{700})$$

3.3 Data Input Matrix Used for CNN Training

To perform CNN training, there is first the need to organize a data structure to be used as input to CNN. This data structure must store specific information extracted from the audios containing the speaker's features to be used in CNN training and is formed by a three-dimensional matrix using the dimensions: Samples X Timestep X Features. The method used to extract features is MFCC, which is traditionally used in Speech Processing research, as presented in Chapter 2. For each audio reading time unit, we established that a 26-dimensional MFCC vector would be extracted, that is, 26 speaker features. There was then the need to establish what would be the lenght of time for reading the audio to perform each MFCC extraction. As the speaker's speech is CNN's learning focus, audio reading for feature extraction needs to be based on a lenght of time that represents a minimum unit of speech. We then resorted to literature in search of this representative period of a minimum unit of speech.

About a minimum unit of speech, [8] say that words can be broken into syllables and phonemes, and the phoneme is the unit in the speech stream represented by the symbols in an alphabetic script. In [129] different units of word representations are mentioned, when

they say that: “In English, Dutch, and other European languages, it is well established that the fundamental phonological unit in word production is the phoneme; in contrast, recent studies have shown that in Chinese it is the atonal syllable and in Japanese the mora”. In article [29] the authors suggest that language rhythm may be the key to predicting the basis of speech segmentation. They explain that each language can have a different segmentation that represents this language rhythm: just as stress is the basis of speech rhythm in English, in French the rhythm is based on syllables and in Japanese the unit of language rhythm is the mora. In Japanese, the phonological unit for speech production is considered to be the mora [78]. Mora is a rhythmical unit typically consisting of consonant e vowel or just vowel, but never consonant alone [129]. For [79], the language rhythm on Japanese, which is sometimes called syllable-timed, is based on the mora which roughly corresponds to a Japanese letter or consonant-vowel syllable.

In [79] an investigation has been made for individual phonemes on Japanese, focusing mainly on their duration in continuous speech, spoken at different speeds: fast, normal and slow. The conclusion of this investigation was the normal speaking rate (n-speech) is, on average, 150 milliseconds/mora (or 400 morae/minute). Although we have identified minimum speech representations for some languages, we could not find any research that proposed the average duration of time of a minimum speech unit for the existing languages in the dataset SIWIS. Only in [79] we find a time duration proposal, but for the Japanese language, using mora as the rhythmic unit concept of the language. As we needed to determine a length of time for reading the audio features through a sliding window (Timestep) we decided to take as a reference what was proposed by [79] and established as a representation of the minimum unit of speech the period of 150 milliseconds (ms). We also established that, during the audio reading process, there is an overlap of the sliding windows of 20%, that is, the last 20% of the audio of each timestep read is repeated at the beginning of the next timestep. This window overlap helps CNN learn during its training as it establishes a link between the last sample read and the next sample to be read, which are labeled with the same speaker class. A timestep, with its 150 ms, then considers for each reading window: 80% new audio data and 20% repeat data from previous window. Therefore, the sliding window stride for reading new audio data is 120 ms.

To complete the structure of the data input matrix, it remained to establish the number of timesteps that represent an audio sample. We identified that the length of time of the smallest audio file in the dataset SIWIS is 0.7 seconds, evaluating the possibility of using it as a smaller size reference for an audio sample. We also verified that in the entire dataset SIWIS only 3 files were less than 1 second. When running these 3 files, we could

not identify any speech sounds. Unlike other audios, which we selected by sampling, and we clearly hear the speeches of the speakers. These 3 smaller files were then excluded from the dataset SIWIS for CNN training. We then observed that the minimum audio sample size could not be less than 1 second. Considering that 1 timestep is 150 ms, the number of timesteps closest to 1 second is 7, totaling 1.05 seconds. As there is no audio with a length of 1.05 seconds in the dataset SIWIS, we have established that the number of timesteps representing an audio sample is 8, totaling 1.2 seconds.

During the reading of an audio, if the last part does not complete the exact size of a sample, being therefore less than 8 timesteps, the initial timesteps of this audio are repeated to complement this last sample. Each sample is labeled with the audio speaker's name. The amount of audio samples depends on the size of the audio. A group of samples is stored in a Batch for each CNN data input. We set the quantity of 50 samples for each lot. Therefore, the total amount of batches in CNN training depends on the amount and sizes of the audio files. In summary, the data structure organized for input the neural network presents a matrix, formed by Batch x Samples x Timesteps x Features, having the following definitions:

- Feature - Minimum unit of speaker features representation. Each feature unit is the representation of one MFCC;
- Timestep - Phonetic representation unit of speech defined with a time of 150 ms. Contains 26 MFCC Features representing 26 speaker features, read in each 150 ms section of the audio;
- Sample - Unit of sample of the speaker's speech. It is the smallest representative unit of the speaker's speech extracted from the audio file. Contains the size of 8 Timesteps;
- Batch - Sample pooling unit. Divides the read samples from the audios into predefined batch sizes. It stores all data extracted from the audio files of the dataset SIWIS to use them as input to CNN in each training epoch. The defined size for each batch is 50 Samples.

Figure 3.4 shows the numerical representation of MFCC features extracted from an audio file of the dataset SIWIS. It is also shown the representation of the configuration established for the data input matrix that feeds the CNN. The representation of the data input matrix for CNN is also shown. The matrix contains 3 dimensions: Samples X Timesteps X Features. Timesteps and Features dimensions were predefined with sizes 8 and 26 respectively. Samples dimension varies according to the audio file size. The

data in Figure 3.4 refer to the audio file EN_A3_LEO_008.wav, 4.28 seconds long and represented in the matrix by 5 samples.

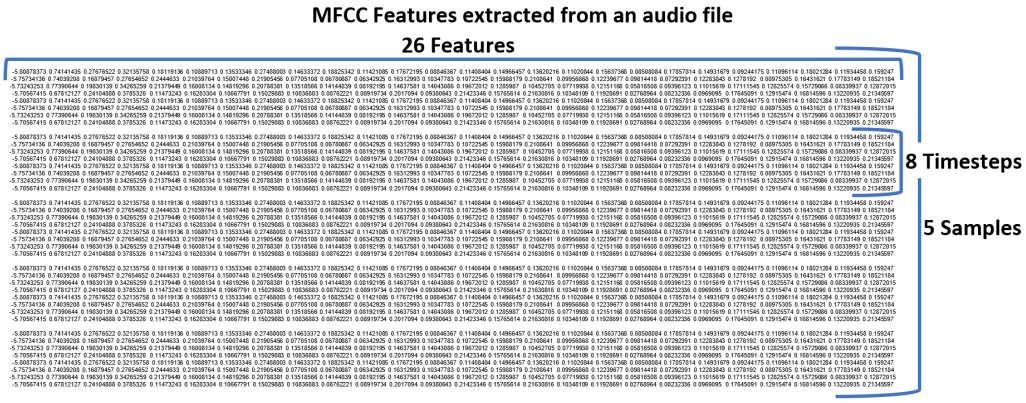


Figura 3.4: MFCC features extracted from an audio file of the dataset SIWIS and representation of the dimensions that constitute the data input matrix: Samples X Timesteps X Features.

3.4 Criteria used for Evaluating CNN Performance

Two types of evaluations were used to analyze CNN performance: Accuracy and F1 metrics, to evaluate the CNN performance in its training; and two prediction calculations, to evaluate the CNN performance executing the SI task.

3.4.1 Accuracy and F1 Metrics

Accuracy and F1 metrics were used by the experimental plan, detailed in Chapter 4, to evaluate CNN performance in its training. The two metrics are widely used in the scientific literature and use in their formulas the 4 values existing in the Confusion Matrix: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). These values are counted during CNN training by comparing the predicted class for a problem with its true class. Figure 3.5 presents a confusion matrix model. In this Chapter the Accuracy and F1 metrics are explained.

Accuracy: According to [96] "Accuracy is a qualitative performance characteristics, expressing the closeness of agreement between a measurement result and the value of the measurand". Accuracy represents the number of correct predictions divided by the total number of predictions. It is a widely used metric that assesses how well a binary classification test correctly identifies or excludes a condition. In this case,

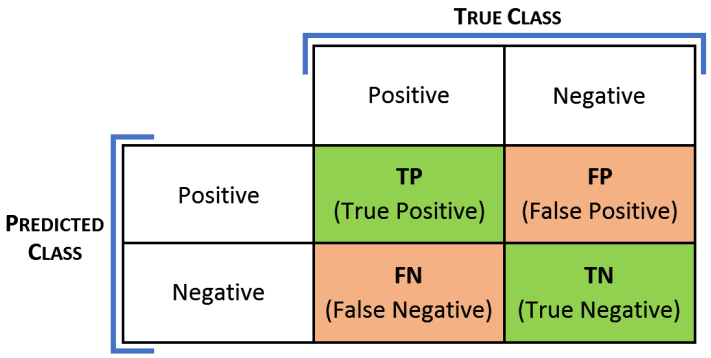


Figura 3.5: Confusing matrix of predictive results.

the answer to be checked is whether the audio speaker predicted by the network is correct. A maximum accuracy score is 1.0 and means that all elements correctly predicted correspond to the total amount of predictions. The Accuracy formula is:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} = \frac{TP + TN}{TP + TN + FP + FN}$$

F1 Score: It is defined as the harmonic mean of precision and recall [140]. This score takes both FP and FN into account. A good F1 Score means that there are low FP and low FN. Accuracy is a better metric to use when the distribution of the class is similar, while the F1 score is a better metric when there are unbalanced classes. In many real-life classification problems there is an unbalanced distribution of classes and therefore the F1 score is the most suitable metric. An F1 Score is considered perfect when it's 1, while the model is a total failure when it's 0. The F1 Score formula is:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN}$$

Precision: It tries to answer the following question: "Which portion of the positive elements was really correct?". Another way to explain its meaning would be to consider relevant and selected elements to then answer the following question: "How many selected elements are relevant?" [59]. Figure 3.6 highlights who are the relevant elements and the selected elements. Precision is also referred to as Positive Predictive Value (PPV). A perfect precision score has a value of 1.0 and means that all selected (predicted) elements are relevant but does not inform whether all relevant elements have been selected. The Precision formula is:

$$Precision = \frac{TP}{TP + FP}$$

Recall: It is a synonym for True Positive Rate (TPR) and more commonly called Sensi-

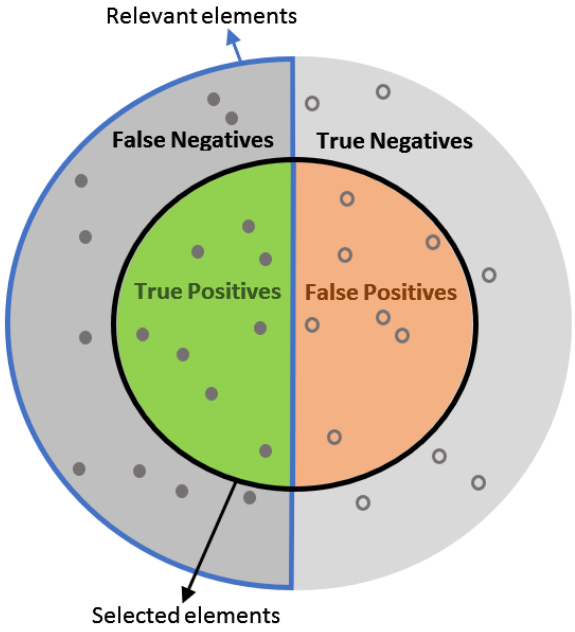


Figura 3.6: Relevant and selected elements.

tivity. Like Precision the Recall calculation is also based on the relevant elements. Recall tries to answer the question: "Which portion of the truly positive elements was identified correctly?". Another way to explain its meaning would be to consider relevant and selected elements to then answer the following question: "How many relevant elements are selected?" [59]. A perfect Recall score has a value of 1.0 and means that all relevant elements have been selected (predicted) but it does not inform how many irrelevant elements have been also selected. The Recall formula is:

$$Recall = \frac{TP}{TP + FN}$$

3.4.2 Class and Probabilistic Prediction Calculations

These two prediction calculations were also used by the experimental plan, detailed in Chapter 4, but to evaluate CNN performance when performing a SI task. When the CNN model processes an audio, the prediction results show the percentage of chances that all classes of speakers known by CNN have to be the speaker of the analyzed audio. The operation of the two prediction calculations is different and both are explained in detail in this Chapter. The prediction is performed for each minimal audio partition, called audio sample. To consolidate the calculation, we average the sample predictions and present the final prediction result for the analyzed audio.

Class Prediction by Audio Sample: This prediction points to only one speaker, among all known speakers, as the supposed owner of speech in each of the analyzed audio samples. After the python function "model.predict_classes[7]" predicted all audio samples, we calculated the sample percentage for each predicted speaker. The speaker with the highest percentage of samples is selected as the supposed audio speaker and if his percentage is greater than the limit value for acceptance (65%) the result is considered valid. If this percentage is less than or equal to 65%, the result is considered inconclusive. In order to monitor the result, the supposed audio speaker is also compared with the real audio speaker and if it is the same, the result is considered correct, otherwise it is considered incorrect. This prediction calculation is named by us as "Class Prediction".

Probabilistic Prediction of each Class by Audio Sample: This prediction calculates, for each audio sample, a prediction percentage for each of the known speakers, showing the probability that each one has to be the speaker of the analyzed sample. After the python function "model.predict_proba[8]" predicts all audio samples, we present the final prediction percentage for each speaker by averaging the predictions made for them in each audio sample. The speaker with the highest average prediction percentage is selected as the supposed audio speaker and if his percentage is greater than the threshold value for acceptance (65%) the result is considered valid. If this percentage is less than or equal to 65%, the result is considered inconclusive. In order to monitor the result, the supposed audio speaker is also compared with the real audio speaker and if it is the same, the result is considered correct, otherwise it is considered incorrect. This prediction calculation is named by us as "Probabilistic Prediction".

As a hypothetical example of using the two prediction calculations, suppose that there is an audio, belonging to speaker "A" and that it has 10 sample units. We adopts 65% as the minimum prediction threshold for identifying the speaker as the owner of the audio. Table 3.1 and 3.2 show the hypothetical results of the Prediction Calculation 1 and 2, respectively. Table 3.1 shows that the 10 samples of the supposed audio had a prediction distributed among 4 speakers. CNN predicted that the audio speaker would be speaker "B" in 7 samples of this audio, which means 70% prediction for speaker "B". As 70%

[7]Python function that generate class predictions for the input samples, detailed in `https://www.kite.com/python/docs/tensorflow.keras.Sequential.predict_classes`.

[8]Python function that return estimates for all classes ordered by class label, detailed in `https://www.kite.com/python/docs/sklearn.linear_model.LogisticRegression.predict_proba`.

is greater than the 65% threshold, the speaker "B" is appointed as the supposed audio speaker, obtaining an incorrect prediction result. Table 3.2 shows that the 10 samples of the supposed audio had a prediction distributed among the 5 speakers. The system calculated for each speaker the average percentage of prediction over the 10 samples. Speaker "A" obtained an average percentage of 89.86% in the audio prediction. As this value is greater than the 65% threshold, the speaker "A" is appointed as the supposed audio speaker, obtaining a correct prediction result.

Tabela 3.1: Hypothetical example of Prediction Calculation 1 (Class Prediction) with an incorrect prediction result

Speaker	**Amount of Samples Predicted per Speaker**	**Prediction Percentage per Speaker**	**Predicted Speaker**	**Correct Speaker**
A	1	10%		X
B	7	70%	X	
C	1	10%		
D	1	10%		
E	0	0%		
Result	**10**	**100%**	**Incorrect Prediction**	

Tabela 3.2: Hypothetical example of Prediction Calculation 2 (Probabilistic Prediction) with a correct prediction result

Speaker	**Average Percentage of Sample Prediction per Speaker**	**Predicted Speaker**	**Correct Speaker**
A	89.86%	X	X
B	6.16%		
C	2.51%		
D	1.45%		
E	0.02%		
Result	**100%**	**Correct Prediction**	

In this research, the value of 65% was established as Threshold for the audio speaker prediction classification. This value was empirically established for this investigation of the CNN robustness, being created in a parameterizable way and can be changed in future investigations. That is, if the prediction result is greater than 65% the predicted speaker is appointed as the supposed audio speaker. If the prediction result is less than or equal to 65%, the result is considered inconclusive. The prediction result can be correct or incorrect. Five classifications were used for the prediction results. These classifications have the following meaning:

- Top 5 - percentual when the correct speaker was among the 5 speakers with the highest prediction percentage;
- Top 3 - percentual when the correct speaker was among the 3 speakers with the highest percentage of prediction;
- Top 1 (Speaker Correct) - percentual when the correct speaker was correctly predicted and got the highest prediction percentage among all 36 speakers;
- Top 1 $>$ 65% (Threshold) - percentual when the correct speaker was correctly predicted and reached the prediction percentage above 65% (threshold to point the speaker), being established by this research as the speaker who owns the audio;
- Top 1 $>$ 99% - percentual when the correct speaker was correctly predicted and got the highest possible percentage of prediction.

Capítulo 4

The Experimental Plan

Sections of this Chapter explain the Materials and Methods used by the Experimental Plan created for this CNN robustness analysis. First section shows the material used by the Experimental Plan, which is the audio dataset SIWIS. In the next section, the Experimental Plan is presented, showing an overview of all experimental scenarios performed. Then, the general steps for executing the Experimental Plan are presented, in an organized way, through the representation of business processes. Finally, each scenario of the experimental plan is explained.

4.1 Material Used by the Experimental Plan

For the execution of the Experimental Plan, an audio dataset was used as a strategic item for the elaboration of the experimental scenarios.

4.1.1 Audio Dataset Used

The audio dataset used in these experiments was presented by [14] and is called SIWIS (Spoken Interaction With Interpretation in Switzerland). It is a well-structured multilingual speaker base, with a considerable amount of audios by speaker and which presented good possibilities for experiments to explore SR, SI and SV scenarios. SIWIS research project is a Swiss-NSF (National Science Foundation) funded project gathering several research teams in Switzerland and the CSTR (Centre for Speech Technology Research) in University of Edinburgh [48]. Part of the SIWIS[1] project is this audio dataset that

[1]The SIWIS project website is `https://www.idiap.ch/project/siwis/`, and by redirecting to the University of Geneva website, through the link `https://www.unige.ch/lettres/linguistique/research/latl/siwis/database/`, the audio dataset can be requested.

contains speakers who speak 2 and 3 languages in a universe of 4 languages: the 3 main official languages of Switzerland (French, German and Italian) and English. The dataset has a total of 36 speakers, of which 22 speak 2 languages and 14 speak 3 languages. Table 4.1 shows the number of speakers who speak 2 and 3 languages by gender. Figure 4.1 shows a Venn diagram with the distribution of the 36 speakers, who speak 2 or 3 languages, among the 4 languages of the dataset SIWIS.

Tabela 4.1: Number of speakers in the dataset SIWIS that speak two or three languages by gender.

Gender	**Speaks 2 Languages**	**Speaks 3 Languages**	**Total**
Female	10	10	20
Male	12	4	16
Total	**22**	**14**	**36**

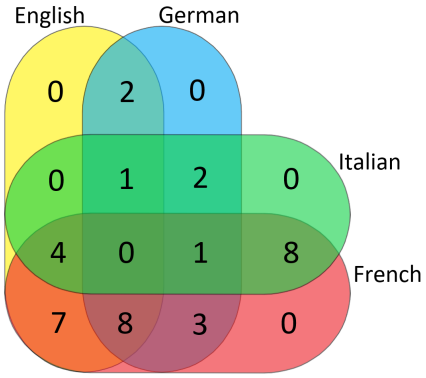


Figura 4.1: Venn diagram showing the distribution of the 36 speakers among the 4 languages, in the dataset SIWIS.

There are approximately 170 audio files for each language speakers speak, corresponding to a total recording of approximately 20 minutes. Dataset SIWIS has a size of 7.0 GBytes, contains 14,744 files in wav format and a total audio time over 23 hours and 30 minutes. Table 4.2 shows the number of speakers and audio files by language in the dataset SIWIS. Table 4.3 shows an accounting of audio files, from the dataset SIWIS, by time ranges.

4.2 Experimental Plan Steps for CNN Analysis

We present experimental plan steps to analyze the CNN robustness. CNN architecture used is shown in Figure 3.1. Experimental scenarios were designed to explore the CNN

Tabela 4.2: Number of speakers and audio files by language, in dataset SIWIS.

Language	**Nº of speakers**	**Nº of audio files**
English	22	3,771
French	31	5,332
German	17	2,903
Italian	16	2,738
Total	**86**	**14,744**

Tabela 4.3: Accounting of audio files by time ranges (in minutes : seconds), from SIWIS.

Time Ranges (min : sec)	**Number of Audio Files**
$<$ 00:01	3
00:01 to 00:02	1,120
00:03 to 00:05	8,727
00:06 to 00:10	4,621
00:11 to 00:15	177
00:16 to 00:20	32
00:21 to 00:35	19
01:14 to 02:00	39
02:05 to 02:27	6
Total	**14,744**

performance when used in SI tasks. Each scenario exposes different situations and variations of the SIWIS audio dataset. Table 4.4 presents the steps for the execution of this experimental plan. For the preparation of each experimental scenario, a different version of the Original SIWIS audio dataset was created and used in CNN training. For the execution of the experimental scenarios, their CNN models performed the SI task using 10% of the audio files from the SIWIS dataset, separated for testing. As a reference model for the experimental scenarios, we used the SI results presented by CNN trained with the Original SIWIS audio dataset. Accuracy and F1 metrics, explained in Chapter 3.4.1, were used to train the CNN to select the model for each experimental scenario. Class and Probabilistic prediction calculations, explained in Chapter 3.4.2, were used to obtain the SI results during the execution of each experimental scenario. The percentages of samples used in scenarios 1, 2 and 3 to reduce the number of speakers, the number of samples or the size of the samples were made empirically, based on common sense to reduce these parameters.

Tabela 4.4: Experimental plan steps for CNN analysis.

Scenario Nº	Experimental Scenario		Criteria for Evaluation
1	Speaker Reductions	80% of speakers	Class and Probabilistic Prediction Calculations
		60% of speakers	
		40% of speakers	
		20% of speakers	
2	Audio File Reductions per Speaker	80% of audio files	
		60% of audio files	
		40% of audio files	
		20% of audio files	
3	Variations in Audio File Size	50% of audio size	
		75% of audio size	
		125% of audio size	
		150% of audio size	
4	SI task using an Unknown Language (German)	A language totally unknown by CNN	
		A language partially unknown by CNN	
5	Adding a new speaker class		
6	SI task using another Unknown Language (Portuguese) for the new speaker class		

4.3 Research Procedure for Executing the Experimental Plan

For the execution of the Experimental Plan, a roadmap was elaborated considering its general steps in a standardized and organized way. For this, business processes were created[2] [3] for the fulfillment of each step. Figure 4.2 shows the business process used to execute each experimental scenario. The business process "Execution of Experimental Scenario for CNN Analisys" contains two sub-processes that group specific activities for CNN training and for using CNN performing SI tasks. Sub-processes are visually identified by Business Process Model and Notation (BPMN) with a "plus" sign inside its blue box. These two sub-processes are explained in sections of this Chapter.

[2]The business processes were mapped using the standard Business Process Model and Notation (BPMN™), `https://www.bpmn.org/`.

[3]The Bizagi Modeler® tool, available at `https://www.bizagi.com/en/platform/modeler`, was used for mapping business processes.

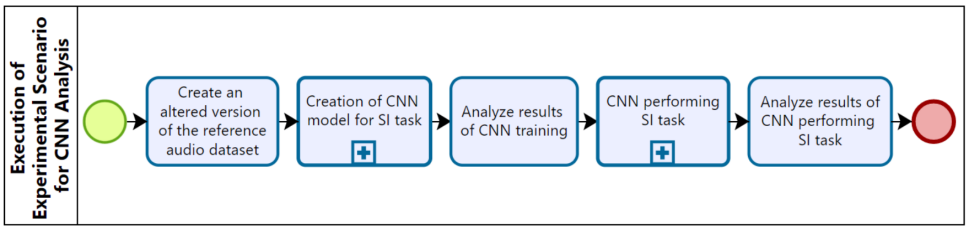


Figura 4.2: Research Procedure using BPMN to Experimental Scenario for CNN Analisys.

4.3.1 Creation of CNN Model for SI task

This business process performs the creation of a CNN model trained specifically for SI task, using SIWIS as the audio dataset and MFCC as the feature extraction method. In this process, specific parameters for CNN training are configured, the analysis of the training results is performed and the CNN model is saved. Figure 4.3 presents this process. The parameters used by each activity during the execution of this business process are detailed in Appendix A.0.1.

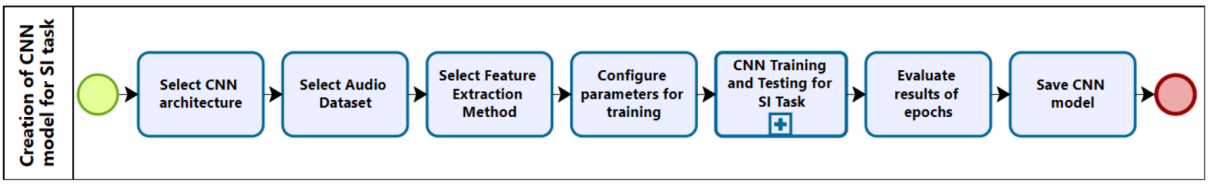


Figura 4.3: Research Procedure using BPMN to Creation of CNN Model for SI task.

4.3.1.1 CNN Training and Testing for SI Task

Sub-process of the Creation of CNN model for SI task process, its purpose is the training, validation and testing of the CNN architecture for SI task. Figure 4.4 presents this sub-process. The parameters used by each activity during the execution of this business process are detailed in Appendix A.0.2.

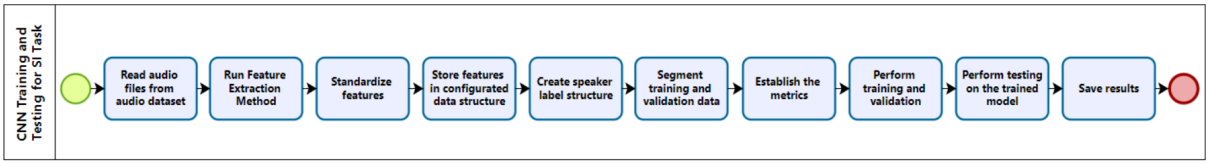


Figura 4.4: Research Procedure using BPMN to CNN Training and Testing for SI Task.

4.3.2 CNN Performing SI Task

This process represents the use of the CNN model, already trained, to perform the SI task proposed by each scenario of the experimental plan. Figure 4.5 shows the mapping of this business process. To automate the execution of the scenarios, a python tool was developed that executes all its activities. The figure 4.6 shows the tool in operation. The parameters used by each activity during the execution of this business process are detailed in Appendix A.0.3.

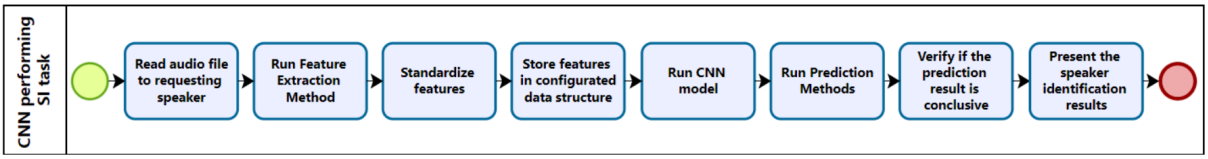


Figura 4.5: Research Procedure using BPMN to CNN performing SI Task.

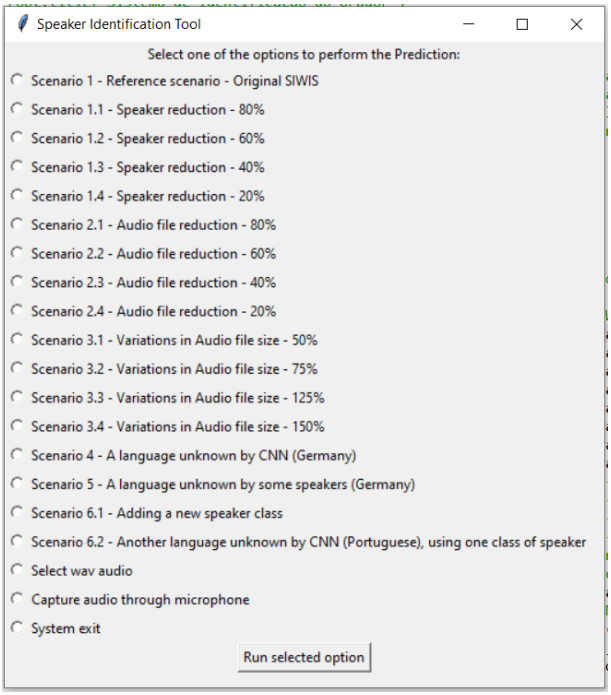


Figura 4.6: Tool constructed in Python for automation of the business process "CNN performing SI task".

4.4 Experimental Scenarios

This section details the experimental plan' scenarios.

4.4.1 Speaker Reductions

This scenario evaluates the CNN performance when being trained with SIWIS containing different amounts of classes (speakers). The purpose of this scenario is to analyze how the number of speaker classes influences the CNN performance when used in SI task. Four new SIWIS versions with speaker reductions were created, using approximately 80%, 60%, 40% and 20% of the total 36 speakers. After training, CNN performs the SI task with the test audio files from SIWIS to evaluate its prediction performance. These new SIWIS versions are identified as versions 2, 3, 4 and 5 in Table 4.5.

Tabela 4.5: Number of speakers used in scenarios for speaker reductions.

SIWIS Scenario	**SIWIS Version**	**Number of Speakers**	**Percentage of Speakers**
Original SIWIS	1	36	100%
Speaker Reductions	2	29	80.55%
	3	22	61.11%
	4	14	38.88%
	5	7	19.44%

4.4.2 Audio File Reductions per Speaker

This scenario evaluates the CNN performance when being trained with SIWIS containing different amounts of audio files to represent the speaker classes. The audio files represent the samples used in CNN training to learn speaker features. The purpose of this scenario is to analyze how the number of audio files influences the CNN performance when used in SI task. Four new SIWIS versions with audio file reductions per speaker were created, using approximately 80%, 60%, 40% and 20% of each speaker's audio files. The file reduction percentage was applied equally to the speakers' languages. After training, CNN performs the SI task with the test audio files from SIWIS to evaluate its prediction performance. These new SIWIS versions are identified as versions 6, 7, 8 and 9 in Table 4.6.

4.4.3 Variations in Audio File Size

This scenario evaluates the CNN performance when trained with SIWIS containing variations in the sizes of the audio files that represent the speaker classes. The purpose of this scenario is to analyze whether increasing or decreasing sample sizes influence CNN performance when used in SI task. Four new SIWIS versions were created with variations

Tabela 4.6: Number of audio files used in CNN training and validation, corresponding to scenarios for audio file reductions per speaker.

SIWIS Scenario	SIWIS Version	Number of Audio Files	Percentage of Audio Files	Discarded Files (less than 1s)
Original SIWIS	1	13,268	100%	3
Audio File Reductions per Speaker	6	10,609	79.96%	2
	7	7,943	59.87%	1
	8	5,326	40.14%	1
	9	2,660	20.05%	0

in audio file sizes, using approximately 50%, 75%, 125% and 150% of its original sizes. After training, CNN performs the SI task with the test audio files from SIWIS to evaluate its prediction performance. These new SIWIS versions are identified as versions 10, 11, 12 and 13 in Table 4.7. Variations in audio files sizes was made by removing 50% and 25% of their endings, for SIWIS versions 10 and 11, and with a doubling of 25% and 50% of their endings, for SIWIS versions 12 and 13.

Tabela 4.7: Number of samples, from audio files, used in scenarios for variations in audio file size.

SIWIS Scenario	SIWIS Version	Total Size of Audio Files (in number of samples)	Percentage of Size
Original SIWIS	1	85,764	100%
Variations in Audio File Size	10	50,149	58.47%
	11	68,537	79.91%
	12	104,737	122.12%
	13	122,293	142.59%

4.4.4 SI task using an Unknown Language (German)

This analysis evaluate the CNN performance when executing the SI task, processing audios of speakers known by CNN, but speaking a language unknown by CNN. The purpose is to evaluate the impact that an unknown languages can reflect on CNN performance. Two situations were planned in this scenario: when the language is totally and partially unknown by CNN.

I - Language totally unknown by CNN: To create this scenario with the same speakers existing in SIWIS, it was necessary to simulate this situation. Therefore, one of the four SIWIS languages was excluded. The excluded language was German, represented

by 17 speakers in SIWIS. Figure 4.7 shows a new version of the Venn diagram, previously shown by Figure 4.1, but now without German audio for 17 speakers, as identified by the white numbers. All speakers were kept in this new SIWIS version despite the exclusion of the German language. Old German-speaking speakers are now represented by only 1 or 2 other languages in SIWIS. This new SIWIS version was then used for CNN training and the CNN model performed the SI task for the test audios of all speakers, including German language audios.

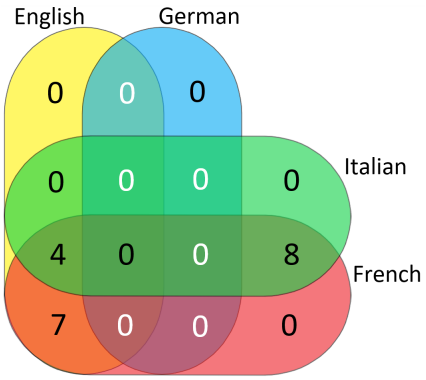


Figura 4.7: Venn diagram showing the distribution of speakers, after removing the German language from SIWIS. White numbers identify reductions, in relation to Figure 4.1.

II - Language partially unknown by CNN: In this scenario CNN performs the SI task, for known classes of speakers, speaking a language known only by some of these speakers. In this other simulation German language is represented in SIWIS by only half of the German-speaking speakers. This trained CNN model processes test audios from all speakers, including German audios. Figure 4.8 shows a new version of the Venn diagram after the removal of audios in German language for some speakers, as identified by the white numbers. All speakers were kept in this new SIWIS version, but of the 17 speakers who speak German, 8 speakers had their German audios excluded, continuing to be represented in SIWIS by the other languages they speak.

4.4.5 Adding a new speaker class

In this scenario CNN performs the SI task for a new speaker class. The new speaker, labeled "LEO", was added in SIWIS increasing the number of speakers to 37. The purpose of this scenario is to analyze CNN's performance when receiving the inclusion of one more speaker class speaking 3 languages. In addition, the new speaker is also used in a next scenario, for tests with another language. To include this new speaker 514 new audio files were recorded by him in three languages already existing in SIWIS: English (167 files),

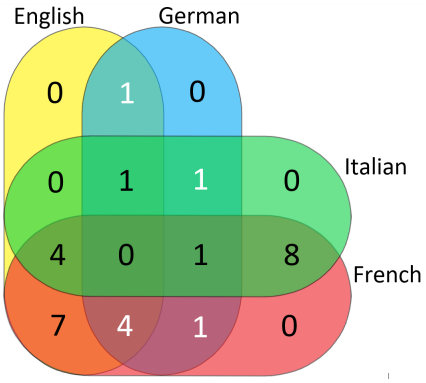


Figura 4.8: Venn diagram showing the distribution of speakers after a reduction of German language for some speakers, in SIWIS. White numbers identify reductions, in relation to Figure 4.1.

Italian (173 files) and French (174 files). Table 4.8 shows the number of audio files for each language, after adding the new speaker LEO. Figure 4.9 shows the updated Venn diagram adding the new speaker LEO with audios in three languages.

Tabela 4.8: Increase in number of speakers and audio files per language, in SIWIS, after the addition of the new speaker LEO.

Language	**Speakers by language**	**Nº of files**
English	23	3,938
French	32	5,506
German	17	2,903
Italian	17	2,911
Total	**89**	**15,258**

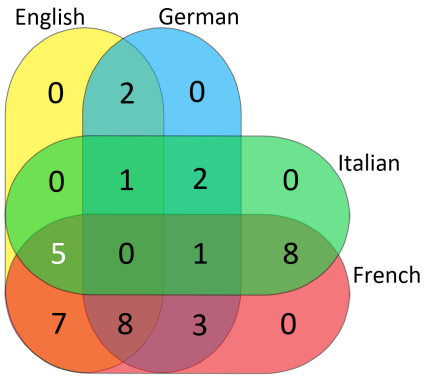


Figura 4.9: Venn diagram showing the new distribution of speakers after inclusion of the new speaker LEO, in SIWIS. The white number identify the increase, in relation to Figure 4.1.

4.4.6 SI task using another Unknown Language (Portuguese) for the New Speaker Class

In this scenario, Portuguese was chosen to represent the fifth language in SI tests with CNN. Portuguese is a non-existent language in SIWIS and therefore the scenario is about another language unknown by CNN. The purpose is to observe if the CNN performance, in the SI tests, suffers any variation when being tested with a fifth language unknown by CNN. To construct this scenario only the new LEO speaker was used. Ten new audio files were recorded in Portuguese by speaker LEO. The sentences of these ten audio files are shown in Table 4.9. SI tests were performed with the test files of the speaker LEO plus the 10 new audio files in Portuguese.

Tabela 4.9: Speaker LEO's 10 new sentences recorded in Portuguese.

Nº	**Audio Size (in seconds)**	**Portuguese Sentences**
1	5.95	O Império Romano foi o período pós-republicano da Roma Antiga.
2	20.68	A Lagarta foi a primeira a falar. "Qual é o tamanho que você quer ter?" perguntou. "Oh, eu não sou exigente com relação à altura", Alice respondeu apressadamente; "Só não gosto de mudar com tanta frequência, sabe." "Não sei", disse a Lagarta [22].
3	12.41	Dorothy morava no meio das grandes pradarias do Kansas, com o tio Henry, que era fazendeiro, e a tia Em, que era a esposa do fazendeiro [86].
4	11.77	Há três coisas na vida que nunca voltam atrás: a flecha lançada, a palavra pronunciada e a oportunidade perdida.
5	3.12	Quem ri por último ri melhor.
6	1.42	Bom dia!
7	12.89	Os Jogos Olímpicos, no formato que conhecemos, foram disputados pela primeira vez em 1896 na cidade de Atenas, na Grécia [42].
8	8.42	As frutas são alimentos ricos em nutrientes e substâncias que contribuem com a saúde.
9	3.79	Viajar faz bem para a vida e para a alma!
10	3.0	Um dois três quatro cinco.

Capítulo 5

Presentation of Experimental Results

In this Chapter, initially, the results of CNN training are shown. Then the results of each experimental scenario are analyzed. These analysis allowed the evaluation of CNN's robustness. According to the robustness definitions shown in Chapter 1.1 we can highlight what was recorded by [108] and [143], who quite objectively said that robustness is the state where the technology or process performance is minimally sensitive to factors causing variability. Therefore, robustness is a very important property for a system, technology or process as it means their ability to guarantee their desired performance or behavior in the face of deviant environmental behaviors or external and internal disturbances.

The prediction results of the experimental scenarios had as reference the results of the CNN trained with the Original SIWIS dataset. All scenario results are presented in Tables and Graphs that show the values of the Class and Probabilistic Prediction calculations arranged in the 5 rankings, presented in Chapter 3.4.2. That prediction values represent the average prediction percentages of CNN performing the SI task for the speakers involved in each scenario.

5.1 CNN Training Results for the Experimental Scenarios

CNN training results are shown in Table 5.1. Accuracy and F1 results, collected from the validation and test steps, are shown for each experimental scenario. CNN training results for the reference model, which used the Original SIWIS dataset, are also shown. We evaluated the metrics results as satisfactory but some CNN training results for the

scenarios were a little surprising. Reference scenario showed a result within our expectations. Speaker reduction scenarios showed an increase in metrics, in relation to the reference scenario, as the number of speakers decreased, as expected. Scenarios of audio file reductions per speaker showed a reduction in metrics only with the amount of 20% of the total. The expectation was this performance would start to drop in first scenario of audio file reduction. In scenarios of variations in audio file size, metric results were also a little surprising because the reduction in audio file size reduces the amount of samples, but it presented a better result than the reference scenario. And when file sizes increased, metrics were expected to increase much more. For scenario of SI using an Unknown Language (German), the complete removal of a language did not present an improvement in metrics in relation to its partial removal. The last two scenarios, whose CNN model was trained to contemplate one more speaker, showed increases in the Accuracy and F1 results in relation to the Original SIWIS, contrary to what was expected, because theoretically adding one more speaker class makes CNN training more complex.

Tabela 5.1: CNN training results for the experimental scenarios.

Scenario Nº	**Experimental Scenario**		**Epoch**	**Validation**		**Test**	
				Accuracy	**F1**	**Accuracy**	**F1**
Reference	Original SIWIS		151	0.8300	0.8344	0.8348	0.8382
1	Speaker Reductions	80% of speakers	122	0.8579	0.8615	0.8551	0.8585
		60% of speakers	179	0.9174	0.9192	0.9139	0.9179
		40% of speakers	91	0.9646	0.9645	0.9607	0.9608
		20% of speakers	33	0.9814	0.9811	0.9803	0.9800
2	Audio File Reductions per Speaker	80% of audio files	94	0.8436	0.8490	0.8468	0.8504
		60% of audio files	100	0.8489	0.8541	0.8467	0.8506
		40% of audio files	92	0.8300	0.8347	0.8399	0.8449
		20% of audio files	155	0.7975	0.8078	0.7839	0.7956
3	Variations in Audio File Size	50% of audio size	89	0.8494	0.8501	0.8442	0.8489
		75% of audio size	101	0.8681	0.8728	0.8618	0.8662
		125% of audio size	131	0.8384	0.8405	0.8286	0.8372
		150% of audio size	51	0.8607	0.8656	0.8604	0.8655
4	SI using an Unknown Language (German)	Language totally unknown by CNN	187	0.8350	0.8383	0.8370	0.8405
		Language partially unknown by CNN	133	0.8430	0.8435	0.8385	0.8413
5	Adding a new speaker class		134	0.8411	0.8427	0.8375	0.8438
6	SI using another Unknown Language (Portuguese) for the new speaker class						

5.2 Results of Experimental Scenarios

This Chapter presents the results of the experimental scenarios, planned in Table 4.4 of Chapter 4.3.

5.2.1 Scenarios of Speaker Reduction

In this scenario, Table 5.2 and Figure 5.1 show prediction results for the Original SIWIS scenario and the scenarios with speaker reductions. Differences in relation to the reference scenario begin to be perceived more clearly in scenario with 60% of speakers, but the biggest differences are noticed in scenarios with large speaker reductions, when the number of speakers is 40% and 20% of total. In scenario with 80% of speakers, the results do not show improvement in CNN performance. Comparing the results in rankings view, we see that Top 1, Top 1 $>$ 65% and Top 1 $>$ 99% show a gradual improvement in CNN performance, as the number of speakers decreases. Top 3 ranking presents high numbers only in scenarios with 40% and 20% of speakers. The scenario results showed that the number of speaker classes influenced CNN performance. This confirms the assumption we had before running the scenario, as theoretically adding one more speaker makes learning more complex for CNN. However, we noticed that the results were more evident only when there was a very large reduction in number of speakers. In scenario with 80% of speakers, the results showed no improvement in CNN performance, despite the reduction in speakers. It was not possible to identify a general pattern of CNN behavior that could contribute to the creation of a prediction equation with the aim of estimating the CNN performance with possible additions of new speaker classes. The two prediction calculations presented similar results, but in the individual comparison of results, the Probabilistic prediction calculation presented better results. In general, we consider CNN performance to be below expectations, considering only the Top 1 ranking, which is when CNN hits the audio speaker, the prediction results were very low and even having only 20% of original speakers did not reach 50% of the average prediction percentage.

5.2.2 Scenarios of Audio File Reduction

Table 5.3 and Figure 5.2 present the CNN prediction results for Original SIWIS scenario and for scenarios with audio file reductions. Results of the 4 audio file reduction scenarios were practically all lower than the Original SIWIS scenario. As the audio files reduction implies a smaller amount of samples, we imagined that the result would be a progressive

Tabela 5.2: Comparison of average prediction percentages between the scenario trained with Original SIWIS and the speaker reduction scenarios.

Scenario	Perc. Speak.	Nº of Speak.	Lang.	Nº of Files	Predict.	Top 5	Top 3	Top 1	Top 1 > 65%	Top 1 > 99%
Original SIWIS	100%	36	EN,FR, GE,IT	1,473	Class Prob.	98.17% 94.23%	56.75% 56.48%	19.55% 20.1%	0.95% 1.9%	0.07% 0.07%
Speaker Reduction	80%	29	EN,FR, GE,IT	1,233	Class Prob.	97% 91.97%	52.96% 52.64%	20.92% 19.79%	1.46% 4.7%	0.08% 0%
	60%	22		942	Class Prob.	96.5% 93.42%	52.44% 54.78%	23.35% 26.33%	2.44% 7.43%	0.53% 0.11%
	40%	14		617	Class Prob.	94.81% 95.3%	62.07% 65.15%	32.74% 35.66%	5.67% 12.48%	0.49% 0.32%
	20%	7		306	Class Prob.	98.69% 99.02%	70.59% 80.39%	32.35% 43.79%	18.95% 23.2%	11.76% 10.78%

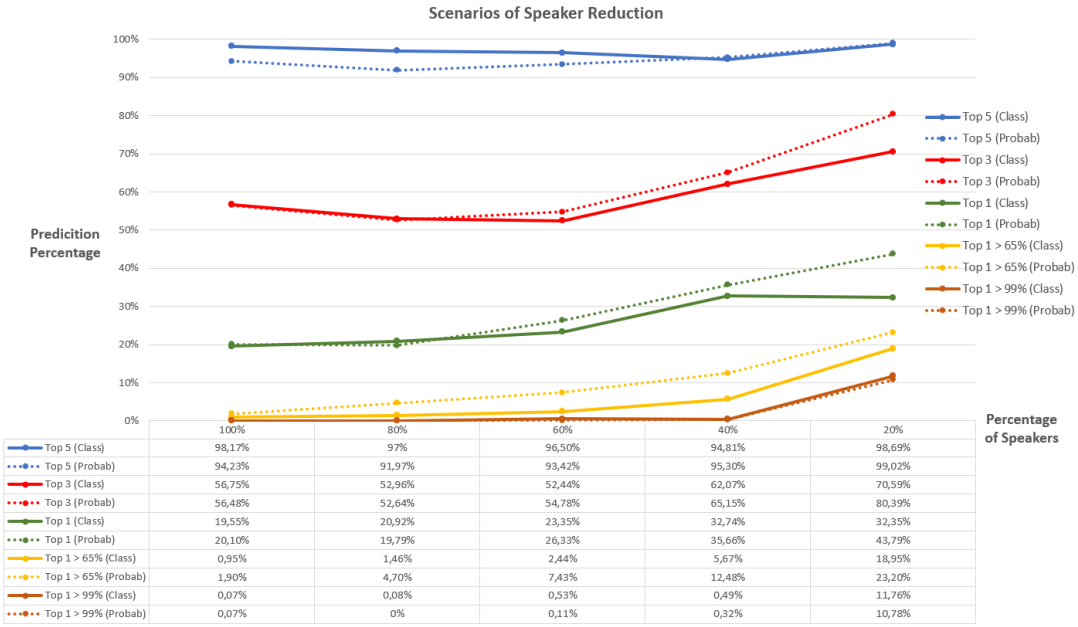
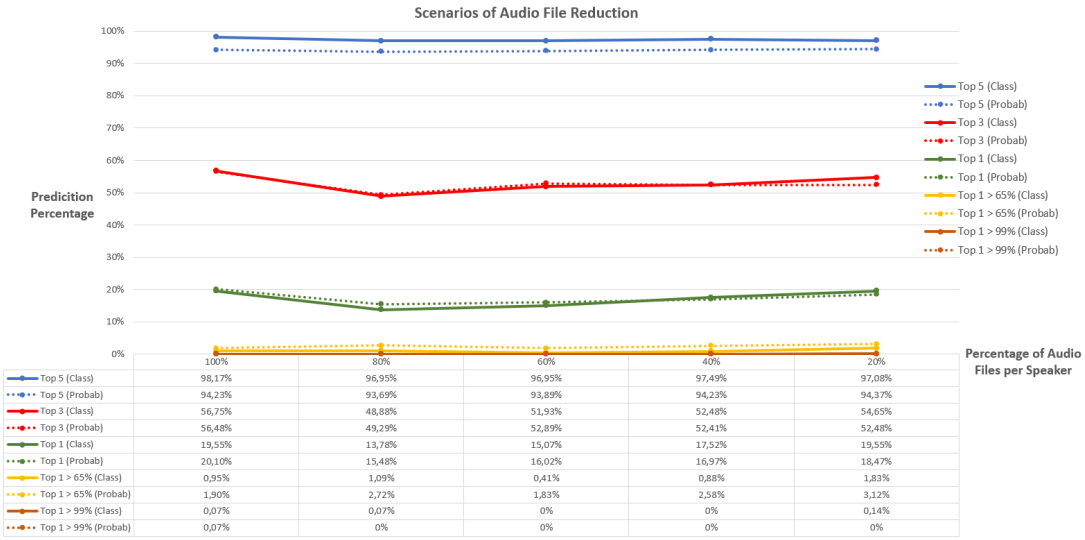


Figura 5.1: Comparison of CNN prediction results, in 5 rankings, involving scenarios of speaker reductions and the Original SIWIS.

decrease in CNN performance in relation to the reference scenario. But that's not what happened. Analyzing Top 3 and Top 1 rankings, we initially see the scenario with 80% of audio files showing a reduction in CNN performance, but considering the other scenarios, as the amount of audio files decreases, CNN performance improves, getting closer to the reference scenario. Unfortunately it was not possible to draw an exact definition of CNN behavior in face of audio file reductions, but through the results we can see that the drastic sample reduction did not cause a drastic drop in CNN performance. As we observed in speaker reduction scenarios, average prediction percentages showed poor CNN performance, especially in Top 1 ranking.

Tabela 5.3: Comparison of average prediction percentages between the scenario trained with Original SIWIS and the audio file reduction scenarios.

Scenario	Perc. Audio	Nº of Speak.	Lang.	Nº of Files	Predict.	Top 5	Top 3	Top 1	Top 1 > 65%	Top 1 > 99%
Original SIWIS	100%	36	EN,FR, GE,IT	1,473	Class	98.17%	56.75%	19.55%	0.95%	0.07%
					Prob.	94.23%	56.48%	20.1%	1.9%	0.07%
Audio File Reduction	80%	36	EN,FR, GE,IT	1,473	Class	96.95%	48.88%	13.78%	1.09%	0.07%
					Prob.	93.69%	49.29%	15.48%	2.72%	0%
	60%				Class	96.95%	51.93%	15.07%	0.41%	0%
					Prob.	93.89%	52.89%	16.02%	1.83%	0%
	40%				Class	97.49%	52.48%	17.52%	0.88%	0%
					Prob.	94.23%	52.41%	16.97%	2.58%	0%
	20%				Class	97.08%	54.65%	19.55%	1.83%	0.14%
					Prob.	94.37%	52.48%	18.47%	3.12%	0%



	100%	80%	60%	40%	20%
Top 5 (Class)	98,17%	96,95%	96,95%	97,49%	97,08%
Top 5 (Probab)	94,23%	93,69%	93,89%	94,23%	94,37%
Top 3 (Class)	56,75%	48,88%	51,93%	52,48%	54,65%
Top 3 (Probab)	56,48%	49,29%	52,89%	52,41%	52,48%
Top 1 (Class)	19,55%	13,78%	15,07%	17,52%	19,55%
Top 1 (Probab)	20,10%	15,48%	16,02%	16,97%	18,47%
Top 1 > 65% (Class)	0,95%	1,09%	0,41%	0,88%	1,83%
Top 1 > 65% (Probab)	1,90%	2,72%	1,83%	2,58%	3,12%
Top 1 > 99% (Class)	0,07%	0,07%	0%	0%	0,14%
Top 1 > 99% (Probab)	0,07%	0%	0%	0%	0%

Figura 5.2: Comparison of CNN prediction results, in 5 rankings, involving scenarios of audio file reductions and the Original SIWIS.

5.2.3 Scenarios of Variations in Audio File Size

CNN prediction results for Original SIWIS scenario and for scenarios with variation in audio file sizes are demonstrated in Table 5.4 and Figure 5.3. Top 5 ranking results are very close in the 4 scenarios, not favoring this analysis. Percentages presented in Top 3 and Top 1 rankings, once again, are the ones that show the greatest variation, allowing for the analysis. Through them we see that the two scenarios with reduction in audio file sizes influenced the loss of CNN performance, when compared to the reference scenario. But the two scenarios with increasing audio file sizes had different results. In scenario with a 25% increase in the audio file size, CNN improved its performance in relation to the two previous scenarios, which had reductions. But in relation to the reference

scenario, despite the amount of samples having increased, there was no improvement in CNN performance. The scenario with the biggest increase in the audio file sizes, with a growth of 50%, showed the worst result among all the scenarios. Regarding the Top 1 $>$ 65% and Top 1 $>$ 99% rankings, they showed insignificant results, not favoring this analysis. Therefore, we see that reducing the audio file sizes, which means reducing the number of samples, influenced the loss of CNN performance. But scenarios with added samples showed no improvement. Files growth was 25% and 50% duplication of their final parts. In this case, we supose that the addition of samples already known does not contribute to CNN learning.

Tabela 5.4: Comparison of average prediction percentages between the scenario trained with Original SIWIS and scenarios of variations in audio file sizes.

Scenario	**Size Perc.**	**Nº of Speak.**	**Lang.**	**Nº of Files**	**Predict.**	**Top 5**	**Top 3**	**Top 1**	**Top 1 > 65%**	**Top 1 > 99%**
Original SIWIS	100%	36	EN,FR, GE,IT	1,473	Class	98.17%	56.75%	19.55%	0.95%	0.07%
					Prob.	94.23%	56.48%	20.1%	1.9%	0.07%
Variation in Audio File Size	50%	36	EN,FR, GE,IT	1,473	Class	96.81%	51.19%	16.84%	2.72%	0.41%
					Prob.	92.26%	50.64%	16.63%	3.26%	0.07%
	75%				Class	97.15%	53.09%	18.67%	0.54%	0%
					Prob.	94.84%	52.68%	18.13%	2.17%	0%
	125%				Class	96.33%	54.65%	19.14%	1.15%	0.14%
					Prob.	92.94%	53.7%	19.28%	2.44%	0.07%
	150%				Class	96.88%	51.39%	14.05%	0.48%	0%
					Prob.	93.69%	51.53%	15.95%	1.77%	0%

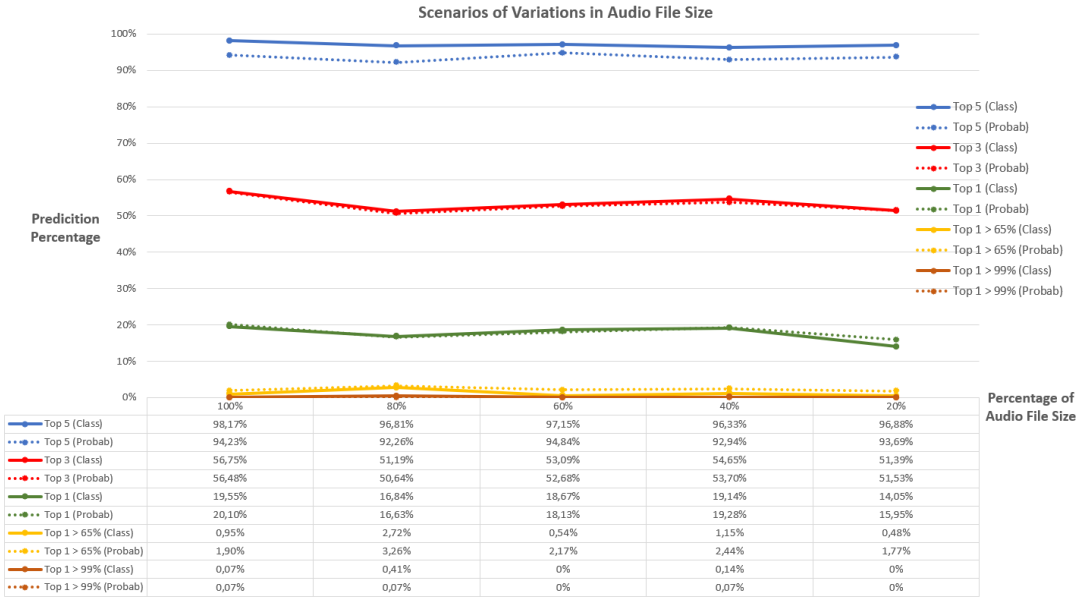


Figura 5.3: Comparison of CNN prediction results, in 5 rankings, involving scenarios of variations in audio file sizes and the Original SIWIS.

5.2.4 Scenarios of SI task using an Unknown Language (German)

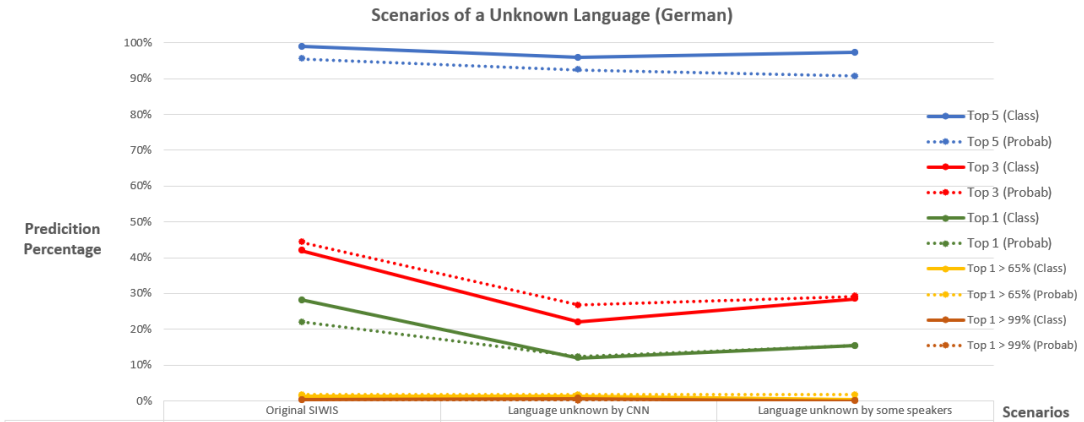
Table 5.5 and Figure 5.4 present the average prediction percentages for Original SIWIS scenario and two other scenarios that use a language unknown by CNN. The construction of these two scenarios was explained in Chapter 4.4.4. Observing the results, we see that only the Top 3 and Top 1 rankings show considerable differences between the 3 scenarios analyzed. In other 3 rankings the average results are very close for the 3 scenarios. In scenario where the German is totally excluded, the average prediction percentages for SI dropped by almost half for Top 3 and Top 1 rankings. In scenario where the German language is partially excluded, although approximately half of the German-speaking speakers were kept, the drop in average prediction percentages was large, approximately 35% in relation to the Original SIWIS scenario. These results raise suspicion about the possible interference the language unknown by CNN may have caused during the SI. In other words, during CNN training, possibly in addition to the speaker features, the learning of language features would also influence CNN's better performance in SI task. During CNN training, the strategic function of capturing the speaker features was performed by the MFCC. Therefore, possibly a large share of responsibility for the results presented in this scenario is related to the MFCC.

Tabela 5.5: Comparison of average prediction percentages between the scenario trained with Original SIWIS and the SI scenarios using an unknown language (German).

Scenario	Scenario Nº	Nº of Speakers	Lang.	Nº of Files	Predict.	Top 5	Top 3	Top 1	Top 1 > 65%	Top 1 > 99%
Original SIWIS	Reference	17	GE	291	Class	98.97%	41.92%	28.18%	1.37%	0.34%
					Prob.	95.53%	44.33%	21.99%	1.72%	0.34%
Language totally unknown by CNN	4.1	17	GE	291	Class	95.88%	21.99%	12.03%	1.37%	0.69%
					Prob.	92.44%	26.80%	12.37%	1.72%	0%
Language partially unkonwn by CNN	4.2	17	GE	291	Class	97.25%	28.52%	15.46%	0.34%	0%
					Prob.	90.72%	29.21%	15.46%	1.72%	0%

5.2.5 Scenario of Adding a New Speaker Class

Table 5.6 and Figure 5.5 present the prediction results for the Original SIWIS scenario and the scenario where one more speaker class was added to the SIWIS. The included speaker is labeled LEO and speaks three languages. As SIWIS is composed of speakers who speak two or three languages, Table 5.6 and Figure 5.5 show average prediction percentages grouped in speakers with 2 and 3 languages. New speaker LEO's individual results were also presented for comparison. Top 3 and Top 1 rankings are the ones that best highlight the differences between the scenario results. But for the new speaker, the



	Original SIWIS	Language unknown by CNN	Language unknown by some speakers
Top 5 (Class)	98,97%	95,88%	97,25%
Top 5 (Probab)	95,53%	92,44%	90,72%
Top 3 (Class)	41,92%	21,99%	28,52%
Top 3 (Probab)	44,33%	26,80%	29,21%
Top 1 (Class)	28,18%	12,03%	15,46%
Top 1 (Probab)	21,99%	12,37%	15,46%
Top 1 > 65% (Class)	1,37%	1,37%	0,34%
Top 1 > 65% (Probab)	1,72%	1,72%	1,72%
Top 1 > 99% (Class)	0,34%	0,69%	0%
Top 1 > 99% (Probab)	0,34%	0%	0%

Scenarios

Figura 5.4: Comparison of CNN prediction results, in 5 rankings, between scenarios of a language totally and partially unknown by CNN and the Original SIWIS.

values of all rankings are high. Comparing scenarios results, before and after the inclusion of the new speaker, it is noticeable that after the inclusion there was a slight decrease in CNN performance for both groups of 2 and 3 language speakers. Observing now the new speaker's individual results we see a great performance achieved by CNN for the identification of this new speaker. This good performance contribution was not enough to raise CNN's average prediction percentage. Tables 5.7 and 5.8 present individual results of 10 speakers with the highest average prediction percentage in Top 1 ranking, evaluated before and after the addition of the new speaker. Comparing these results we can see that most speakers had a worsening in their prediction percentage, but for some speakers (25 and 21) there was a small improvement after the inclusion of one more speaker. In Top 1 $>$ 65% ranking, some improvements also occurred for other speakers, in Probabilistic Prediction calculation, but with percentages still much lower than those presented by the new speaker LEO. Therefore, the addition of a new speaker class in this SI solution causes a decrease in CNN performance when predicting the speaker classes that already existed in SIWIS dataset. But for this new speaker, CNN performance showed extraordinary SI results, far above the prediction results presented for the original speakers from SIWIS dataset.

Tabela 5.6: Comparison of average prediction percentages between the scenario trained with Original SIWIS and the scenario of adding a new speaker class.

Scenario	Scen. Nº	Nº of Speakers	Nº of Lang.	Nº of Files	Predict.	Top 5	Top 3	Top 1	Top 1 > 65%	Top 1 > 99%
Original SIWIS	Ref.	22	2	755	Class	98.82%	52.63%	10.96%	0.26%	0%
					Prob.	96.12%	52.09%	14.99%	1.73%	0%
		14	3	718	Class	97.51%	61.06%	28.65%	1.67%	0.14%
					Prob.	92.20%	61.09%	25.60%	2.10%	0.14%
Adding a new speaker class	5	22	2	755	Class	99.21%	51.04%	9.15%	0.66%	0%
					Prob.	94.94%	49.41%	13.31%	1.20%	0%
		15	3	770	Class	96.13%	58.76%	22.54%	0.98%	0%
					Prob.	92.20%	57.92%	21.84%	3.36%	0%
Adding a new speaker class (only new speaker results)	5	1	3	52	Class	100%	100%	100%	96.15%	71.15%
					Prob.	100%	100%	100%	90.38%	51.92%

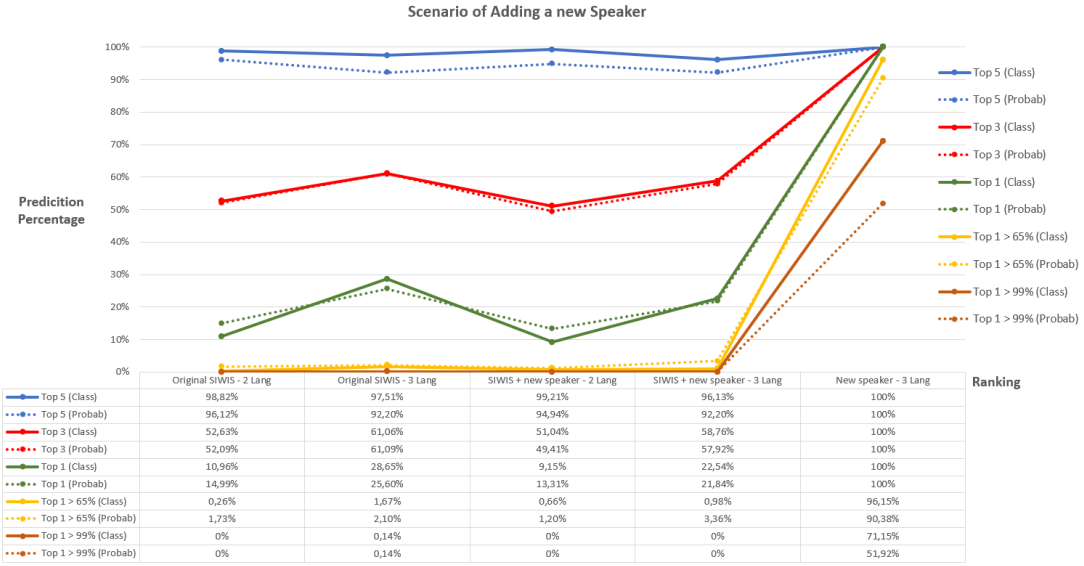


Figura 5.5: CNN prediction results, in 5 rankings, for speakers who speak 2 and 3 languages, before and after the addition of the new speaker.

5.2.6 Scenario of SI task using another Unknown Language (Portuguese) for the New Speaker Class

In this scenario, Table 5.9 and Figure 5.6 present the prediction results just for new speaker LEO. Audios were processed using three known languages and one new language unknown by CNN (Portuguese). Results were very good, in general, presenting high prediction percentages. CNN's prediction of unknown-language audios was practically on par with the other three known languages. Comparing number by number, we see in Top 1 $>$ 65% ranking the new language is below the other languages and in Top 1 $>$ 99% this difference increases in relation to some known languages. It is possible to say despite the differences identified, CNN performance remained stable with the unknown language, as the small differences between the languages occurred only in the two most rigorous

Tabela 5.7: List of 10 speakers with the highest average prediction percentage in Top 1 ranking (speaker correct prediction), before adding the new speaker.

Scenario	Sc. Nº	Nº	Speak.	Nº Lang.	Lang.	Nº File	Predict.	Top 5	Top 3	Top 1	Top 1 > 65%	Top 1 > 99%
Original SIWIS	Ref.	1	39	2	FR, IT	34	Class Prob.	100% 100%	97.06% 100%	44.12% 97.06%	2.94% 32.35%	0% 0%
		2	36	3	EN, GE, IT	51	Class Prob.	98.04% 94.12%	86.27% 90.2%	74.51% 72.55%	0% 3.92%	0% 0%
		3	06	3	EN, FR, GE	51	Class Prob.	100% 98.04%	80.39% 82.35%	56.86% 66.67%	1.96% 0%	0% 0%
		4	03	3	EN, FR, GE	51	Class Prob.	100% 92.16%	100% 78.43%	70.59% 41.18%	7.84% 5.88%	0% 0%
		5	40	2	FR, IT	34	Class Prob.	97.06% 100%	64.71% 91.18%	17.65% 50%	0% 2.94%	0% 0%
		6	14	2	FR, IT	35	Class Prob.	97.14% 100%	71.43% 91.43%	48.57% 51.43%	0% 0%	0% 0%
		7	05	3	EN, FR, GE	51	Class Prob.	92.16% 96.08%	56.86% 64.71%	41.18% 41.18%	3.92% 5.88%	0% 0%
		8	35	3	EN, FR, GE	51	Class Prob.	100% 98.04%	62.75% 76.47%	21.57% 43.14%	1.96% 5.88%	0% 0%
		9	25	3	EN, FR, IT	51	Class Prob.	98.04% 84.31%	68.63% 56.86%	19.61% 21.57%	0% 0%	0% 0%
		10	21	2	GE, IT	33	Class Prob.	100% 84.85%	42.42% 30.3%	12.12% 21.21%	0% 0%	0% 0%

Tabela 5.8: List of 11 speakers with the highest average prediction percentage in Top 1 ranking (speaker correct prediction), after adding the new speaker.

Scenario	Sc. Nº	Nº	Speak.	Nº Lang.	Lang.	Nº File	Predict.	Top 5	Top 3	Top 1	Top 1 > 65%	Top 1 > 99%
Adding a new speaker class	5	1	LEO	3	EN, FR, IT	52	Class Prob.	100% 100%	100% 100%	100% 100%	96.15% 90.38%	71.15% 51.92%
		2	39	2	FR, IT	34	Class Prob.	97.06% 100%	91.18% 97.06%	35.29% 91.18%	0% 17.65%	0% 0%
		3	36	3	EN, GE, IT	51	Class Prob.	94.12% 94.12%	80.39% 82.35%	49.02% 74.51%	0% 13.73%	0% 0%
		4	06	3	EN, FR, GE	51	Class Prob.	100% 94.12%	90.2% 84.31%	52.94% 66.67%	0% 0%	0% 0%
		5	03	3	EN, FR, GE	51	Class Prob.	100% 98.04%	98.04% 86.27%	66.67% 23.53%	3.92% 5.88%	0% 0%
		6	40	2	FR, IT	34	Class Prob.	97.06% 91.18%	52.94% 76.47%	8.82% 47.06%	0% 2.94%	0% 0%
		7	14	2	FR, IT	35	Class Prob.	100% 100%	82.86% 88.57%	45.71% 40%	5.71% 5.71%	0% 0%
		8	05	3	EN, FR, GE	51	Class Prob.	90.2% 100%	49.02% 54.9%	27.45% 39.22%	1.96% 7.84%	0% 0%
		9	35	3	EN, FR, GE	51	Class Prob.	98.04% 98.04%	64.71% 74.51%	35.29% 37.25%	3.92% 13.73%	0% 0%
		10	25	3	EN, FR, IT	51	Class Prob.	100% 90.2%	72.55% 68.63%	21.57% 21.57%	1.96% 3.92%	0% 0%
		11	21	2	GE, IT	33	Class Prob.	100% 84.85%	45.45% 33.33%	18.18% 27.27%	0% 0%	0% 0%

rankings. Supposedly, this slight drop in CNN performance is really due to the fact that the language is unknown. But in this scenario, it can be said the unknown language did not compromise CNN performance in SI task.

Tabela 5.9: Comparison of average prediction percentages using languages already known by the new speaker and another unknown language (Portuguese).

Scenario	Scenario Nº	Speaker	Lang.	Nº of Files	Predict.	Top 5	Top 3	Top 1	Top 1 > 65%	Top 1 > 99%
Adding a new speaker class	5	LEO	EN	17	Class Prob.	100% 100%	100% 100%	100% 100%	88.24% 82.35%	82.35% 29.41%
		LEO	FR	18	Class Prob.	100% 100%	100% 100%	100% 100%	100% 94.44%	55.56% 66.67%
		LEO	IT	17	Class Prob.	100% 100%	100% 100%	100% 100%	100% 94.12%	76.47% 58,82%
SI using another unknown language for the new speaker class	6	LEO	PO	10	Class Prob.	100% 100%	100% 100%	100% 100%	80% 90%	50% 20%

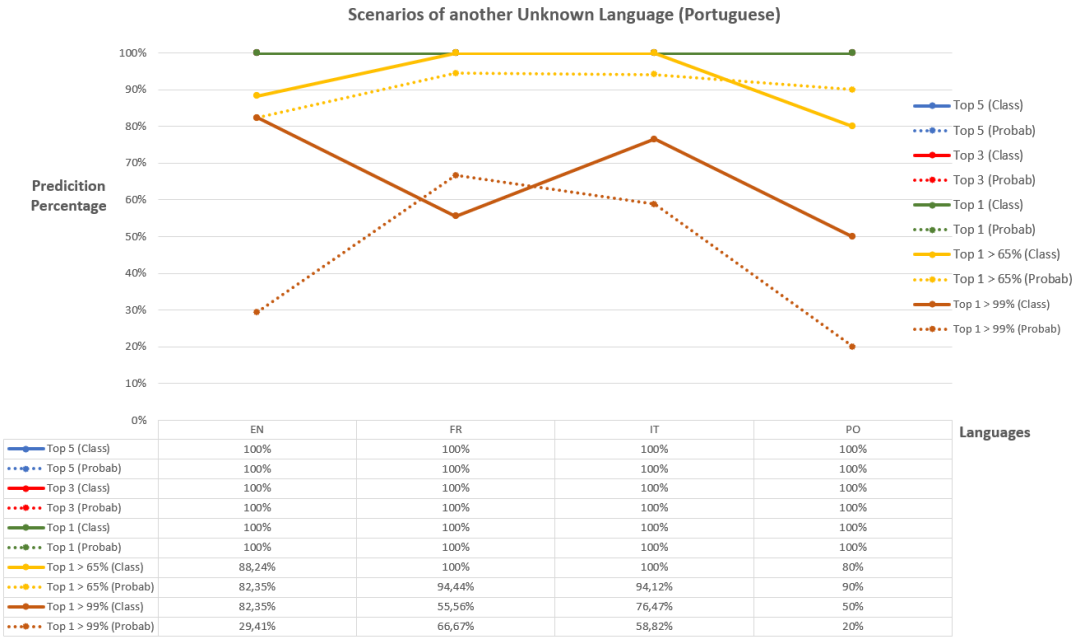


Figura 5.6: CNN prediction results for the new speaker, in 5 rankings, using audios in the 3 known languages and in the unknown language (Portuguese).

5.3 Final Remarks

This Chapter presents the final remarks about the results of the experimental scenarios. CNN training results for the experimental scenarios, in general, presented good Accuracy and F1 values, which are in line with our expectations. But comparing these results with each other, we were able to identify some unexpected situations, as detailed in Chapter 5.1. Comparing the validation and testing values of each scenario, we see that they were almost identical for all scenarios, with no drop in CNN performance during the tests, remaining stable. Now looking at results of experimental scenarios and making a general comparison of the results presented by the two prediction calculations, it is possible to say that their results are very similar. In a few situations there were major differences

between the two. Comparing the two calculations, number with number, we see the first 5 experimental scenarios presented Probabilistic Prediction results slightly higher than Class Prediction results in Top 1 rankings, which are the ones who point out the correct speaker in the first place. In scenario 6, which analyzes only one speaker, the results showed a very high CNN performance and most of them were 100% of prediction in both calculations. Comparing the two calculations in this scenario 6, Class Prediction presented better results in relation to Probabilistic Prediction. We consider as a result of comparison between the two prediction calculations that both were useful for the analyses, presented reliable and similar results, and there are possibilities of using both in some SI solution.

CNN performance in 5 rankings were analyzed. Top 5 ranking results remained stable in all 6 scenarios, not showing average prediction percentages with values lower than 90% for the speakers' collective results. This strongly demonstrates the correct speaker was among the 5 speakers with the highest prediction. Despite the stability in Top 5 results and their percentages above 90%, the average prediction percentage of 100% was expected to occur many times, but it did not happen in speakers' collective results, only in speakers' individual results, presented in scenarios 5 and 6. Top 3 ranking, despite having presented the average prediction percentages much lower than Top 5 ranking, was widely used by the analyzes because its variation allowed the comparison between the scenarios results. Its average prediction result with the highest value was 80.39%, in Speaker Reduction scenario, being mostly between 50% and 60%, in scenarios that had the 36 speakers. Its lowest average prediction percentage was 21.99% in 4.1 scenario, when the German language was completely removed from the dataset. In scenarios 5 and 6, Top 3 ranking achieved 100% with just the new speaker LEO, in all his results. Top 1 ranking, just like Top 3 ranking, served to compare the scenarios due to the variations in its results, but presented very low average prediction percentages. It is the ranking that counts when CNN correctly predicts the audio speaker as the most rated. Its highest average predictions were 43.79% in speaker reduction scenario and 20.1% with 36 speakers in Original SIWIS scenario. Its lowest average prediction was in scenario 5, which adds a new speaker to the dataset, presenting 9.15% as average prediction for speakers who speak 2 languages. In the individual results, shown in scenarios 5 and 6, only the new speaker LEO achieved 100% in Top 1 ranking, on both prediction calculations. As for the other speakers, only 4 were highlighted (speakers 39, 36, 06 and 03) who obtained prediction percentages above 50%. Top 1 $>$ 65% ranking verifies when CNN hits the audio speaker, presenting a prediction percentage greater than 65%. The highest average prediction per-

centages were 23.2% in speaker reduction scenario and 3.26% with 36 speakers in audio file size variation scenario. Its worst prediction performance was 0.41%, with 36 speakers, in audio file reduction scenario. These results showed a very low performance for CNN. In the individual results, in general, Top 1 > 65% ranking also presented very low prediction percentages, showing great oscillation between the two prediction calculations and also between the moments before and after adding the new speaker. Many speakers had 0% prediction in their results. Analyzing the prediction of the new speaker, on the other hand, CNN presented a very high result, standing out among other 36 speakers, scoring 96.15% and 90.38% as average prediction percentage in class and probabilistic prediction calculations. Top 1 > 99% ranking is when CNN correctly predicts the speaker unanimously. This ranking did not contribute much to the analysis as the average prediction percentages of the 36 original speakers was 0%. Only the new speaker presented significant and very high percentages for this ranking, obtaining 71.15% and 51.92% in class and probabilistic prediction calculations.

Analyzing the four experimental Speaker Reductions scenarios from Chapter 5.2.1, we see that there was a progressive increase in CNN's average prediction percentages as the number of speakers decreased. Results confirmed the suspicion that the number of speaker classes influence CNN performance. Regarding the numbers presented, CNN performance did not present high results, especially if we consider the Top 1 ranking, the one where the speaker is correctly predicted. Results of the Audio File Reductions scenarios, from Chapter 5.2.2, showed a very different situation from what we assumed would happen. In each scenario, the amount of audio files is reduced more and more, and so it was expected that CNN performance would also decrease more and more. CNN performance drop happened initially, in scenario of 80% audio files. In following scenarios, with fewer and fewer audio files, CNN performance improved. And in the last scenario, with 20% of audio files, CNN achieved the same performance as the original scenario. Unfortunately, it was not possible to draw a conclusive analysis of CNN performance taking into account the evolution of the four scenarios. We only verified that, in the scenario containing 20% of the original number of audio files, it was possible to obtain the same performance as the CNN trained with the original dataset. Which means that a very large number of samples will not necessarily provide a big increase in CNN performance. Scenarios with variations in Audio File Size, from Chapter 5.2.3, showed results very different from what was expected. Two scenarios had reductions in audio file sizes and two had increases, with repetition of their final parts. Scenarios with reduced audio file sizes clearly show a reduction in CNN performance compared to the reference scenario. But the two scenarios with

increasing showed no performance gains. In this case, we understand that the addition of repeated samples did not favor CNN learning. In experimental scenarios that evaluated the CNN performance using German as an unknown language, from Chapter 5.2.4, the performance decrease was clear. In scenario that partially excludes German (scenario 4.2), 8 of the 17 speakers had their language excluded. CNN performance in this scenario showed a very large drop, approximately 40% in relation to the reference scenario, in Top 1 ranking. When the total exclusion of German occurs (scenario 4.1), this percentage decreases to approximately 50% in relation to the reference scenario, in Top 1 ranking. Therefore, in this scenario, it was evident the exclusion of a language greatly influenced CNN performance drop when executing the SI task.

In scenario 5, from Chapter 5.2.5, where a new speaker who speaks three languages was added to the SIWIS dataset, results of this new speaker are presented and compared with results of the original speakers who speak 2 and 3 languages, belonging to the same dataset and the original dataset (Table 5.6). After the addition of the new speaker, CNN performance decreased on the SI task for the original speakers. But for the new speaker, CNN performance proved to be spectacular, presenting a prediction result not yet seen with the original speakers. In no other result had the Top 1 ranking $> 65\%$ reached prediction values above 90%, as presented for the new speaker. In Top 1 $> 99\%$ ranking, which checks when there is a unanimous prediction for a single speaker, the prediction values had not yet reached 1% in scenarios with 36 speakers and reached 71% for the new speaker identification. In individual comparisons, shown in Table 5.8, the 11 speakers with the best results in Top 1 ranking after the addition of the new speaker were listed. Results showed that CNN performance when executing the new speaker's prediction is much superior in relation to the other speakers. CNN performance to identify the new speaker LEO can be considered excellent. Unfortunately, we were unable to discover what influenced the occurrence of such a large difference between the prediction results of the new speaker and the other 36 speakers. Even the phrases of the 3 languages, used for recording the new speaker LEO's audios, were the same used by speakers already existing in SIWIS dataset. When listening to the new speaker's audios and the other 36 speakers' audios, it was not possible to identify differences in quality between them. Even though these differences are imperceptible to human hearing, there is supposedly some issue related to the recording time of the new speaker's audios that may have provided more quality or technical compatibility with the CNN used, favoring its performance when executing the SI task for the new speaker. The investigation necessary to discover the motivating factors of these differences in results was not included in planning of this work.

Supposedly, factors such as the recording environment, the equipment and technologies used, the quality of the speaker's speech, the influence of other sounds, may be related to these differences in results.

In the last scenario, CNN performance is tested using audios for the new speaker in another unknown language: Portuguese. CNN performance was very good because the SI with the unknown language presented prediction results at the same level as the other three languages known by CNN. Only in Top 1 $>$ 99% ranking was it possible to see greater differences in SI results between the unknown and the known languages. We believe that the language spoken by the speaker in the audio may exert some influence on his identification by CNN. In scenario 4, using German as an unknown language for a larger number of speakers, this influence could be better noticed in the average of CNN performance results. But we suppose that it was not so evident for the new speaker due to CNN good performance in identifying this speaker, as shown by the results of scenarios 5 and 6.

The roadmap created to execute the Experimental Plan, using the representation of business processes, aimed to perform the CNN robustness analysis in a standardized and organized way. The planning really worked out because the experimental scenarios followed the same pattern of execution, from CNN training, to fulfill the requirements of each scenario, until the execution of experimental tests with the CNN model. In addition, the roadmap followed a standard business process notation that allows an easy understanding of all steps. MFCC method fulfilled its function in capturing the speakers features but we could not evaluate its contribution to CNN performance. To achieve this, in future research it would be interesting to perform this same robustness analysis using others feature extraction methods identified by SLR. The SIWIS dataset of multi-lingual speakers contributed greatly to the Experimental Plan creation, enabling the elaboration of varied scenarios that explored very different situations for CNN performance evaluation. It is an interesting audio dataset with possibilities of use in future research related to multi-lingual speakers. And yet, a reservation must be made in this statement. Unfortunately, the CNN architecture used by this work and trained with the SIWIS dataset to perform the SI task, did not perform well when predicting the original speakers from SIWIS dataset. CNN good performance was only achieved when a new speaker, which did not exist in SIWIS, was inserted after creating its audios using the same languages and sentences as the speakers already existing in the SIWIS dataset. Regarding the CNN architecture used, it is a little complex architecture compared to some ANN architectures identified by the SLR that showed great complexity and greater amounts of layers. It was originally

presented by article [26] performing the SI task for noisy audio. Therefore, being trained with a dataset that contains multi-language speakers, it showed that it can be used for other SI problems.

In conclusion, CNN's average prediction results in scenarios 1 to 4 showed very poor performance in Top 1 rankings. But in scenarios 5 and 6, with the addition of the new speaker, the individual prediction results showed that CNN performance was excellent in Top 1 rankings, reaching maximum percentages in prediction of the new speaker. Still in scenarios 5 and 6, we could see the addition of a new speaker influenced an even greater decrease in CNN performance, in prediction of the other existing speakers. This performance difference is possibly related to some technical or quality difference between the new audios recorded for the new speaker and the original speakers' audios from the SIWIS dataset. These technical and quality comparisons between the audios could not be investigated in this work. Regarding the evaluation of CNN performance in SI task, we rely only on scenarios 5 and 6, to say that CNN can achieve very good prediction results in SI, but the ideal characteristics of the audios to obtain this great performance still need to be clarified in future work. Despite the poor prediction performance in scenarios 1 to 4, during the analysis of the experimental scenarios we took into account the variation of their results when comparing them. In scenario 1, Speaker Reduction, it was clear there was a decrease in CNN performance as the number of speakers increased. In scenario 2, Audio File Reductions, it was shown that a minimum amount of audio files allowed CNN to learn the speakers' features and keep its performance practically equal to the scenario with the original number of speakers. In scenario 3, Variation in audio file sizes, both the removal and the addition of audio sizes showed a worsening in CNN performance, with no contributions. In scenario 4, where CNN performs SI for an unknown language, the average prediction results showed an evident performance drop when the language is unknown to CNN, giving the impression of a great influence. But this drop in CNN performance did not happen in the individual prediction results of scenario 6, used for the new speaker, where another unknown language was used. These differences reinforce the existence of a big difference between the new speaker's audio and the original speakers' audios from the SIWIS dataset, showing that the new speaker's audios present a better quality for learning by CNN.

Capítulo 6

Conclusions

The SFSP theme and the ANN computational model are two prominent subjects in the academic universe and that present concrete contributions to the technological evolution of machine learning and the automation of speech processing solutions. There was a motivation to deepen the knowledge on these two very relevant subjects and the best way we found was to carry out an SLR to identify the state of the art in SFSP using ANN. The result produced by SLR was very enriching because in addition to the idenfication of the state of the art, it brought us a lot of other related information. But what caught our attention was the fact that we did not identify any work that was concerned with carrying out an ANN robustness analysis, that is, that evaluated how much the performance of an ANN executing an SFSP task remains minimally sensitive to factors that cause variability [108] and [143]. The absence of this type of work in the analyzed literature was a new motivation for its accomplishment, as it is a contribution to the literature and for the understanding that robustness is a very relevant property for a computational model. Considering the wide variety of research identified by the SLR, we found a relevant dataset that contains speakers who speak multi-languages, presented by article [14]. We also identified a CNN, whose article [26] was well ranked by QA carried out during the SLR and which we believe was a very good architecture used for this work. Given this scenario and considering the motivational factors mentioned, we decided to produce this robustness analysis work, evaluating the performance of this CNN, based on an experimental plan that explored the SI task, using multi-language scenarios.

One of the specific objectives of this work was to identify the state of the art related to SFSP using ANN, through a SLR. The objective was fulfilled very satisfactorily because in addition to identifying the state of the art on SFSP using ANN, a large amount of enriching information related to this topic was learned. Among this information, we highlight the

interesting and varied research problems related to SFSP, the use of a wide variety of ANN models and architectures and different speaker feature extraction methods. We adapted from the literature a process for executing the SLR and created a QA on the selected articles. These procedures were essential for conducting activities in an organized manner, resulting in the identification of the state of the art related to SFSP using ANN. The full details of this SLR were presented in Chapter 2. Another specific objective was: evaluate the execution of an experimental plan to explore the CNN's robustness performing a SI task. An experimental plan was created to contain scenarios that explore the CNN performance using variations of different characteristics related to the SI task. The SIWIS dataset [14] was used for the elaboration of different scenarios in the experimental plan. Dataset originally contains 36 speakers who speak 3 or 2 languages out of 4 existing languages. Different scenarios addressed variations in the number of speakers, in the number of audio files per speaker and in the size of the speaker's audios. Another 3 scenarios addressed CNN performance when executing SI using unknown languages. For the execution of the experimental plan, research procedures were created that helped a lot in the agility, organization and standardization of the experimental scenarios. Planning, creation and execution of this experimental plan were performed in the best possible way, obtaining good results and becoming a contribution of this work as a roadmap model for the execution of future robustness analyzes for ANNs or other computational models. This objective was also fulfilled very satisfactorily. All of this planning and creation is detailed in Chapter 4.

The main objective of this work was to perform an exploratory analysis of CNN's robustness when performing an SI task in multi-language scenarios. The executed experimental plan, the results of the CNN training and the execution of the experimental scenarios are presented in Chapters 5.1 and 5.2. The analysis of CNN's performance results was detailed in Chapter 5.3, where the results of the scenarios were compared with each other, taking as the main comparison the reference scenario, which contains the original composition of the dataset SIWIS. The final result of this exploratory analysis showed that CNN had its robustness affected differently by each experimental scenario. While for scenario 6, CNN's robustness analysis was practically 100%, having irrelevant reflections on its performance, scenarios 1, 4.1 and 4.2 had a stronger effect on its robustness, making it have a greater variation in its performance. Scenarios 2, 3 and 5 had little effect on its robustness. As a response to the problem and main objective of this work, the exploratory analysis of CNN's robustness was well performed and presented interesting results that contributed to a greater understanding of the impacts related to this SI problem. It was

possible to get an idea of the reflexes that the variation of specific parameters, represented in the experimental scenarios, can cause in the CNN performance. But it is necessary to evolve this analysis model so that it can present a final result on robustness, instead of partial results for each scenario.

Contributions provided by this work are:

- a SLR containing a large amount of relevant and enriching information related to the SFSP using ANN theme, which shows: a variety of research problems specific to the SFSP areas, a large number of ANN models and architectures and speaker feature extraction methods;
- the term SFSP, not found in the literature analyzed by the SLR, coined by this us and which represents speech processing research with a learning focus on speaker features;
- a methodological contribution, due to the strategic way in which the work of setting up the experiments was addressed: the creation of an experimental plan containing different scenarios that explored characteristics of a SI task, and the creation of a roadmap for the execution of the experimental activities, organized in the form of research procedures and performed in a standardized way;
- the execution of the experiments and the presentation of their results, showing the analysis of CNN's robustness when performing a SI task for multilingual speakers.

6.1 Limitations and Future Works

During the experiments, it was evident the great differences between the prediction results of original speakers from the dataset SIWIS and the new speaker inserted in the same dataset. While CNN's prediction values were very low for the original speakers, for the new speaker CNN's performance was exceptional. This situation was unexpected and would need to be investigated to understand the reason for such a big difference in CNN's performance. Another research limitation was the use of Class and Probabilistic prediction calculations instead of some specific metric to evaluate robustness. An opportunity for future work would be the use of proper metrics for robustness analysis after a literature search. Another observation refers to the use of only the selected CNN as the analyzed computational model. There is the possibility of using other types of ANN or other types of computational models in the same experimental plan to compare results between them. In this same vein, only MFCC was used as a speaker feature extraction method. Other

methods presented by the SLR could also be used, becoming one more variable parameter for the experimental plan. Still as a suggestion to increase parameters, variations in the CNN architecture characteristics could be used, such as the number of layers or kernels. Finally, the absence of a method that presents a proposal for a final result for the robustness analysis, representing all the individual results of the experimental scenarios, would be an interesting work that would contribute to the creation of a robustness classification and a comparative robustness ranking between the computer models that were analyzed.

Referências

[1] Abdel-Hamid, O.; Jiang, H. Fast speaker adaptation of hybrid nn/hmm model for speech recognition based on discriminative learning of speaker code. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (2013), IEEE, pp. 7942–7946.

[2] Abdel-Zaher, A. M.; Eldeib, A. M. Breast cancer classification using deep belief networks. *Expert Systems with Applications 46* (2016), 139–144.

[3] Abdulkader, S. N.; Atia, A.; Mostafa, M.-S. M. Authentication systems: Principles and threats. *Computer and Information Science 8*, 3 (2015), 155.

[4] ACM. About ACM DL. Available online: https://dl.acm.org/about (accessed on 01 May 2022).

[5] Albawi, S.; Mohammed, T. A.; Al-Zawi, S. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)* (2017), IEEE, pp. 1–6.

[6] Albrizio, A. Biometry and anthropometry: from galton to constitutional medicine. *Journal of Anthropological Sciences 85* (2007), 101–123.

[7] Antony, A.; Gopikakumari, R. Speaker identification based on combination of mfcc and umrt based features. *Procedia computer science 143* (2018), 250–257.

[8] Ball, E. W.; Blachman, B. A. Phoneme segmentation training: Effect on reading readiness. *Annals of Dyslexia 38*, 1 (1988), 208–225.

[9] Baxter, B.; Malak, R. Increasing system robustness through a utility-based analysis. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (2013), vol. 55867, American Society of Mechanical Engineers, p. V02BT02A020.

[10] Beigi, H. Speaker recognition. In *Fundamentals of Speaker Recognition*. Springer, 2011, pp. 543–559.

[11] Bhatia, S. Systematic review of biometric advancement and challenges. *Int. J. Electron. Eng 11*, 1 (2019), 812–821.

[12] Bhattacharya, G.; Alam, M. J.; Kenny, P. Deep speaker embeddings for short-duration speaker verification. In *Interspeech* (2017), pp. 1517–1521.

[13] Bhattacharyya, D.; Ranjan, R.; Alisherov, F.; Choi, M., et al. Biometric authentication: A review. *International Journal of u-and e-Service, Science and Technology 2*, 3 (2009), 13–28.

[14] Bianco, S.; Cereda, E.; Napoletano, P. Discriminative deep audio feature embedding for speaker recognition in the wild. In *2018 IEEE 8th International Conference on Consumer Electronics-Berlin (ICCE-Berlin)* (2018), IEEE, pp. 1–5.

[15] Boccato, V. R. C. Metodologia da pesquisa bibliográfica na área odontológica e o artigo científico como forma de comunicação. *Rev. Odontol. Univ. Cidade São Paulo, São Paulo 18*, 3 (2006), 265–274.

[16] Boithias, F.; El Mankibi, M.; Michel, P. Genetic algorithms based optimization of artificial neural network architecture for buildings' indoor discomfort and energy consumption prediction. In *Building Simulation* (2012), vol. 5, Springer, pp. 95–106.

[17] Bordes, A.; Weston, J.; Collobert, R.; Bengio, Y. Learning structured embeddings of knowledge bases. In *Twenty-Fifth AAAI Conference on Artificial Intelligence* (2011).

[18] Boulanger-Lewandowski, N.; Bengio, Y.; Vincent, P. Audio chord recognition with recurrent neural networks. In *ISMIR* (2013), Citeseer, pp. 335–340.

[19] Brereton, P.; Kitchenham, B. A.; Budgen, D.; Turner, M.; Khalil, M. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of systems and software 80*, 4 (2007), 571–583.

[20] Büyük, O.; Arslan, M. L. Combination of long-term and short-term features for age identification from voice. *Advances in Electrical and Computer Engineering 18*, 2 (2018), 101–108.

[21] Campbell, J. P. Speaker recognition: A tutorial. *Proceedings of the IEEE 85*, 9 (1997), 1437–1462.

[22] Carrol, L. Alice's adventures in wonderland. *Basington, UK: Macmillan Publisher* (1865).

[23] Chang, S.; Han, W.; Tang, J.; Qi, G.-J.; Aggarwal, C. C.; Huang, T. S. Heterogeneous network embedding via deep architectures. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2015), ACM, pp. 119–128.

[24] Chelali, F.; Djeradi, A. Audiovisual speaker identification based on lip and speech modalities. *International Arab Journal of Information Technology (IAJIT) 14*, 1 (2017).

[25] Chen, N.; Qian, Y.; Dinkel, H.; Chen, B.; Yu, K. Robust deep feature for spoofing detection—the sjtu system for asvspoof 2015 challenge. In *Sixteenth Annual Conference of the International Speech Communication Association* (2015).

[26] Chowdhury, A.; Ross, A. Extracting sub-glottal and supra-glottal features from mfcc using convolutional neural networks for speaker identification in degraded audio signals. In *2017 IEEE International Joint Conference on Biometrics (IJCB)* (2017), IEEE, pp. 608–617.

[27] Chung, H.; Iorga, M.; Voas, J.; Lee, S. Alexa, can i trust you? *Computer 50*, 9 (2017), 100–104.

[28] Chung, J. S.; Nagrani, A.; Zisserman, A. Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv:1806.05622* (2018).

[29] Cutler, A.; Otake, T. Mora or phoneme? further evidence for language-specific listening. *Journal of memory and language 33*, 6 (1994), 824–844.

[30] Da Silva, E. L.; Menezes, E. M. Metodologia da pesquisa e elaboração de dissertação. *UFSC, Florianópolis, 4a. edição 123* (2005).

[31] de Luis-Garcia, R.; Alberola-Lopez, C.; Aghzout, O.; Ruiz-Alzola, J. Biometric identification systems. *Signal processing 83*, 12 (2003), 2539–2557.

[32] Delac, K.; Grgic, M. A survey of biometric recognition methods. In *Proceedings. Elmar-2004. 46th International Symposium on Electronics in Marine* (2004), IEEE, pp. 184–193.

[33] Dinkel, H.; Qian, Y.; Yu, K. Investigating raw wave deep neural networks for end-to-end speaker spoofing detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 26*, 11 (2018), 2002–2014.

[34] Edquist, C. Identification of policy problems in systems of innovation through diagnostic analysis. In *PRIME Conference, Mexico City* (2008).

[35] Edquist, C. Design of innovation policy through diagnostic analysis: identification of systemic problems (or failures). *Industrial and corporate change 20*, 6 (2011), 1725–1753.

[36] Elsevier. About ScienceDirect | Premier platform for discovering peer-reviewed scientific, technical and medical information | Elsevier. Available online: https://www.elsevier.com/solutions/sciencedirect (accessed on 01 May 2022).

[37] Elsevier. Content - How Scopus Works - Scopus - | Elsevier solutions. Available online: https://www.elsevier.com/solutions/scopus/how-scopus-works/content (accessed on 01 May 2022).

[38] Elsevier. Ei Compendex | Most complete Engineering Database. Available online: https://www.elsevier.com/solutions/engineering-village/content/compendex (accessed on 01 May 2022).

[39] Elsevier. Engineering Research and Resources | Engineering Village Database. Available online: https://www.elsevier.com/pt-br/solutions/engineering-village (accessed on 01 May 2022).

[40] Elsevier. ScienceDirect | Elsevier's leadership information solution | Elsevier. Available online: https://www.elsevier.com/pt-br/solutions/sciencedirect (accessed on 01 May 2022).

[41] Fan, Z.-C.; Jang, J.-S. R.; Lu, C.-L. Singing voice separation and pitch extraction from monaural polyphonic audio music via dnn and adaptive pitch tracking. In *2016 IEEE Second International Conference on Multimedia Big Data (BigMM)* (2016), IEEE, pp. 178–185.

[42] Gavini, F. Jogos olímpicos da grécia antiga: a origem da olimpíada da era moderna, Jun 2020.

[43] Gehring, J.; Miao, Y.; Metze, F.; Waibel, A. Extracting deep bottleneck features using stacked auto-encoders. In *2013 IEEE international conference on acoustics, speech and signal processing* (2013), IEEE, pp. 3377–3381.

[44] Ghaemmaghami, H.; Rahman, M. H.; Himawan, I.; Dean, D.; Kanagasundaram, A.; Sridharan, S.; Fookes, C. Speakers in the wild (sitw): The qut speaker recognition system. In *Proceedings of the 17th Annual Conference of the International Speech Communication Association (ISCA):* (2016), International Speech Communication Association (ISCA), pp. 838–842.

[45] Ghahabi, O.; Hernando, J. Deep learning backend for single and multisession i-vector speaker recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 25*, 4 (2017), 807–817.

[46] Ghani, I.; Yasin, I. Software security engineering in extreme programming methodology: A systematic literature review. *Science International 25*, 2 (2013).

[47] Gil, A. C. Como classificar as pesquisas. *Como elaborar projetos de pesquisa 4*, 1 (2002), 44–45.

[48] Goldman, J.-P.; Honnet, P.-E.; Clark, R.; Garner, P. N.; Ivanova, M.; Lazaridis, A.; Liang, H.; Macedo, T.; Pfister, B.; Ribeiro, M. S., et al. The siwis database: a multilingual speech database with acted emphasis. In *Proceedings of Interspeech* (2016), no. CONF.

[49] Gubin, M. Using convolutional neural networks to classify audio signal in noisy sound scenes. In *2018 Global Smart Industry Conference (GloSIC)* (2018), IEEE, pp. 1–6.

[50] Guo, J.; Xu, N.; Qian, K.; Shi, Y.; Xu, K.; Wu, Y.; Alwan, A. Deep neural network based i-vector mapping for speaker verification using short utterances. *Speech Communication 105* (2018), 92–102.

[51] Gusenbauer, M.; Haddaway, N. R. Which academic search systems are suitable for systematic reviews or meta-analyses? evaluating retrieval qualities of google scholar, pubmed, and 26 other resources. *Research synthesis methods 11*, 2 (2020), 181–217.

[52] Hanilçi, C. Data selection for i-vector based automatic speaker verification antispoofing. *Digital Signal Processing 72* (2018), 171–180.

[53] Hansen, J. H.; Hasan, T. Speaker recognition by machines and humans: A tutorial review. *IEEE Signal processing magazine 32*, 6 (2015), 74–99.

[54] Harris, R. A. *Voice interaction design: crafting the new conversational speech systems*. Elsevier, 2004.

[55] Hasan, M. R.; Jamil, M.; Rahman, M., et al. Speaker identification using mel frequency cepstral coefficients. *variations 1*, 4 (2004).

[56] He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 770–778.

[57] Hilyer, L. A. Basic research methods for librarians, lynn silipigni connaway, ronald r. powell (eds.), libraries unlimited, santa barbara, ca (2010), 317 p. isbn 978-1-59158-865-8, 2011.

[58] Ispirova, G.; Eftimov, T.; Seljak, B. K. Comparing semantic and nutrient value similarities of recipes. In *2019 IEEE International Conference on Big Data (Big Data)* (2019), IEEE, pp. 5131–5139.

[59] Jaffali, S.; Jamoussi, S.; Hamadou, A. B.; Smaili, K. Grouping like-minded users for ratings' prediction. In *International Conference on Intelligent Decision Technologies* (2016), Springer, pp. 3–14.

[60] Jain, A.; Hong, L.; Pankanti, S. Biometric identification. *Communications of the ACM 43*, 2 (2000), 90–98.

[61] Jati, A.; Georgiou, P. G. Speaker2vec: Unsupervised learning and adaptation of a speaker manifold using deep neural networks with an evaluation on speaker segmentation. In *INTERSPEECH* (2017), pp. 3567–3571.

[62] Jati, A.; Georgiou, P. G. An unsupervised neural prediction framework for learning speaker embeddings using recurrent neural networks. In *Interspeech* (2018), pp. 1131–1135.

[63] Jayasankar, T.; Vinothkumar, K.; Vijayaselvi, A. Automatic gender identification in speech recognition by genetic algorithm. *Appl. Math. Inf. Sci 11*, 3 (2017), 907–913.

[64] Jin, M.; Yoo, C. D. Speaker verification and identification. In *Behavioral Biometrics for Human Identification: Intelligent Applications*. IGI Global, 2010, pp. 264–289.

[65] Joy, N. M.; Baskar, M. K.; Umesh, S. Dnns for unsupervised extraction of pseudo speaker-normalized features without explicit adaptation data. *Speech Communication 92* (2017), 64–76.

[66] Jung, J.-W.; Heo, H.-S.; Yang, I.-H.; Shim, H.-J.; Yu, H.-J. A complete end-to-end speaker verification system using deep neural networks: From raw signals to verification result. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), IEEE, pp. 5349–5353.

[67] Karnan, M.; Akila, M.; Krishnaraj, N. Biometric personal authentication using keystroke dynamics: A review. *Applied soft computing 11*, 2 (2011), 1565–1573.

[68] Keele, S., et al. Guidelines for performing systematic literature reviews in software engineering. Tech. rep., Technical report, Ver. 2.3 EBSE Technical Report. EBSE, 2007.

[69] Kitano, H. Biological robustness. *Nature Reviews Genetics 5*, 11 (2004), 826–837.

[70] Kitchenham, B.; Brereton, O. P.; Budgen, D.; Turner, M.; Bailey, J.; Linkman, S. Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology 51*, 1 (2009), 7–15.

[71] Kłaczyński, M.; Pawlik, P. Automatic detection system of aircraft noise events during acoustic climate long-term monitoring near airport. *Vibroengineering PROCEDIA 6* (2015), 352–356.

[72] Klaczynski, M.; Wszolek, T.; Cioch, W.; Wszolek, W.; Pawlik, P.; Mleczko, D.; Grzeczka, A. Identification of acoustic event of selected noise sources in a long-term environmental monitoring systems.

[73] Kolinsky, R.; Pattamadilok, C.; Morais, J. The impact of orthographic knowledge on speech processing. *Ilha do Desterro A Journal of English Language, Literatures in English and Cultural Studies*, 63 (2012), 161–186.

[74] Koolagudi, S. G.; Rao, K. S. Emotion recognition from speech: a review. *International journal of speech technology 15*, 2 (2012), 99–117.

[75] Korshunov, P.; Marcel, S.; Muckenhirn, H.; Gonçalves, A. R.; Mello, A. S.; Violato, R. V.; Simões, F. O.; Neto, M. U.; de Assis Angeloni, M.; Stuchi, J. A., et al. Overview of btas 2016 speaker anti-spoofing competition. In *2016 IEEE 8th international conference on biometrics theory, applications and systems (BTAS)* (2016), IEEE, pp. 1–6.

[76] Kotti, M.; Moschou, V.; Kotropoulos, C. Speaker segmentation and clustering. *Signal processing 88*, 5 (2008), 1091–1124.

[77] Kozhirbayev, Z.; Erol, B. A.; Sharipbay, A.; Jamshidi, M. Speaker recognition for robotic control via an iot device. In *2018 World Automation Congress (WAC)* (2018), IEEE, pp. 1–5.

[78] Kubozono, H. The mora and syllable structure in japanese: Evidence from speech errors. *Language and Speech 32*, 3 (1989), 249–278.

[79] Kuwabara, H. Acoustic properties of phonemes in continuous speech for different speaking rate. In *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96* (1996), vol. 4, IEEE, pp. 2435–2438.

[80] Le Roux, N.; Bengio, Y. Representational power of restricted boltzmann machines and deep belief networks. *Neural computation 20*, 6 (2008), 1631–1649.

[81] Lee, C.-H.; Lin, C.-H.; Juang, B.-H. A study on speaker adaptation of the parameters of continuous density hidden markov models. *IEEE Transactions on Signal Processing 39*, 4 (1991), 806–814.

[82] Lee, H.; Pham, P.; Largman, Y.; Ng, A. Y. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in neural information processing systems* (2009), pp. 1096–1104.

[83] Lee, M.; Chang, J. H. Augmenting bottleneck features of deep neural network employing motor state for speech recognition at humanoid robots. *arXiv preprint arXiv:1808.08702* (2018).

[84] Leggetter, C.; Woodland, P. Flexible speaker adaptation using maximum likelihood linear regression. In *Proc. ARPA spoken language technology workshop* (1995), vol. 9, Citeseer, pp. 110–115.

[85] Li, X.; Li, F.; Fern, X. Z.; Raich, R. Filter shaping for convolutional neural networks. In *5th Int. Conf. Learning Representations, ICLR 2017* (2017).

[86] Littlefield, H. M. The wizard of oz: Parable on populism. *American quarterly 16*, 1 (1964), 47–58.

[87] Liu, Y.; Qian, Y.; Chen, N.; Fu, T.; Zhang, Y.; Yu, K. Deep feature for text-dependent speaker verification. *Speech Communication 73* (2015), 1–13.

[88] Lleida, E.; Rodriguez-Fuentes, L. J. Speaker and language recognition and characterization: Introduction to the csl special issue, 2018.

[89] Long, Y.; Ye, H.; Ni, J. Domain compensation based on phonetically discriminative features for speaker verification. *Computer Speech & Language 41* (2017), 161–179.

[90] Lupu, C.; Lupu, V. The beginnings of using fingerprints as biometric characteristics for personal identfication purposes. *Annals of the „Constantin Brancusi" University of Targu Jiu, Engineering Series*, 3 (2014), 53–56.

[91] Maltoni, D.; Maio, D.; Jain, A. K.; Prabhakar, S. *Handbook of fingerprint recognition*. Springer Science & Business Media, 2009.

[92] Masek, L., et al. *Recognition of human iris patterns for biometric identification*. Tese de Doutorado, Citeseer, 2003.

[93] Matejka, P.; Zhang, L.; Ng, T.; Glembek, O.; Ma, J. Z.; Zhang, B.; Mallidi, S. H. Neural network bottleneck features for language identification. In *Odyssey* (2014).

[94] McCulloch, W. S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics 5*, 4 (1943), 115–133.

[95] McLaren, M.; Lei, Y.; Ferrer, L. Advances in deep neural network approaches to speaker recognition. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (2015), IEEE, pp. 4814–4818.

[96] Menditto, A.; Patriarca, M.; Magnusson, B. Understanding the meaning of accuracy, trueness and precision. *Accreditation and quality assurance 12*, 1 (2007), 45–47.

[97] Mermelstein, P. Distance measures for speech recognition, psychological and instrumental. *Pattern recognition and artificial intelligence 116* (1976), 374–388.

[98] Mitsianis, E.; Spyrou, E.; Giannakopoulos, T. Speaker verification based on extraction of deep features. In *Proceedings of the 10th Hellenic Conference on Artificial Intelligence* (2018), pp. 1–4.

[99] Morise, M.; Ozawa, K. Speaker identification framework by peripheral and central auditory models. *Acoustical Science and Technology 36*, 4 (2015), 340–343.

[100] Muda, L.; Begam, M.; Elamvazuthi, I. Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques. *arXiv preprint arXiv:1003.4083* (2010).

[101] Mun, S.; Shon, S.; Kim, W.; Ko, H. Deep neural network bottleneck features for acoustic event recognition. In *INTERSPEECH* (2016), pp. 2954–2957.

[102] Nagrani, A.; Chung, J. S.; Zisserman, A. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612* (2017).

[103] Nayana, P.; Mathew, D.; Thomas, A. Comparison of text independent speaker identification systems using gmm and i-vector methods. *Procedia computer science 115* (2017), 47–54.

[104] Neves, J. L. Pesquisa qualitativa: características, usos e possibilidades. *Caderno de pesquisas em administração, São Paulo 1*, 3 (1996), 1–5.

[105] Omid Sadjadi, S.; Pelecanos, J.; Ganapathy, S. The ibm speaker recognition system: Recent advances and error analysis. *arXiv preprint arXiv:1605.01635* (2016).

[106] O'Shea, K.; Nash, R. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458* (2015).

[107] Pang, W.; He, Q. H. A simple neural network based countermeasure for replay attack. In *Proceedings of the 2017 2nd International Conference on Communication and Information Systems* (2017), ACM, pp. 234–238.

[108] Park, G.-J. Robust design. *Analytic Methods for Design Practice* (2007), 393–442.

[109] Park, T. J.; Georgiou, P. Multimodal speaker segmentation and diarization using lexical and acoustic cues via sequence to sequence neural networks. *arXiv preprint arXiv:1805.10731* (2018).

[110] Peacocke, R. D.; Graf, D. H. An introduction to speech and speaker recognition. In *Readings in Human–Computer Interaction*. Elsevier, 1995, pp. 546–553.

[111] Petticrew, M.; Roberts, H. Systematic reviews in the social sciences: a practical guide. 2006. *Malden USA: Blackwell Publishing CrossRef Google Scholar* (2006).

[112] Plchot, O.; Burget, L.; Aronowitz, H.; Matejka, P. Audio enhancing with dnn autoencoder for speaker recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2016), IEEE, pp. 5090–5094.

[113] Pouyanfar, S.; Sadiq, S.; Yan, Y.; Tian, H.; Tao, Y.; Reyes, M. P.; Shyu, M.-L.; Chen, S.-C.; Iyengar, S. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR) 51*, 5 (2018), 1–36.

[114] Qawaqneh, Z.; Mallouh, A. A.; Barkana, B. D. Deep neural network framework and transformed mfccs for speaker's age and gender classification. *Knowledge-Based Systems 115* (2017), 5–14.

[115] Qayyum, R.; Akre, V.; Hafeez, T.; Khattak, H. A.; Nawaz, A.; Ahmed, S.; Mohindru, P.; Khan, D.; ur Rahman, K. Android based emotion detection using convolutions neural networks. In *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)* (2021), IEEE, pp. 360–365.

[116] Qian, Y.; Chen, N.; Yu, K. Deep features for automatic spoofing detection. *Speech Communication 85* (2016), 43–52.

[117] Raheli, B.; Aalami, M. T.; El-Shafie, A.; Ghorbani, M. A.; Deo, R. C. Uncertainty assessment of the multilayer perceptron (mlp) neural network model with implementation of the novel hybrid mlp-ffa method for prediction of biochemical oxygen demand and dissolved oxygen: a case study of langat river. *Environmental Earth Sciences 76*, 14 (2017), 503.

[118] Rahman, M. H.; Himawan, I.; McLaren, M.; Fookes, C.; Sridharan, S. Employing phonetic information in dnn speaker embeddings to improve speaker recognition performance. In *Proceedings of Interspeech* (2018), pp. 3593–3597.

[119] Riaz, M.; Mendes, E.; Tempero, E. A systematic review of software maintainability prediction and metrics. In *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement* (2009), IEEE Computer Society, pp. 367–377.

[120] Rusko, M.; Trnka, M.; Darjaa, S.; Ritomskỳ, M. Weaknesses of voice biometrics–speaker verification spoofing using speech synthesis. In *Proceedings of the 24th International Congress on Sound and Vibration* (2017).

[121] Saeed, U. Person identification using behavioral features from lip motion. In *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)* (2011), IEEE, pp. 155–160.

[122] Schain, M.; Schain, M. *Machine Learning Algorithms and Robustness*. Universitat Tel-Aviv, 2015.

[123] Shinoda, K. Speaker adaptation techniques for speech recognition using probabilistic models. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science) 88*, 12 (2005), 25–42.

[124] Stevens, S.; Volkmann, J.; Newman, E. The mel scale equates the magnitude of perceived differences in pitch at different frequencies. *Journal of the Acoustical Society of America 8*, 3 (1937), 185–190.

[125] Tiwari, V. Mfcc and its applications in speaker recognition. *International journal on emerging technologies 1*, 1 (2010), 19–22.

[126] Tovarek, J.; Partila, P. Speaker identification for the improvement of the security communication between law enforcement units. In *Signal Processing, Sensor/Information Fusion, and Target Recognition XXVI* (2017), vol. 10200, International Society for Optics and Photonics, p. 102001C.

[127] Tovarek, J.; Partila, P.; Rozhon, J.; Voznak, M.; Skapa, J.; Uhrin, D.; Chmelikova, Z. Optimization of multilayer neural network parameters for speaker recognition. In *Machine Intelligence and Bio-inspired Computation: Theory and Applications X* (2016), vol. 9850, International Society for Optics and Photonics, p. 98500C.

[128] Trenn, S. Multilayer perceptrons: Approximation order and necessary number of hidden units. *IEEE transactions on neural networks 19*, 5 (2008), 836–844.

[129] Verdonschot, R. G.; Kinoshita, S. Mora or more? the phonological unit of japanese word production in the stroop color naming task. *Memory & Cognition 46*, 3 (2018), 410–425.

[130] Vuddagiri, R. K.; Vydana, H. K.; Bhupathiraju, J. V.; Gangashetty, S. V.; Vuppala, A. K. Improved language identification in presence of speech coding. In *International Conference on Mining Intelligence and Knowledge Exploration* (2015), Springer, pp. 312–322.

[131] Wang, Q.; Downey, C.; Wan, L.; Mansfield, P. A.; Moreno, I. L. Speaker diarization with lstm. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), IEEE, pp. 5239–5243.

[132] Wohlin, C.; Runeson, P.; Höst, M.; Ohlsson, M. C.; Regnell, B.; Wesslén, A. *Experimentation in software engineering*. Springer Science & Business Media, 2012.

[133] Wu, S.; Zhong, S.; Liu, Y. Deep residual learning for image steganalysis. *Multimedia tools and applications 77*, 9 (2018), 10437–10453.

[134] Wu, Y.; Mao, H.; Yi, Z. Audio classification using attention-augmented convolutional neural network. *Knowledge-Based Systems 161* (2018), 90–100.

[135] Xie, W.; Nagrani, A.; Chung, J. S.; Zisserman, A. Utterance-level aggregation for speaker recognition in the wild. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2019), IEEE, pp. 5791–5795.

[136] Yella, S. H.; Stolcke, A. A comparison of neural network feature transforms for speaker diarization. In *Sixteenth Annual Conference of the International Speech Communication Association* (2015).

[137] Yu, D.; Seltzer, M. L. Improved bottleneck features using pretrained deep neural networks. In *Twelfth annual conference of the international speech communication association* (2011).

[138] Yu, H.; Tan, Z.-H.; Ma, Z.; Martin, R.; Guo, J. Spoofing detection in automatic speaker verification systems using dnn classifiers and dynamic acoustic features. *IEEE transactions on neural networks and learning systems 29*, 10 (2017), 4633–4644.

[139] Zhang, C.; Garlan, D.; Kang, E. A behavioral notion of robustness for software systems. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (2020), pp. 1–12.

[140] Zhang, D.; Wang, J.; Zhao, X.; Wang, X. A bayesian hierarchical model for comparing average f1 scores. In *2015 IEEE International Conference on Data Mining* (2015), IEEE, pp. 589–598.

[141] Zhang, Z.; Wang, L.; Kai, A.; Yamada, T.; Li, W.; Iwahashi, M. Deep neural network-based bottleneck feature and denoising autoencoder-based dereverberation for distant-talking speaker identification. *EURASIP Journal on Audio, Speech, and Music Processing 2015*, 1 (2015), 12.

[142] Zhao, X.; Wang, Y.; Wang, D. Cochannel speaker identification in anechoic and reverberant conditions. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP) 23*, 11 (2015), 1727–1736.

[143] Zhou, L.; Tao, H.; Paszke, W.; Stojanovic, V.; Yang, H. Pd-type iterative learning control for uncertain spatially interconnected systems. *Mathematics 8*, 9 (2020), 1528.

APÊNDICE A – Business Process Parameterizations

This Appendix shows the parameters used to execute the business processes presented in Chapter 4.3.

A.0.1 Business process parameterization: Creation of CNN Model for SI task

We detail the parameters used during the execution of each activity that makes up the business process called: "Creation of CNN Model for SI task". The activities are in bold and are represented in Figure 4.3 of Chapter 4.3.1.

Start (green circle): Starts the process;

Select CNN architecture: Selects CNN architecture showed in Figure 3.1;

Select Audio Dataset: Selects the version of the SIWIS audio dataset for the scenario to be analyzed;

Select Feature Extraction Method: Selects MFCC method;

Configure parameters for training: Established configuration:

1. Number of epochs: 200;
2. Number of MFCC parameters: 26 (Number of features captured in each audio signal reading.);
3. Initial learning rate: 0.002;
4. Neural network learning optimization algorithm: Adam (Method that optimizes CNN's learning calculation using the learning rate.);
5. Loss function: Categorical Cross Entropy (Method used to calculate the loss.);
6. Test percentage: 0.1 (10% of audio files in each speaker directory that is used for testing.);

7. Validation split: 0.2 (20% is the total samples that is used for validation. The remaining 80% will be used for training.);
8. Shuffle: 1 (0 - does not shuffle; 1 - shuffles the order of files, keeping the order of samples by files.);
9. Smaller file size: 1 (In seconds. Files smaller than 1 second were not used.);
10. Phoneme size: 0.15 (In seconds. Unit of time established to represent an audio phoneme and the Timestep. Based on the concepts of [79].);
11. Window overlap percentage: 0.2 (20% of the final samples from the previous window is repeated at the beginning of the posterior window.);
12. Window size: 0.15. (Width of the audio signal reading window. Same as phoneme size.);
13. Stride size: Window size * (1 - Window overlap percentage) = 0.12 (Stride - how far the window goes to read the next frame audio signal. An overlapping is being applied. As the window is 15 ms and we want 20% overlap, the stride is 12 ms. That is, each window takes 3 ms from the previous window as overlapping data.);
14. Batch size: 50 (Number of samples for CNN input.);
15. Timestep amount for each Sample: 8. (Amount established according to the reasoning detailed in Chapter 3.3.);

CNN Training and Testing for SI Task: Executes "CNN Training and Testing for SI Task" Sub-process;

Evaluate results of epochs: Evaluates the times that presented the best CNN training results;

Save CNN model: Saves the trained CNN model.;

Finish (red circle): Terminates the process.

A.0.2 Sub-process parameterization: CNN Training and Testing for SI Task

In this chapter, we detail the parameters used during the execution of each activity that makes up the sub-process called: "CNN Training and Testing for SI Task". The activities are in bold and are represented in Figure 4.4 of Chapter 4.3.1.1.

Start (green circle): Starts the process;

Read audio files from audio dataset: Reads audio files from all speakers in the audio

dataset;

Run Feature Extraction Method: Extract MFCC features from audio files;

Standardize features: Standardizes MFCC features using the following formula:

$$stdMatMFCC = \frac{matMFCC - AM}{SD}$$

where *SD* is the calculation of the Standard Deviation along the specified axis; *AM* is the calculation of the Arithmetic Mean along the specified axis; *matMFCC* is the matrix of original values for MFCC features; *stdMatMFCC* is the matrix resulting from MFCC features with standardized values;

Stores features in configurated data structure: Performed by tasks:

1. Stores MFCC features in a data structure;
2. Creates the data structure on the dimensions: Batch x Samples (50) x Timestep (8) x Features (26), labeling the speakers' samples/timesteps and segmenting them to be used for training, validation and testing;
3. Shuffles audio file samples in the data structure: shuffles the order of files keeping the order of samples in each file;

Create speraker label structure: Performed by tasks:

1. Creates the speakers labels structure;
2. Formats the speakers labels structure with numerical and sequential values replacing speaker names;
3. Segments the speaker label structure for training, validation and testing;

Segment training and validation data: Segments the data structure containing the speaker audio features into two parts: for training and for validation;

Estabilish the metrics: Metrics calculated in Training and Validation are: Accuracy, Precision, Recall and F1;

Perform training and validation: Performed by tasks:

1. Creates files for execution: log files, trained model, model weights, model results and neural network parameter records;
2. Performs CNN training using speakers' data structures and labels for each 200 epochs;
3. Performs CNN validation using speakers' data structures and labels at the end of each epoch;
4. Saves metric values for each epoch, during training and validation;
5. Saves the CNN model in H5 format if the Accuracy Validation value is higher;

Perform testing on the trained model: Performed by tasks:

1. Selects audio files on test directory. Audio files were previously separated;
2. Performs tests for the CNN model and saves the test results;

Save results: Saves the best CNN model results. The model considered best is the one with the best Validation Accuracy. Files containing the training, validation, and testing values for this model are saved separately.

Finish (red circle): Terminates the process.

A.0.3 Business process parameterization: CNN performing SI Task

In this chapter, we detail the parameters used during the execution of each activity that makes up the business process called: "CNN performing SI Task". The activities are in bold and are represented in Figure 4.5 of Chapter 4.3.2.

Start (green circle): Starts the process;

Read audio file to requesting speaker: Reads audio files for a given speaker;

Run Feature Extraction Method: Extract MFCC features from audio files;

Standardize features: Standardizes MFCC features using the same formula used on training process:

$$stdMatMFCC = \frac{matMFCC - AM}{SD}$$

;

Store features in configurated data structure: Stores MFCC features in a data structure with de following dimensions: Samples (dependent on audio file size) x Timestep (8) x Features (26);

Run CNN modeled: Performed by tasks:

1. Loads the CNN model;
2. Inputs in CNN model the data structure containing speaker features;
3. Runs the CNN model;

Run Prediction Methods: Performed by tasks:

1. Runs the Class Prediction calculation;
2. Shows in an orderly way the first 5 speakers pointed by the Class Prediction;
3. Runs the Probabilistic Prediction calculation;

4. Shows in an orderly way the first 5 speakers pointed by the Probabilistic Prediction;

Verify if the prediction result is conclusive: Compares the results of Class Prediction and Probabilistic Prediction with the Acceptance Threshold, set at 65%. If the class indicated by both Predictions is the same and the prediction percentage is greater than 65% in both Predictions, this class is presented as the result of the SI. Otherwise it informs that the result is inconclusive;

Present the speaker identification result: Prints the SI result.

Finish (red circle): Terminate the process;

APÊNDICE B – CNN Architecture's Sequential Layer Chaining

This Appendix present a vision of CNN architecture's sequential layer chaining. This is a complementary view to Figure 3.2, where the CNN architecture summary is shown.

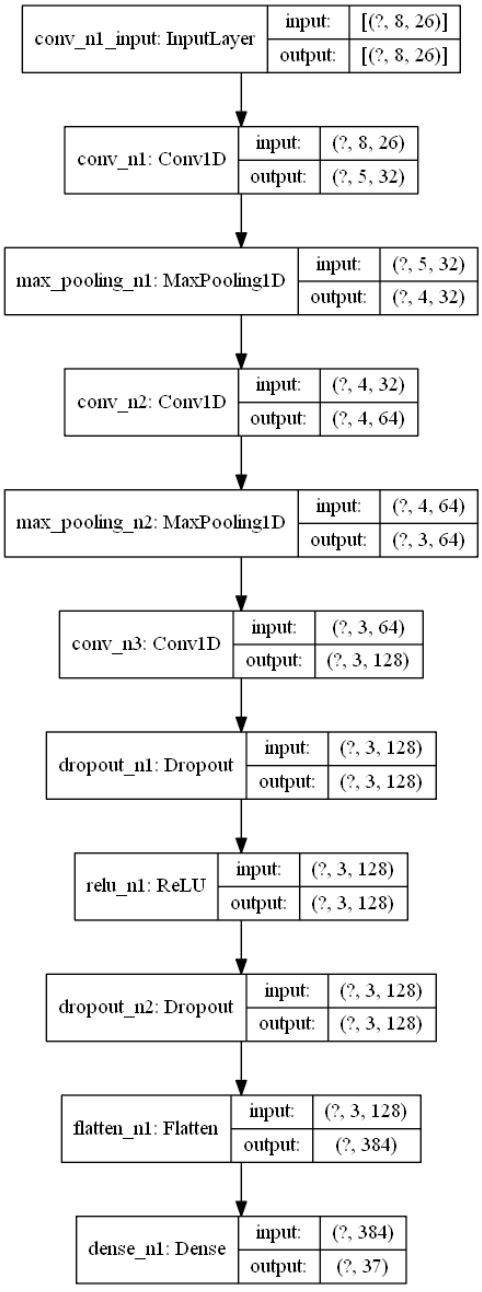


Figura B.1: CNN architecture showing sequential layer chaining and the input and output parameters of each layer.

APÊNDICE C – Additional knowledge

In this Appendix we present brief reports on the history of biometric identification and ANN.

C.1 A Brief About Biometric Identification

Biometrics is a term derived from the Greek words "bios" (life) and "metrikos" (measure) and stands for a personal identification that uses measurable characteristics of a person [90]. It refers to identifying an individual based on his or her distinguishing physiological and/or behavioral characteristics (biometric identifiers) [60]. Historical records point to the use of biometrics 31,000 years ago, when fingerprints were used by prehistoric men as a signature. In 500 B.C. Babylonian business transactions were done on the basis of fingerprints on clay tablets as a means of security. In 14th Century Chinese used fingerprints for business transactions and also to differentiate their children [11]. More recently, around the mid-nineteenth century, the practice of using biometric methods began to become more widespread for identifying people and for creating registers of biometric identifiers. The first record of finger and hand prints which was recorded uniformly was in 1858 by Sir William Herschel who was in Civil Services, India and wanted to make a record of employees to distinguish them [11]. But other ideas emerged that relied on different ways to identify a person. Some were discarded as they proved to be ineffective over time. In 1879 Alphonse Bertillon, a police officer in Paris, introduced the anthropometric registering method. But the anthropometry didn't last too long, really quickly revealing its deficiencies and error possibilities. The main imperfections of this method, we can mention: the instability of the human body's parameters (it wasn't applicable to children and teenagers); subjectivism in measuring the important parts of the human body (police officers didn't place the measurement instrument in the same point all over the time); decalcifications, caused by the aging, diseases or trauma, producing the modification of the dimensions of human skeleton [90]. In 1936 the concept of iris pattern to recognize

humans was proposed by an ophthalmologist Frank Burch [11]. Many researches were carried out using different physiological and behavioral characteristics of the human body and were used as biometric identifiers to obtain a method that presented precision. Some biometric identifiers already used are: fingerprint, DNA, face recognition, iris recognition, hand, signature, voice, gait, keystroke [11]. Given several examples of people who have dedicated themselves to research contributing to the evolution of biometric methods, the scientific literature considers that Francis Galton played a noteworthy part in the systematic introduction of quantitative methods to investigate biological phenomena. Because of his considerable contributions, Galton is designated as the "pioneer of heredity and biometry" [6]. With the advent of technology and its evolution, biometric systems were developed where several previously researched methods were codified. A biometric system provides automatic identification of an individual based on a unique feature or characteristic possessed by the individual [92]. It is essentially a pattern recognition system that makes a personal identification by establishing the authenticity of a specific physiological or behavioral characteristic possessed by the user [60]. The proof of the effectiveness of biometric systems had as a return the popular credibility in their use. As a result, tools and devices were created that facilitated its use. Biometric systems then began to be used in several areas. Some areas of use of biometrics are: Forensic, with application in criminal investigation, corpse identification, parenthood determination; Civilian, with application in national ID, driver's license, welfare disbursement, border crossing; Commercial, with application in Automated Teller Machine (ATM), credit card, cellular phone, access control [60]. Biometrics presents fundamental concepts that need to be taken into account when designing a biometric system. An ideal biometric should be universal, where each person possesses the characteristic; unique, where no two persons should share the characteristic; permanent, where the characteristic should neither change nor be alterable; and collectable, where the characteristic is readily presentable to a sensor and is easily quantifiable [60]. The relationship between biometrics and security systems became very strong with the advancement of technology. In general, large investments in security areas have become a necessity for companies and corporations. Hacker attacks on corporate systems, especially banking systems, have intensified more and more and have become a daily risk not only for institutions but also for their customers. The provision of ever more effective security tools has become a market demand made by customers of technological services and products. Updating security systems is a frequent need for corporations but also for citizens, for the protection of their personal data. Use of biometrics in security systems is a reality. During a security project the designer of a practical biometric system must

consider a number of issues, including: Performance, that is, a system's accuracy, speed, robustness, as well as its resource requirements, and operational or environmental factors that affect its accuracy and speed; Acceptability, or the extent people are willing to accept for this particular biometric identifier in their daily lives; Circumvention, as in how easy it is to fool the system through fraudulent methods [60]. According to [31], the acceptance of a biometric identification system depends, on the one hand, on its operational, technical and manufacturing characteristics and, on the other, on the final application and its financial possibilities. Thus, it should minimally consider the following characteristics to attest to its feasibility: reliability, ease of use, user acceptance, ease of implementation and cost.

An automatic biometric pattern recognition system can establish a person's authenticity through their specific physiological or behavioral characteristics [31]. Some examples of behavioral biometrics are: keystroke dynamics, signature recognition, SR, voice recognition, gait recognition and lip motion [121], [67], [32], [13]; and examples of physiological biometrics are: fingerprint, face recognition, iris recognition, hand geometry, retina geometry, palmprint, hand vein geometry, dna, thermal imaging, ear shape, body odor, fingernail bed (dermal structure under the fingernail) [13]. With regard to the different biometric technologies in use, fingerprint continues to be the leading technology in terms of market share. It is probably the best-known biometric technology, and currently employed in a number of real-world applications. Face recognition is also a popular biometric technique in some countries, as it seems to be the most user-friendly, although it does not reach a high degree of accuracy. Iris recognition, although not being quite popular, is probably the most accurate biometric technology developed so far, according to [31]. In 2003 [31] also showed the results of a survey on biometric market share by technology, from 2002, where the use of fingerprint appeared as the most used method, corresponding to 42.5% of the solutions, followed by face recognition (12.6%), hand geometry (12%), iris recognition (11.3%), biometric middleware (9.5%), signature recognition (6%) and voice recognition (5.3%). In 2009, a new survey presented by [91] showed that fingerprint was still among the most used methods and presented the following result: Automated Fingerprint Identification System (AFIS)/Live scan (38.3%), fingerprint (28.4%), face recognition (11.4%), biometric middleware (8%), iris recognition (5.1%) and voice recognition (3%). In [3], a more current survey, from 2013, brings the following results with major highlights: AFIS/Live scan (34%), fingerprint (25%), face recognition (13%), iris recognition (5%), hand geometry (5%), biometric middleware (5%), voice recognition (3%), vein recognition (3%) and multiple traits (3%). In results, from 2002 to 2013, fin-

gerprint remained ahead as the most used method. Currently, fingerprint is still widely used and incorporated as a biometric identification method in personal devices such as smartphones.

C.2 A Brief About ANN

The first step taken towards ANN took place in 1943 when neurophysiologist Warren McCulloch and mathematician Walter Pitts described how neurons in the brain might work and modeled a simple neural network using electrical circuits [94]. Since then, much research has contributed to the evolution of ANN and its use in machine learning solutions. In recent years, machine learning has become more and more popular in research and has been incorporated in a large number of applications, including multimedia concept retrieval, image classification, video recommendation, social network analysis, text mining, and so forth [113]. According to [106], ANNs are computational processing systems heavily inspired by biological nervous systems (such as the human brain) operation. ANNs are mainly comprised by a high number of interconnected computational nodes (referred to as neurons), working entwine in a distributed fashion to collectively learn from the input in order to optimise its final output. An ANN having many multiple hidden layers, stacked upon each-other, is commonly called DNN [106]. DNN architectures are characterized by one or more hidden layers consisting of hidden nodes, with each hidden node representing a nonlinear activation function [41]. In recent years, the use of DNN became a hot research topic in machine learning, also achieving a breakthrough in speech recognition [87] and in other speech processing activities. DNNs have presented so far an excellent ability to automatically learn feature representations from high-dimensional input data, as a result of their outstanding performance in many areas [134]. One of the most popular DNNs is the Convolutional Neural Network (CNN). It take this name from mathematical linear operation between matrixes called convolution [115]. CNN has an excellent performance in machine learning problems [5]. It has been popular in pattern recognition for non-relational data, such as images and sound processing [134].