RAFAEL GERALDO DOS SANTOS

PT-PT SYNTHETIC SPEECH DETECTION

UNIVERSIDADE DO ALGARVE
Faculdade de Ciências e Tecnologia
2023

RAFAEL GERALDO DOS SANTOS

PT-PT SYNTHETIC SPEECH DETECTION

Master in Informatics Engineering

Work done under the supervision of: José Valente de Oliveira
Joana Coutinho Sousa

UNIVERSIDADE DO ALGARVE
Faculdade de Ciências e Tecnologia
2023

PT-PT SYNTHETIC SPEECH DETECTION

Declaração de autoria de trabalho

Declaro ser o autor deste trabalho, que é original e inédito. Autores e trabalhos consultados estão devidamente citados no texto e constam da listagem de referências incluída.

Declaration of authorship of the work

I hereby declare to be the author of this work, which is original and unpublished. Authors and works consulted are properly cited in the text and included in the reference list.

(Rafael Geraldo dos Santos)

Acknowledgements

I would like to express my gratitude, especially to my supervisors, teacher José Valente de Oliveira and Joana Coutinho Sousa, for their constant support, guidance, and help during the elaboration of this project. My sincere thanks also go to my professors in both bachelors and masters degrees in informatics engineering, for sharing their knowledge and for all the incentives they gave me to pursue this degree.

I want to further extend my appreciation to the entire team of NOS Inovação for sharing their skills and for always being available to lend a hand, as well as to NOS for providing the necessary resources and environments for the completion of this work.

Last but not least, i must express my profound gratitude to my family because this journey without them wouldn't be possible. Their encouragement and support have been invaluable to me. Additionally, I want to thank my friends, with whom i shared great laughs and moments both inside and outside of the University of Algarve. They were very important for my personal and professional growth and i will forever be thankful.

Abstract

Recent developments in the field of artificial intelligence (AI) have led to the creation of powerful generative models. These models have demonstrated such capabilities that it becomes nearly impossible for a human to distinguish between generated and human utterances, between synthetic and natural speech. A relatively recent example of this fact is the deepfake video of former U.S. President Barack Obama [1]. This video not only serves as a demonstration of the capabilities of AI models but also highlights the potential for misinformation, as these models can deceive individuals into believing in fabricated scenarios. This extends to the realm of synthetic speech, where models like Google Duplex [2], leveraging WaveNet technology, a deep neural network for seamless speech creation, exhibit an impressive degree of realism and naturalness. For this reason, two situations may arise. The first is related to new business opportunities, such as the creation of realistic voiceovers for films and animations or enhancement in the communication for individuals with hearing or speech impairments [3]. The other, raises concerns about privacy and security since voice impersonation is easily achievable with today's tools.

Given this fact, an analysis of approaches applied in the ASVspoof challenge [4] was carried on. The ultimate goal is to develop a system capable of distinguishing between real voices and cloned voices, by adapting the research done on this challenge to the portuguese from Portugal (PT-PT) language. For this purpose, we first created a PT-PT dataset using both text-to-speech (TTS) and speech-to-speech (STS).

Then, we employed and implemented some models from the literature and tested in several datasets that encompass both english and PT-PT voices, to evaluate their performance and reach conclusions. From this, we found out that while this is a difficult task, by augmenting the data with different impulse response devices (IRs) and compressions codecs, there was an improvement in the generalization to different attacks from different datasets.

Overall, after the evaluation process the best models found through statistical analysis were the ResNet-OC and ECAPA-TDNN. Being our goal tailored to PT-PT, by fine-tuning them, we further improved their performance.

At the end future steps are highlighted, one of which may be very important to complement the work made so far, which is the integration of the fraud detection component.

Keywords: Deep learning, speech detection, PT-PT dataset, asvspoof challenge

Resumo

Os recentes desenvolvimentos na área de inteligência artificial (IA) levaram à criação de poderosos modelos generativos. Estes modelos demonstraram tais capacidades que torna quase impossível para um humano a tarefa de distinguir entre afirmações geradas na IA e humanas. Um exemplo relativamente recente deste facto é o do vídeo falso do ex-Presidente dos Estados Unidos, Barack Obama [1]. Este vídeo não serve apenas como uma demonstração das capacidades dos modelos de IA, mas também destaca o potencial de desinformação, uma vez que estes modelos podem induzir indivíduos a acreditar em cenários "fabricados". Isto estende-se ao domínio da fala sintética, onde modelos como o Google Duplex [2], que aproveitando a tecnologia WaveNet, uma rede neuronal profunda que serve para a criação perfeita de um discurso, exibem um elevado grau de realismo e naturalidade. Por este motivo, podemos-nos deparar com duas situações: a primeira está ligada à oportunidade para novos negócios, como a criação de dobragens realistas para filmes e animações [3], enquanto que a outra, abre caminho a problemas relacionados com a privacidade e segurança, uma vez que a falsificação da voz é de fácil execução com as ferramentas que existem hoje em dia.

Dado este fato, foi realizada uma análise das abordagens aplicadas no desafio ASVspoof [4]. O objetivo final é desenvolver um sistema capaz de distinguir entre vozes reais e vozes clonadas, adaptando a investigação realizada no mesmo, para a língua portuguesa de Portugal (PT-PT). Para este fim, primeiro criámos um conjunto de dados PT-PT usando tanto a conversão de texto-para-fala como a conversão de fala-para-fala.

Em seguida, utilizámos e implementámos alguns modelos da literatura e testámos em vários conjuntos de dados, que englobam tanto vozes em inglês como em PT-PT, para avaliar o seu desempenho e tirar conclusões. Desta forma, descobrimos que, embora esta seja uma tarefa difícil, ao aumentar os dados com diferentes dispositivos de resposta ao impulso e tipos de compressões, houve uma melhoria na generalização para diferentes ataques de diferentes conjuntos de dados.

No geral, após a fase de testes, os melhores modelos encontrados através da análise estatística foram o ResNet-OC e o ECAPA-TDNN. Sendo o nosso objetivo focado para PT-PT, ao refiná-los, conseguimos melhorar ainda mais o desempenho dos dois.

No final, são destacados os passos futuros, um dos quais pode vir a ser muito importante para complementar o trabalho feito até ao momento que é a incorporação da componente de deteção de fraude.

Palavras Chave: aprendizagem profunda, detecção de fala, conjunto de dados PT-PT, desafio asvspoof

Contents

List of Figures

List of Tables

List of Acronyms

AI	Artificial Intelligence
PT-PT	Portuguese from Portugal
ASV	Automatic Speech Verification
TTS	Text-to-Speech
SS	Speech Synthesis
STS	Speech-to-Speech
VC	Voice Conversion
DNN	Deep Neural Network
LFCC	Linear Frequency Cepstral Coefficient
MFCC	Mel-Frequency Cepstral Coefficient
CMVN	Cepstral Mean and Variance
CQCC	Constant Q Cepstral Coefficient
GMM	Gaussian Mixture Model
MLP	Multilayer Perceptron
CNN	Convolutional Neural Network
FFT	Fast Fourier Transform
CQT	Constant Q Transform
SE	Squeeze-and-Excitation
LA	Logical Access subset
DF	Deepfake subset

PA	Physical Access subset
FoR	Fake or Real dataset
MLS	Multilingual Librispeech dataset
RVC	Retrieval based Voice Conversion WebUI
IRs	Impulse Responses
OC-Softmax	One-Class Softmax
ECAPA-TDNN	Emphasized Channel Attention, Propagation and Aggregation Time Delay Neural Networks
FC	Fully Connected
TE	Transformer Encoder
BLS	Broad Learning System
CE	Cross-Entropy
EER	Equal Error Rate
TN	True Negative
FP	False Positive
TP	True Positive
FN	False Negative
FAR	False Acceptance Rate
FRR	False Rejection Rate

1

Introduction

1.1 Motivation

As technology advances, numerous schemes emerge, posing challenges to security and reliability in communication channels. One reason is that synthetic voices can be used to attempt to deceive individuals, organizations, and even automated systems. Therefore, it is necessary to address various issues in this area, including voice spoofing and impersonation, phishing and vishing attacks, social engineering, call center fraud, voice authentication vulnerabilities, and the threat of deepfakes. Deepfakes, in particular, represent a serious challenge as they consist of highly convincing audio clips that are difficult to distinguish from genuine recordings. To tackle these situations, a multifaceted approach is required, involving technological advancements, robust security

measures, user awareness, and education, as well as collaboration between technology providers, law enforcement agencies, and regulatory authorities.

In this context, the development of a system to counter these attacks will be discussed in Section 1.2. Section 1.3 explains key challenges and contributions, while section 1.4 outlines upcoming chapters.

1.2 Objectives of the work

In recent times, synthetic voices have been exploited for fraud schemes, and with the evolution of strategies employed by malicious actors, there is a need to develop robust countermeasures, that extends from traditional legal frameworks to technological solutions such as fraud detection systems. Therefore, the focus of this work is on those systems, because currently not only they are hard to find but they don't encompass the portuguese from Portugal (PT-PT) language, making this work innovative and crucial for the safety of the users. Due to this fact, we directed our focus to the ASVspoof challenges — automatic speaker verification spoofing challenges designed to assess and push the boundaries of anti-spoofing technologies. This choice was motivated by two reasons. First of all, because it fosters research in anti-spoofing by providing a platform for evaluating and advancing the techniques used to detect spoofed or synthetic speech. Then, because it has one of the most extensive and diverse databases for the task of speech detection, which is fundamental if we want to ensure that our model classifies correctly across different attacks and conditions. [14]

From the current state of research on this challenge, we notice the preference of adopting deep learning models instead of traditional statistical models like hidden markov model (HMM) and gaussian mixture model (GMM), to build a speech detection system. The reason for this has to do with the advancements in the deep learning field which led to a superior performance in capturing intricate patterns and representations from complex data [15, 16]. However, while these changes are happening there still is a gap related to the generalization to unseen attacks during the training pro-

cess. This problem is influenced by factors like varying attack difficulty and recording conditions. Despite efforts in feature engineering, model design, and other aspects, further advancements are needed to address the existing challenges and enhance the robustness of synthetic speech detection systems [8, 9].

Another concern to this study is related to the scarcity of PT-PT language datasets, while there are multiple sources of english audio datasets such as LibriSpeech ASR corpus, options for PT-PT language are limited. This scarcity poses a significant challenge as it directly affects the acquisition of suitable and diverse data necessary for analyzing the possibility of creating a language-agnostic model.

Lastly, while developing a synthetic speech detection system for the PT-PT language is half way to achieving our complete system, there are numerous variables to consider and explore particularly in the context of fraud detection. In this regard, specific keywords and phrases, can serve as important indicators for detecting the likelihood of identifying potential threats. This last component is left to be explored in depth on the future.

Summing up, this internship tries to address this research gap and make a substantial contribution to the existing literature by examining diverse solutions to tackle these issues and advance research in the field.

1.3 Problem and Contributions

The main challenge that this internship will try to solve is to distinguish between synthetic and human voices, by employing models that are allegedly able to identify speech patterns in both PT-PT dataset and English-based datasets.

From this, major contributions are: Development of a model capable of discerning between synthetic and human voices. Supported by robust performance evidence especially found in [10], we explore its capabilities on some datasets comparing it to other models, and at the same time enhance its performance through data augmentation techniques and selection of optimal parameters; Creation of a new dataset to

address challenges related to the limited availability of existing datasets in the PT-PT language for SSD, containing synthetic and human voices.

1.4 Organization of the thesis

This report is structured into additional four chapters. Chapter 2 reviews the literature, focusing on state-of-the-art systems and techniques, such as data augmentation. Chapter 3 outlines the methodology used to achieve our goal. In Chapter 4, a comparison with other models is conducted to assess our approach. Finally, Chapter 5, presents the main conclusions, highlights future steps, and outlines the overall thesis plan.

2

State of the art, including materials and methods

Throughout recent times, events held by tech giants such as Google and Microsoft have captured the attention of the AI community. The projects presented, namely Google Duplex [2] and Microsoft Xiaoice [17], demonstrated significant advances towards reproducing human conversations with machines so realistically that it becomes challenging for listeners to distinguish between artificial and natural speech, ending up being a cause for concern, since intruders can impersonate other people, reproducing their voices perfectly [18]. This poses a potential threat to various domains, including the spread of misinformation [1, 15], telecommunications issues [19, 20, 21], and the reliability of Automatic Speech Verification (ASV) systems [22]. The authorities

have already recognized this risk and began to propose regulations, e.g., in California, the so-called "Bot bill", SB-1001 [15, 23], aiming to prohibit the use of bots in online communications without disclosure of their artificial nature. However this alone cannot prevent fraudulent actions, therefore the need for urgent solutions. So, to tackle this problem, we direct our attention to the many works done around the detection of false speech to protect ASV systems [24]. These works involve solutions that combine front-end feature extractor with back-end binary classifier, which turns out to be the standard framework for the detection of synthetic speech. Within this framework, a majority of the existing works have focused on the development of hand-crafted features, resulting in a wide variety of choices, including but not limited to:

- Cepstral Mean and Variance (CMVN), which normalizes utteraces by forcing them to have zero mean and unit variance [25];
- Mel-Frequency Cepstral Coefficient (MFCC) that captures the vocal tract features (cepstral coefficients) by leveraging insights from the non-linear scale of the human ear [26, 27];
- Constant Q Cepstral Coefficient (CQCC), which extracts cepstral coefficients, similar to MFCC, but it does so using the Constant Q Transform (CQT) [10, 28].

This extraction of hand-crafted features involves the process of transforming raw data into numerical features. This transformation retains essential information from the original dataset, and it is widely preferred over applying machine learning directly to raw data due to its tendency to yield superior results. [29].

After this extraction step, three scenarios are then possible for classification tasks. The first one that implies the training of a multilayer perceptron (MLP) and convolutional neural network (CNN), replacing models such as gaussian mixture model (GMM) and support vector machine (SVM); the second one, which uses the deep neural networks (DNNs) at the front-end followed by conventional back-end classifiers; the third one, which uses DNNs for both back-end and front-end with pre-transformed features as input, such as, CQT [5, 30].

Despite the naturalness of this process, people question about the necessity of applying pre-transforms and hand-crafted features, given the fact that deep neural networks excel in feature extraction and that by applying these transformations usually some information gets discarded. This led to the emergence of an alternative scenario: end-to-end DNNs, which aims to outperform state-of-the-art methods in the ASVspoof2019 challenge without the necessity of such preprocessing steps [5].

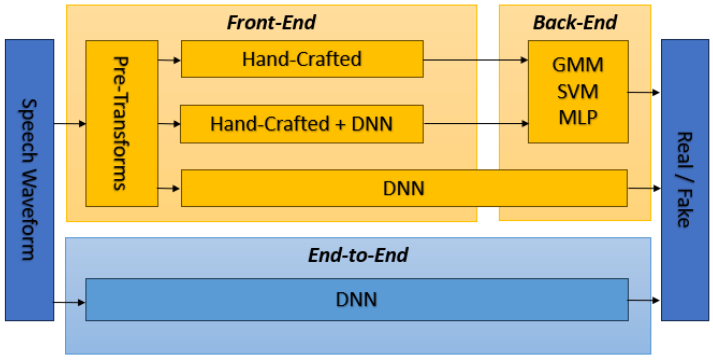

Figure 2.1: Speech detection scenarios (Based on [5])

Therefore, through the analysis of works that portray these scenarios (Figure 2.1), we aim to understand the techniques and strategies currently being employed. More specifically, we focus on works such as: Li, Xu, et al. [16] who introduced Res2Net and the SE (Squeeze-and-Excitation) block, which achieved state-of-the-art performance in the ASVspoof2019 dataset; Hua et al. [5], where two types of DNN were applied, one of which surpassed the performance of some state-of-the-art systems; Zhang, You, et al. [8], who introduces the concept of channel robustness to improve the performance of the model as well as that of Chen, Xinhui, et al. [9], which further deepens this idea and ends up being the work that succeeded the previous one; Zhang, Zhenyu, Xiaowei Yi, and Xianfeng Zhao. [10] who uses together the Transformer Encoder model and ResNet and finally Lieto, Alessandro, et al. [15], who ends up introducing this problem of classification between real and false speech, as well as the application of a basic architecture of a CNN.

So, for the next sections, we first conducted an overview of speech generation techniques in view of creating the new SSD PT-PT dataset. Then, based on the findings presented in these studies, we identified the most promising SSD models for our re-

search, considering several factors such as datasets, data augmentation techniques, feature extraction methods, model architectures, training strategies, evaluation metrics, and we also made a brief introduction to the fusion strategy implemented in some papers [9, 10, 16] that is used to enhance system's performance, which is a valuable approach in ASVspoof where achieving the best possible score is crucial [5], despite not using it.

For future endeavors, we also started to analyze some works for the domain of fraud detection, which is explained in Section 2.8. Namely Rui Zhong, et al. [11], which used an incremental system to detect fraud based on the first 15 seconds of audio and Guojun Chu, et al. [31], that implemented a fraud detection model based on the intertwined spatial-temporal patterns of user behaviors, tailored for the communications industry. Section 2.8, details the architecture that we found more promising.

2.1 Speech generation overview

Before jumping into the dataset part, its important to emphasize the types of spoofing techniques/attacks that the models will aim to detect as synthetic. These are mainly composed by two of them: text-to-speech and speech-to-speech (voice conversion), due to their accessibility and effectiveness according to [32]. Here, accessibility refers to the ease with which the attack may be performed, i.e., whether the technology is widely known and available or whether it is limited to the technically knowledgeable. Whereas effectiveness, reflects the increase in FAR (false acceptance rate) caused by each attack or the risk it poses to systems such as ASV [32].

Building on this understanding, going over the fundamentals of voice generation technologies is imperative. Unlike impersonation, which is not accessible to everyone and primarily relies on mimicking a target speaker's voice timber and prosody without the aid of computer-assisted technologies [6], voice generation methods possess a medium to high level of accessibility and demonstrate high effectiveness [6, 32]. Moreover, both text-to-speech and speech-to-speech techniques are incorporated in all

spoofing datasets so exploring its components might be important for this study.

2.1.1 Text-To-Speech

Text-to-speech (TTS) also known as speech synthesis (SS), is a technique for generating intelligible and natural-sounding artificial speech given any arbitrary text using machine learning based models [33, 34]. TTS services have three stages to address the problem of mapping a sequence of characters to a sequence of continuous numbers (audio) as shown in Figure 2.2.

Figure 2.2: The three stage pipeline of TTS (Reproduced from [6])

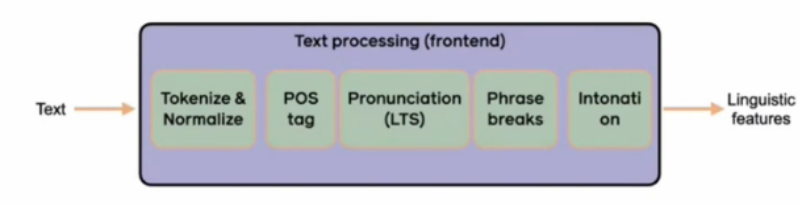

Figure 2.3: Text processing (Reproduced from [6])

In the first phase, we have the text processing, where the input text is converted into linguistic features. This process involves dividing a given sentence into a sequence of text tokens. Any non-standard text, such as "5", is then rewritten as its corresponding linguistic expression "five" (normalization). Next, the part-of-speech (POS) of each word is extracted by considering both its definition and contextual meaning. This process consists in essentially assigning to the word the correct tag. Then, the task is to predict the pronunciation of the letters, such as e.g transforming "world" into "werld" as illustrated in Figure 2.4. Lastly, it comes down to predicting the prosodic features which consists of defining the rhythm, intonation, and timing patterns of speech [6, 35].

Figure 2.4: Text processing example (Based on [6])

Then, comes the regression part. The main goal here is to convert these linguistic features into an acoustic feature such as a spectogram, using models such as CART(+HMM), feed forward network (FFN), recurrent network and more recently encoder-decoder [6]. One of the most known examples is the end-to-end Tacotron model [36]. This model uses sequence-to-sequence regression, where it turns the linguistic features into numerical values and then uses the decoder with attention mechanisms to predict the corresponding acoustic features [6].

For the waveform generator, the task is to reconstruct the audio, more specifically, synthesize waveforms from the predicted spectrogram [36]. Regarding the development of this kind of generators, there is a long history ranging from waveform concatenation to more recently neural vocoders, such as the wavenet and its variations [6, 33]. This marked a significant shift in the paradigm, as the outcomes of the TTS systems developed in Blizzard Challenge 2019 [37], resulted in recordings closely resembling natural ones.

2.1.2 Speech-To-Speech

Speech-to-speech (STS), commonly known as voice conversion (VC), aims to change the timbre and prosody of a given speaker's speech to that of another speaker, while the content of the speech remains the same [34]. Contrary to TTS, the input of a VC system normally is a natural utterance of the given speaker. This technology has a similar process to TTS which consists in three modules (Figure 2.5).

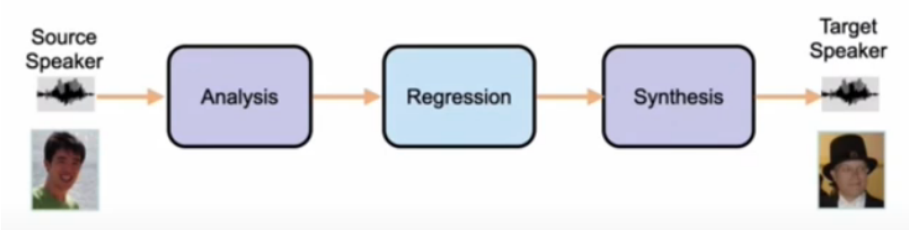

Figure 2.5: The three stage pipeline of VC (Reproduced from [6])

In the analysis step, typically the signal is not fed directly to the system, but rather represented in a more compact way, by applying feature extraction methods. This aims to capture prosodic features such as the fundamental frequency, intonation and duration, and spectral characteristics which relates to voice timbre, in order to identify and manipulate the distinctive elements of a speaker's voice [6, 33, 34].

Regarding the regression, before the encoder-decoder architectures, frame-level processing was commonly employed. This involved analyzing speech signals in short time segments, typically lasting around 20 milliseconds. The duration of each frame was selected to maintain the assumption of signal stationary, ensuring that statistical parameters remained relatively fixed within each frame [38]. Some examples of architectures performing this kind of processing are: GMM, DKPLS and recurrent networks such as LSTM [6, 7]. Unlike these architectures, which are classified as parallel voice conversion methods because they require parallel data — training data from both speakers that share the same linguistic content — there are also non-parallel methods that address this limitation. These methods enable the development of systems capable of handling source and target speakers who speak different languages or have distinct accents [39]. One such example is the CycleGAN-based VC [39]. Originally developed for unpaired image-to-image translation [40], CycleGANs aim to uncover relationships between distributions. They employ a cycle-consistency loss to ensure that input information remains invariant as it passes through the network, while an adversarial loss makes the generated data indistinguishable from real target data [39]. This approach enables learning from unpaired data without the need for directly matching features [39]. Also and similarly to TTS, the encoder-decoder has proved valuable for this domain. In this sequence-to-sequence mapping, instead of characters it encodes a se-

quence of speaker representations to then decode it to match that of the target speaker. [6] (Figure 2.6). An example of this approach is AttS2S-VC [6, 41]. Moreover, GANs have also been employed to sequence-to-sequence VC reaching great results [40, 42].

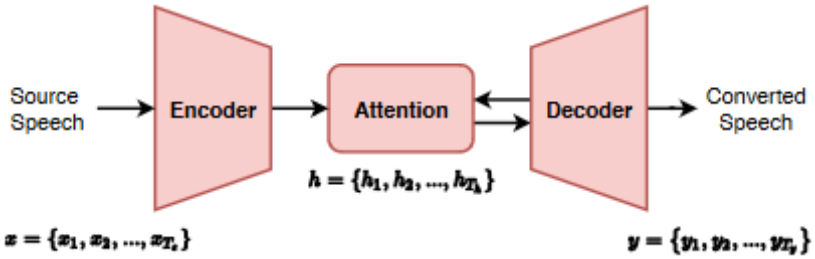

Figure 2.6: Encoder-decoder for VC (Reproduced from [7])

For the synthesis part, it's again very similar to TTS. The goal is to reconstruct the waveform from the mapped features in the regression. For this, we have well-known neural vocoder and signal processing vocoder, such as STRAIGHT, Griffin-Lim and others [6]. A good source for tracking advancements in quality and similarity within this technology is the Voice Conversion Challenge (VCC) [43].

2.2 Datasets overview

In this section, we start by introducing the datasets used for validating the effectiveness of these speech detection approaches. In Table 2.1, it can be found an overview of the ones employed by the researchers that might contribute to our work.

Table 2.1: Authors' used speech detection datasets

Datasets	**Description**
ASVspoof2015 dataset [33]	The first ever asvspoof dataset developed with a particular focus on spoofing detection. Contrary to the upcoming ones, the spoofing samples have been generated by simple and, in some cases, outdated VC and TTS algorithms.

Continued on next page

Table 2.1 – *Continued from previous page*

Datasets	**Description**
ASVspoof2019 dataset [44]	Comprises a diverse collection of bona fide and synthetic speech. It features a wide range of attacks generated by text-to-speech, such as Merlin TTS, and voice conversion algorithms for the logical access (LA) subset. Also it has a physical access (PA) subset which contains samples of real bona fide and, captured and replayed speech signals.
ASVspoof2021 dataset [14]	Different from previous ASVspoof challenges, the LA task in 2021 presents codec and transmission channel variability, while the speech deepfake (DF) task presents general audio compression without speaker verification, so for this last one, the goal is not to confirm the identity of the speaker but to detect whether the provided audio is human or artificially generated. Regarding the PA task, contrary to ASVspoof2019 dataset, it has recordings made in real physical environments.
FoR-norm dataset [45]	Comprises human speech samples from datasets such as the Arctic dataset, and spoofed samples generated by commercially available TTS options such as Google Cloud TTS. Furthermore, this "norm" version comprises a balanced version of the FoR set, meaning that it ensures equal distribution between genders (male and female) and classes (fake and real).
Custom dataset [15]	Composed by human speech samples from datasets available online namely LibriSpeech ASR corpus and CMU ARCTIC, and bot speech, generated using as input the same sentences found in the transcripts of the human dataset to feed TTS services such as Google TTS and Amazon Polly.

Although some of these datasets are useful for comparing the performance of different models, there is a significant drawback: they only contain English audio data.

As a result, in order to continue our research, we found necessary to create a dataset in the PT-PT language, as there is currently none available in the market. To accomplish this, we followed a similar approach as in [15], which is described in the next section.

2.3 Dataset creation

For the creation of our synthetic speech dataset, we looked for both PT-PT datasets as well as the latest speech synthesis methodologies. First, the datasets in Table 2.2, were selected and incorporated into the genuine dataset part. After this selection, we obtained only the Portuguese audios portion from each dataset, except for NOS Sotaques, since it exclusively consists of Portuguese audios.

Table 2.2: Genuine speech datasets

Dataset	**Description**
Voxforge [46]	This open speech dataset has a large collection of audio recordings and corresponding transcriptions in several languages, and is used primarily for speech recognition.
Mozilla Common Voice [47]	This corpus shares similarities with Voxforge in that it's driven by community efforts and encompasses multiple languages. However, it differs from it in that it has a more sustainable data collection pipeline and a data validation step in place.
Multilingual Librispeech (MLS) [48]	This multilanguage dataset is built upon the English-only librispeech dataset [49], in order to encompass different languages that are often low-scale or scattered around different places, and rarely available, making use of LibriVox audiobook data [50], which is largely derived from the public domain texts of Project Gutenberg [51].

Continued on next page

Table 2.2 – *Continued from previous page*

Datasets	**Description**
Europarl-ST [52]	This dataset provides coverage for 9 European languages, including Portuguese, and is similar to VoxPopuli [53] in that it's based on European Parliament events. Specifically, it features debates that took place between 2008 and 2012. Although it was originally designed for speech-to-text translation tasks, the paired audio and text samples make it a valuable resource for speech synthesis applications.
NOS Sotaques	This proprietary dataset provided by NOS has Portuguese audios with a variety of accents from north to the south of Continental Portugal and even the archipelago of Azores.

For the Mozilla Common Voice dataset (version 16.1), we created a script that scans through the tsv data files to fetch the audios and transcripts corresponding to certain keywords in the accents' column such as Portugal, because we don't want the Brazilian accent. This process reduced significantly the number of audios resulting in a total of 1700 clips in mp3 format.

The Voxforge dataset went through a similar process but on the web interface. We searched for Portugal keyword resulting in a total of 18 datasets. Then, because the audios in each dataset didn't have a validation step as in Mozilla dataset, what we did was to apply a pipeline of noise reduction, because it was noticeable after listening to the audios a mouse click to stop its recording as well as some background noise in most audios. To accomplish this, we utilized the UVR5 interface [54], which leverages state-of-the-art source separation models to extract vocals from audio files. Specifically, we employed the UVR-DeNoise model from the VR architecture, with an aggression setting configured to 5. This particular setting fine-tunes the intensity of the software's vocal extraction process from an audio track. Increasing the aggression beyond 5 is reported to produce a muddied sound quality, as it can lead to a degradation of the overall audio clarity. This conservative approach aimed to mitigate potential biases

towards classifying the audio as synthetic. While this method successfully attenuated some background noise, we observed residual noise artifacts within periods of silence. To address this, we took an additional step and employed the pyannote voice activity detection (VAD) model to identify and exclude speech-inactive segments. This process was automated using the pyannote API, and the resulting text was formatted in the rttm format, providing information on the speech duration and speaker labels. We further converted this data into a pandas dataframe format for convenient manipulation and subsequently generated the audio using the pydub library in python. In certain cases, we also utilized the audacity tool to further refine the audio files and eliminate any remaining artifacts. This comprehensive process yielded a total of 182 paired audio-text wav files.

Regarding the Europarl-ST Portuguese dataset part, all the clips were European Portuguese so there was no need to apply any filters. However, there was a real need to shorten the audios since they had a duration of 1 minute or more. So, for this, we relied on two files found on the dataset, one with the transcripts and another with the audio filenames as well as the timestamps of the segments. Then with a script, we took that information and again generated the audio-text files which resulted in a total of 8950.

With this complete, what we did next was to generate a single tsv file that possessed all this information to then generate the audios using a TTS or STS service. Since a total of 10832 sentences seemed good enough, next we tried to ensure diversity in the dataset, meaning that we tried to gather a greater variety of recording conditions and accents as well as more voices from all genders to ensure that the model doesn't classify based on a specific feature [55]. And so, for that, we dived into the MLS and NOS Sotaques datasets. For the MLS dataset a total of 15 datasets in European Portuguese were found, corresponding to 10 voices in a total of 233 audio files of 10-20 seconds, while the NOS Sotaques has a total of 97 audios with durations that goes up to around 1 minute. In both cases, since there was no need for more transcripts we freely applied a pipeline using VAD in order to get a higher number of audio clips of shorter length,

which led to a total of 149 audios for NOS sotaques and 503 for MLS.

After finishing this, there was one final step remaining - normalizing the dataset. To accomplish this, we converted all audio files to the wav format. Subsequently, we downsampled the audio files to a sample rate of 16kHz, and finally converted them to mono format.

Now, for the fake/synthetic part of the dataset, the chosen speech-to-speech and text-to-speech services to generate the utterances are depicted in Table 2.3.

Table 2.3: Synthetic speech services

Service	**Number of used voices**	**Description**
Retrieval based Voice Conversion WebUI (RVC) [56]	3	The RVC is a voice conversion, easy to use service. It provides a simple web interface with the latest voice extraction algorithms such as RMVPE [57], that allows the use of multiple pre-trained voice models. It produces good audios, however it is really dependent in the original ones, meaning that if the input audio doesn't have a good quality the output won't be as good as otherwise.
ElevenLabs TTS [58]	3	This service holds a multilanguage model called Eleven Multilingual v2, that allows currently the generation of speech for more than 29 languages. It also produces realistic voices and permits a great diversity of accents namely European Portuguese. Nonetheless, our experience revealed certain limitations. These include restrictions on the number of characters permitted per month for generation and occasional occurrences of audio produced in different idioms such as Brazilian Portuguese and Spanish, despite training the voice on PT-PT data.

Continued on next page

Table 2.3 – *Continued from previous page*

Service	**Number of used voices**	**Description**
Google Cloud TTS [59]	4	The Google Cloud TTS is a tool that possesses support for various language. More importantly, it allows the use of pre-trained European Portuguese voices. For our use case, we used two voices from the standard service (C and D), which resorts to traditional TTS technology based on concatenative synthesis and parametric synthesis, and two voices from the wavenet service (A and B), which employ a deep neural network that directly modifies the raw audio waveform [60, 61]. Although being widely known, the results are nowhere near realistic audios produced from services such as ElevenLabs.

For starters, we decided to take a similar approach as in [15]. So we selected out of the 10832 sentences transcript only 1250 sentences in total wasting a total of 25028 characters per voice or more due to some parameter tuning that might be required, if we are not satisfied with the final result. This choice was made in ascending order of number of characters (we picked the first 1250 sentences), to somehow avoid the 100000 characters limitation imposed by elevenlabs and at the same time leave room for error because each generation wastes characters.

After this step, starting with ElevenLabs, we modified the python script found in [62]. First, it was defined the API key. Then we picked and trained three voices: one that we created using 20 seconds long clips from one random person in the MLS dataset; another that we trained with clips ranging from 1 to 4 minutes long from the Europarl-ST; the last one, had been created by NOS for the purpose of replicating the voice of one Portuguese celebrity (Pedro Abrunhosa) but turned out to have a decent performance so we sticked with it. After this, we made a call to its API to finally gener-

ate the audios using these trained voices and the tsv file containing the 1250 sentences. In total, we obtained 3750 utterances.

To utilize the Google Cloud service, we began by installing the Google Cloud SDK and logging in with our Google Cloud account. This process enabled us to acquire the essential keys for our Google Cloud project, providing us with access to the text-to-speech (TTS) functionality. Then, we created a script that using those keys and the default TTS parameters, generates a wav file for each sentence, similarly to what was done with elevenlabs. In total we acquired a total of 5000 audios.

For the RVC speech-to-speech service, we followed different steps to accomplish the synthetic voice:

- We transfered the audio files containing the 1250 sentences to a separate folder, and using UVR5, we employed an MDX-Net network called Kim Vocal 2 with additional vocal split mode options. This options involve a VR arquitecture, more specifically a Karaoke UVR model combined with the option to deverb vocals from lead vocals only. Additionally, after this process, we manually appended a flag to the audio file because of the transpose parameter.

- After processing the audios through this pipeline, the next step was to pick some voices to test this service. Out of the voices we stuck with two [63, 64] used for converting the human audios, and an additional one [65] to convert the Google Cloud Wavenet B voice audio to this last voice.

- Next, we set up the RVC. So, we extracted these voice models' folders containing a .pth and .index file, to logs directory on RVC project and copied the .pth to assets/weights.

- With all set, our next step was to create a simple script to convert the voices. So, with the gradio-client library installed on the python environment and infer-web script running in the background, we connected to the RVC client and used the API to convert each audio file taking into acccount that if we detect the flag, it means that the audio exhibits a high-pitched voice, and the parameters should

be adjusted accordingly.

At the end, we preprocessed this part of the dataset (16kHz and mono), and we manually filtered all audios in terms of retaining only PT-PT, because of some issues regarding the pronunciation of some words that were mainly found on the elevenlabs voices. This way we ensured that the models are focused on distinguishing between genuine and synthetic speech, eliminating any possibility of detecting idioms. Moreover, due to the use of different techniques, we ended up with a diversified dataset that has 4 different types of attacks. One that uses Elevenlabs (A1), another that uses Google cloud (A2), a third one (A3) which combines both TTS (Google Cloud) and STS (RVC), and the last that relies solely on RVC (A4). As a result, the final dataset ended up with 10566 synthetic and 11484 bona fide PT-PT audio files.

2.4 Data augmentation techniques

Data augmentation plays a vital role in enhancing the performance and generalization capabilities of machine learning models.

The datasets mentioned in Section 2.2, suffer from a lack of generalization ability when it comes to unseen utterances. This issue arises because each dataset has its own set of spoofed and bona fide samples, which leads to a degradation in performance during cross-dataset studies [8, 9, 10], e.g., training on ASVspoof2019 LA subset and testing on FoR-norm dataset. Hence, selecting an appropriate data augmentation procedure is crucial.

This degraded performance can be attributed to several factors [8, 9] such as: The unseen attacks present in some datasets pose greater challenges; The linguistic variation aspect; and the variability of recording conditions, including background noise and channel effects across dataset, which involves the audio alterations imposed onto the speech signal, such as the reverberation from recording environments and compression algorithms [8]. While the first one hardly depends on the dataset, the other two might be overcome by data augmentation techniques. These techniques aim to in-

crease the quantity and diversity of the dataset by applying various transformations to the existing data. These are useful because they generate new training examples that are variations of the original data, while preserving their labels.

Therefore, from the insights and success of them, in the papers [8, 9], the utilization of the acoustic simulator [66] could potentially be paradigm changing. This simulator introduces three types of degradation processes: additive noise, telephone and audio codecs, and device/room impulse responses (IRs). These processes contribute to the development of a more channel-robust model, preventing overfitting to the limited channel effects presented in the training set [8]. As an alternative option, in [10], some "cheap procedures" have been listed including: Gaussian noise addition (GNA), Signal-to-noise ratio noise addition (SnrNA), time shifting, pitch shifting and time stretching. Although explicit explanations for their effectiveness in synthetic speech detection is lacking, these techniques have been reported to yield state-of-the-art performance in audio classification tasks.

2.5 Feature extraction and evaluation metrics

To serve as input for our synthetic speech detection (SSD) models, there are plenty of interesting hand-crafted features, however most of the works decided to implement the ones that are the baseline for ASVspoof challenges, in this case, LFCC [16, 8, 9] and CQCC [16, 10]. Nonetheless, it's important to note that the selection of these features does not guarantee optimal performance. Different cases may benefit from alternative choices, such as [16], which tried both but achieved the best overall performance using a pre-transform (CQT).

As for evaluation metrics, the primary measure used by most is the Equal Error Rate (EER), which is a widely used metric for the ASVspoof challenges, determined by the point at which the false positive rate and false negative rate are equal to each other, meaning that if the EER is low, the system is making fewer errors in both directions. Other common metric is the tandem detection cost function (t-DCF) [16, 9], because it

indicates the reliability of the SSD system on ASV. Outside of this ASVspoof context, the accuracy [15], which provides a measure of overall correctness and area under the receiver operating characteristic (ROC) curve [10], that indicates how well the model is capable of distinguishing between classes, may be sufficient for most of the works.

2.6 SSD models' architectures and training strategies

Several architectures have been introduced to tackle the ASVspoof challenge. Each comes with its own set of advantages and disadvantages. Some try to propose innovative solutions at the architectural level [16], while others focus on optimizing loss functions and employing data augmentation techniques [16, 8].

A common trend among these methodologies is the incorporation of the ResNet architecture into the models. This preference is due to the beneficial features of residual blocks, which enable the training of very deep networks by mitigating issues related to vanishing gradients in deeper layers. Additionally, ResNet simplifies the optimization process and enhances overall performance through skip connections, which ensure the smooth flow of information throughout the network and promote the accurate capture of audio patterns.

Several models exemplifying this trend are:

- The Res-TSSDNet (Figure 2.7), which is an open-source model employing a Time-domain Synthetic Speech Detection (TSSD) Net with ResNet structure that uses raw waveforms for synthetic speech detection [5]. Its training scheme involves the use of the cross-entrophy loss function with mixup regularization [5, 27] to improve the generalization capability to unseen attacks. The advantage of using this model is related to its lightweight nature.

- The ResNet-OC (Figure 2.8), that is another open-source architecture. It has an embedding network composed by the ResNet18 and an attentive pooling layer designed to learn the speech embeddings [8]. The CM Classifier composed by a Fully Connected (FC) layer is responsible for classifying the embeddings into two

categories: spoofing attacks or bona fide (genuine) speech. Its training scheme involves the use of the OC-Softmax loss function [67].

- Following the work of [8], this Emphasized Channel Attention, Propagation and Aggregation Time Delay Neural Networks (ECAPA-TDNN), in Figure 2.9, based on the original TDNN introduces several enhancements, including squeeze-excitation (SE) blocks, multi-scale Res2Net features, multilayer feature aggregation, and channel-dependent attentive statistics pooling [9]. In contrast to ResNet-OC, it leverages Res2Net blocks, which are modified bottleneck blocks from ResNet, that enable multiple feature scales by dividing feature maps into channel groups with inter-group connections [16]. The SE module is used to re-scale feature map channels using global context information, through the use of an adaptive average pooling layer, in order to improve feature learning [16]. Similarly to ResNet-OC, it relies on OC-Softmax loss function and the classifier corresponds to a FC layer.

- TE-ResNet (Figure 2.10), which combines a Transformer Encoder for handling long-term dependencies and a 34-layer ResNet to compute deep features and calculate the score to decide if the speech is fake. It has the downside of requiring a lot of resources due to its size (29.3M parameters), however it's an architecture worth exploring since it's the base for some state-of-the-art models such as ChatGPT. In its implementation, based on [10], we relied on python 3.7 and essential libraries like pyTorch. Additionally, we adapted the Transformer [68] to accelerate the model's construction and followed a similar training approach to [8, 9], where we first extracted the LFCC features and then fed directly to the model.

Apart from the model architecture and the loss function, normally its chosen the dataset and a batch size, which typically consists of 64, 32 or 16 samples at 16 kHz. Then, the number of training epochs is picked along with a learning rate that changes over time. Finally, regarding the optimizer, most of the authors adopted the Adam [69] for their experiences, considering it the most robust option compared to others.

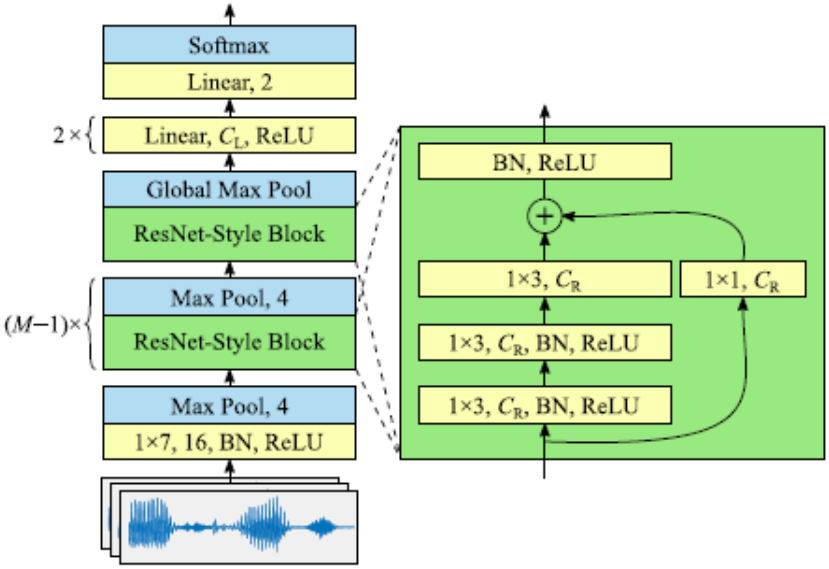

Figure 2.7: Res-TSSDNet Architecture (Reproduced from [5])

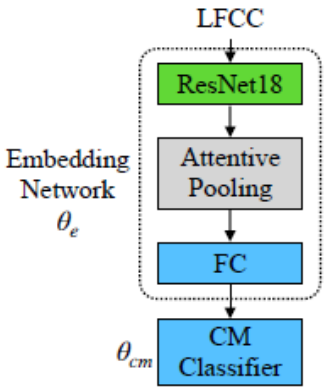

Figure 2.8: ResNet-OC Architecture (Adapted from [8])

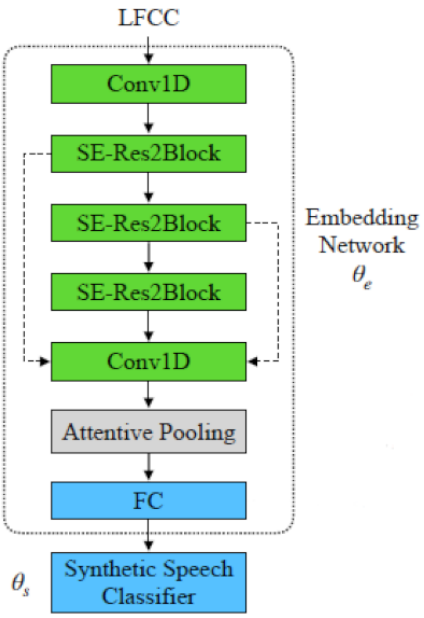

Figure 2.9: ECAPA-TDNN Architecture (Adapted from [9])

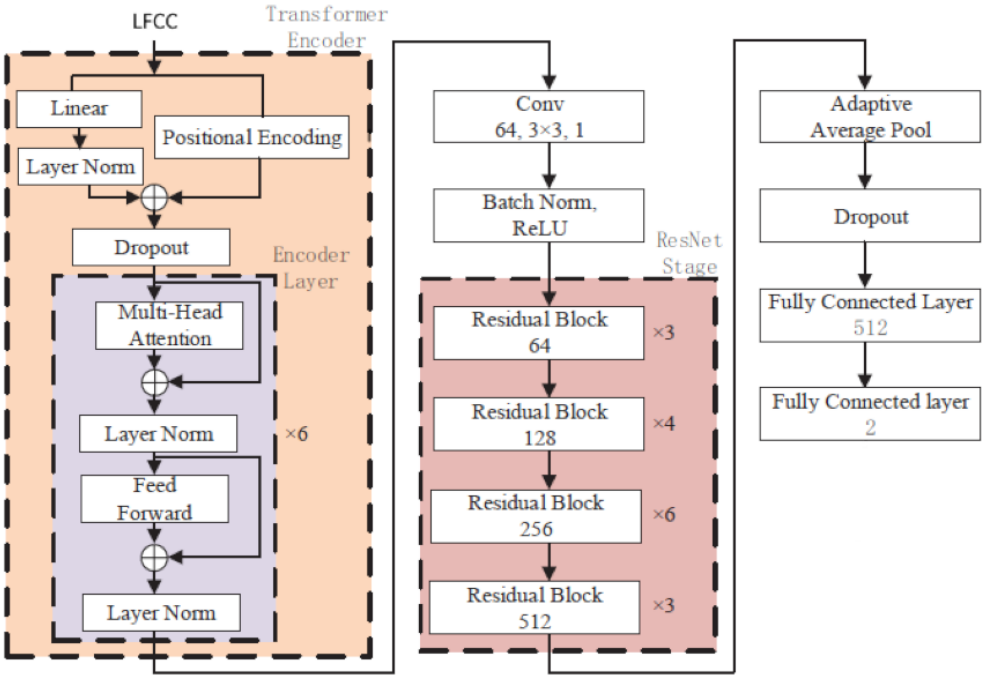

Figure 2.10: TE-ResNet Architecture (Based on [10])

2.7 Fusion strategy

To further enhance the performance of speech detection systems, some authors [9, 10], decided to employ a multi-model fusion strategy. This technique can be found in [10, 70], and it basically involves integrating scores from multiple detectors that utilize different front-end acoustic features and back-end classifiers, via logistic regression, in order to obtain a final score, as shown in Figure 2.11.

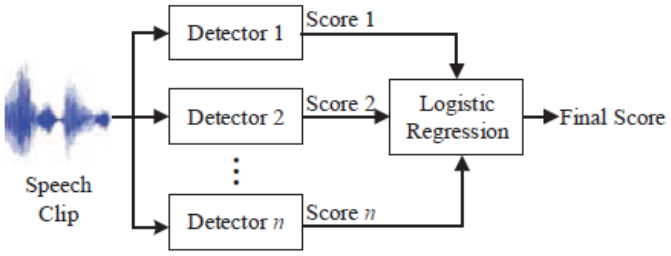

Figure 2.11: Score-level fusion by logistic regression (Reproduced from [10])

2.8 Fraud model's architecture

Some cases demonstrate positive applications for cloned voices, such as the cinema industry utilizing synthetic voices for voiceovers in films and animations. However, it is crucial to acknowledge that not all of them are harmless, as individuals may exploit them for deceptive purposes. Consider a scenario where someone receives an audio message. Typically, a SSD would be employed to identify if the audio is a cloned voice. If the detector successfully detects a synthetic voice, people can be made aware of this fact. However, there is always a possibility that the detector may fail to recognize synthesized speech. In such instances, particularly if fraudulent content is involved, it poses a significant security risk.

Recent scams such as in [71], can further illustrate this issue. By combining audio with video the effectiveness of this particular scheme was notorious, affecting thousands of people.

So, to address this concern, we propose a scheme that combines the SSD with the Broad Learning System (BLS) (Figure 2.12). This way, significant improvements in user

safety and protection can be made, by leveraging the incremental training strategy from the BLS that allows to refine the model on new received fraud attempts without fully retraining it.

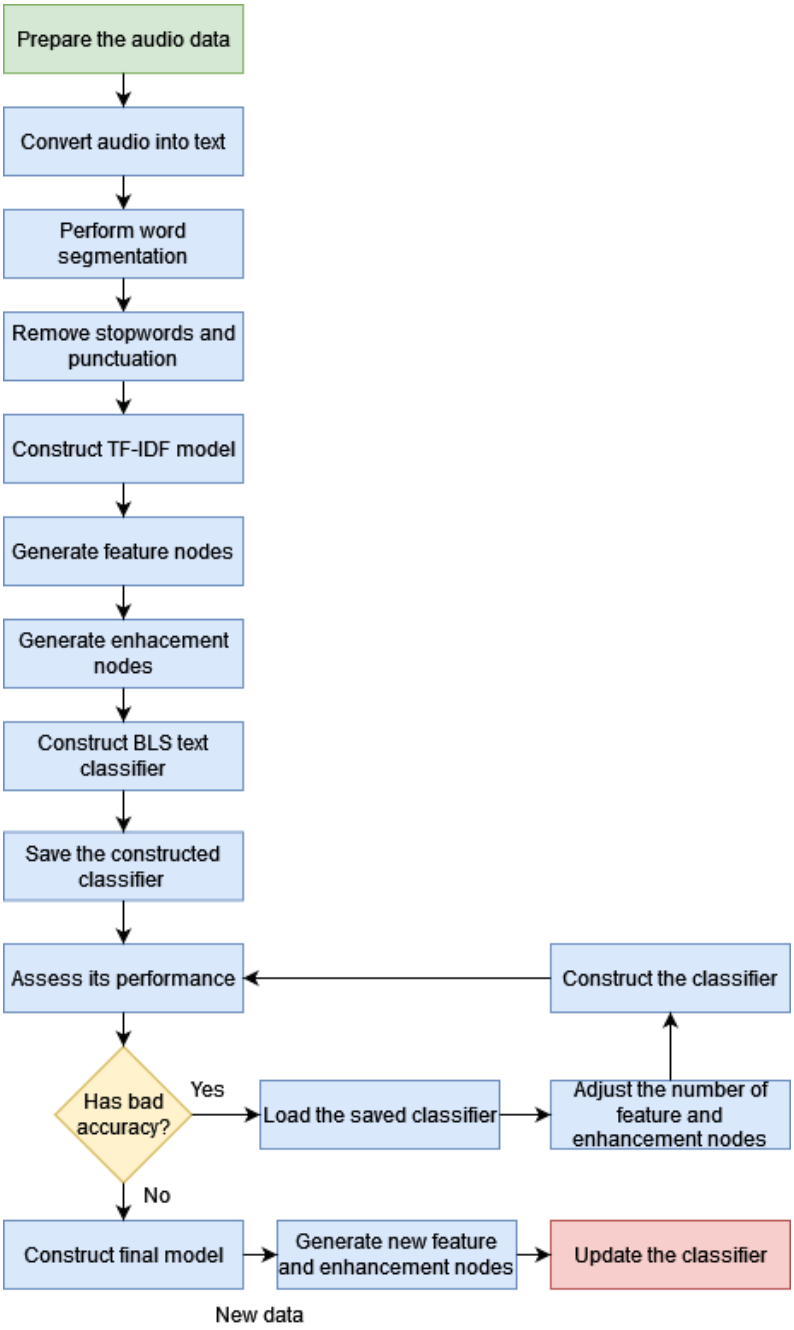

Figure 2.12: Broad learning system based on text classification (Based on [11])

3

Development, methodologies and validation

The purpose of this chapter is to outline the methodology adopted for the primary goal of developing a system capable of identify natural and synthetic speech. So, in the following section, we will describe, and detail the specific decisions made to train and evaluate the models. Also, decisions regarding the application of statistical tests and fine-tuning can be found in Subsection 3.1.3 and 3.1.4, respectively.

The code used here is not publicly available, therefore its access can be provided upon request at https://github.com/Rgsantos12.

3.1 Methodology

After reviewing the approaches taken by researchers such as [8] and [9], our strategy involves practical experimentation with each model. Our objective is not only to evaluate the models' capability to distinguish between natural and synthetic voices under different configurations such as different loss functions, data augmentation procedures and other relevant factors, but also to gain a deeper understanding of developing a language-agnostic model. Specifically, how to extend the research done by others, to PT-PT, with the restriction of having few datasets in that language and few studies made on multilanguage datasets.

To address these challenges, we made our own SSD PT-PT dataset, as described in Section 2.3. We trained the models on the ASVspoof2019 LA dataset, with and without data augmentation, to investigate the language agnosticism between PT-PT and English and to identify the optimal settings for each model. Also, 6 runs were made for each experiment to strike a balance between obtaining a sufficient amount of results for analysis and conclusions, and managing the time required to complete all the runs from each architecture. These were conducted on both my local machine equipped with a nvidia rtx 2070 as well as 3 nvidia T4 gpus using predefined seeds to ensure reproducibility and consistency of results.

In the testing phase, we incorporated other datasets to assess which models generalize better to unseen attacks, choosing also the ASVspoof2015 and FoR-norm datasets due to their recognition in the research community and their relevance to our objectives. Notably all these datasets were already pre-divided into training, development and testing sets upon acquisition. Table 3.1 details the number of samples for each one.

Table 3.1: Number of samples on each dataset

Dataset	**Type**	**Training set samples**	**Development set samples**	**Evaluation set samples**
ASVspoof2019 LA	Genuine	2580	2548	7355

Continued on next page

Table 3.1 – *Continued from previous page*

Dataset	**Type**	**Training set samples**	**Development set samples**	**Evaluation set samples**
	Synthetic	22800	22296	63882
ASVspoof2015	Genuine	3750	3497	9404
	Synthetic	12625	49875	184000
FoR-norm	Genuine	26941	5400	2264
	Synthetic	26927	5398	2370
SSD PT-PT	Genuine	*	*	11484*
	Synthetic	*	*	10566*

* The whole set was used for testing, however when fine-tuning, this dataset was splitted

After deciding which classifier has the best performance in all these datasets through statistical tests, we then selected the best-of-the-runs models and fine-tuned them to the PT-PT language.

3.1.1 Preliminary experiments

For this first experiment (Figure 3.1), we opted to start with the Res-TSSDNet, given its lightweight nature and ease of training. In this initial experiment, equal-duration raw waveforms were generated from the ASVspoof2019 training set, each sample lasting 6 seconds with a default sampling rate of 16 kHz. The model was then trained with a batch size of 32 for 100 epochs, employing mixup using the following formula in [5]:

$$\tilde{x}_i = \lambda x_i + (1-\lambda)x_j, \quad \tilde{y}_i = \lambda y_i + (1-\lambda)y_j, \tag{3.1}$$

where x_i, y_i and x_j, y_j are two randomly selected training pairs, $\lambda \sim$Beta(α, α) and α=1. Leading to the final loss function:

$$CE_{mixup}(\tilde{z}, y_i, y_j) = \lambda CE(\tilde{z}, y_i) + (1-\lambda)CE(\tilde{z}, y_j), \tag{3.2}$$

where $\tilde{z}$ contains the softmax probabilities from the mixed samples and CE(·,·) is the standard cross-entrophy (CE) loss.

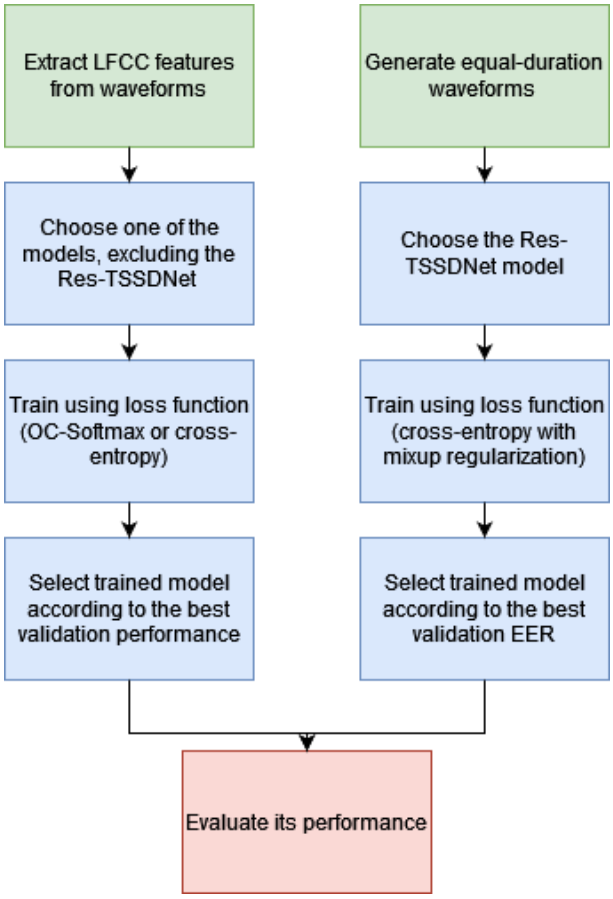

Figure 3.1: Scheme for the preliminary experiment

Next, the second model to test was the ResNet-OC. The strategy utilized is the augmentation (AUG), Figure 2.8. We opted for this approach due to a favorable trade-off between time cost and performance found in the results in [8]. This strategy uses the ASVspoof2019 LA dataset and an acoustic simulator to create a channel augmented dataset accessible from [72]. Within this dataset, all utterances undergo 12 distinct device/room impulse responses (IRs) for the development set and 10 for the training set, respectively creating 298128 and 253800 samples. Subsequently, LFCC features are extracted, resulting in a matrix of 750 frames x 60 coefficients. The model is then trained with a batch size of 64 for 100 epochs, using the OC-Softmax loss function.

In the case of the ECAPA-TDNN model, we employed the same acoustic simulator to construct the training and development set. However, instead of incorporating device impulse responses (IRs), we focused solely on the compression codec options,

which include the Fraunhofer MPEG layer III (MP3) and advanced audio codec (AAC). This choice aligns with our task, which resembles the ASVspoof 2021 Deepfake task, where our objective is to detect the authenticity of audio rather than confirming the speaker's identity [14]. For this first trial, we opted to exclusively utilize the 6 options out of the 24 compression codecs in total without the use of 12 device IRs after the compression codecs, contrary to the paper's recommendation [9]. This choice allows us to explore how the models react when using the IRs (on Resnet-OC) and compression codecs separately. The LFCC features were then extracted with the same configuration and then we trained the ECAPA-3 model from [9] with OC-Softmax loss for 100 epochs with a batch size of 64.

These three open-source models proved helpful, providing flexibility for customization. Studying their designs served as a guide in creating the model, especially when the source code for TE-ResNet was unavailable. In our implementation (explained in Section 2.6), the training scheme is similar to the previous two models. LFCC features were extracted from the ASVspoof2019 LA training set, resulting in a matrix of 63x60 coefficients. Cross-entropy with a batch size of 32 and 100 epochs was then used to obtain the trained model.

Given these architectures, the model from each one with the best validation performance or lowest Equal Error Rate (EER) in the case of the Res-TSSDNet on the respective development set was then chosen for testing.

3.1.2 Further investigation

From the preliminary experiments, some insights that will be explained in Section 4.1 were obtained about some combinations that might work best for the models:

- As a starting point, we decided to go with the cross entrophy using mixup regularization as the loss function. This, because Res-TSSDNet showed a really good cross dataset performance for ASVspoof2019 and 2015. Next, we went with the same compression dataset used in the ECAPA model, expecting that by joining the best of the two worlds, we could elevate the performance of all models.

- Another modification, has to do with the number of frames extracted using LFCC for the TE-ResNet. Since we want specifically to improve this models' performance, we were able to increase the 63x60 matrix to 126x60 in order to let the model learn more frames and potentially improve its predictive performance.

- Last but not least, the introduction of early stopping and of a fixed batch size at 16 samples. This way it prevents the model from overfitting and at the same time save computational resources. Moreover, fixing the batch size for all models granted a better control on GPU memory usage.

The overall scheme for this second experiment is represented in Figure 3.2.

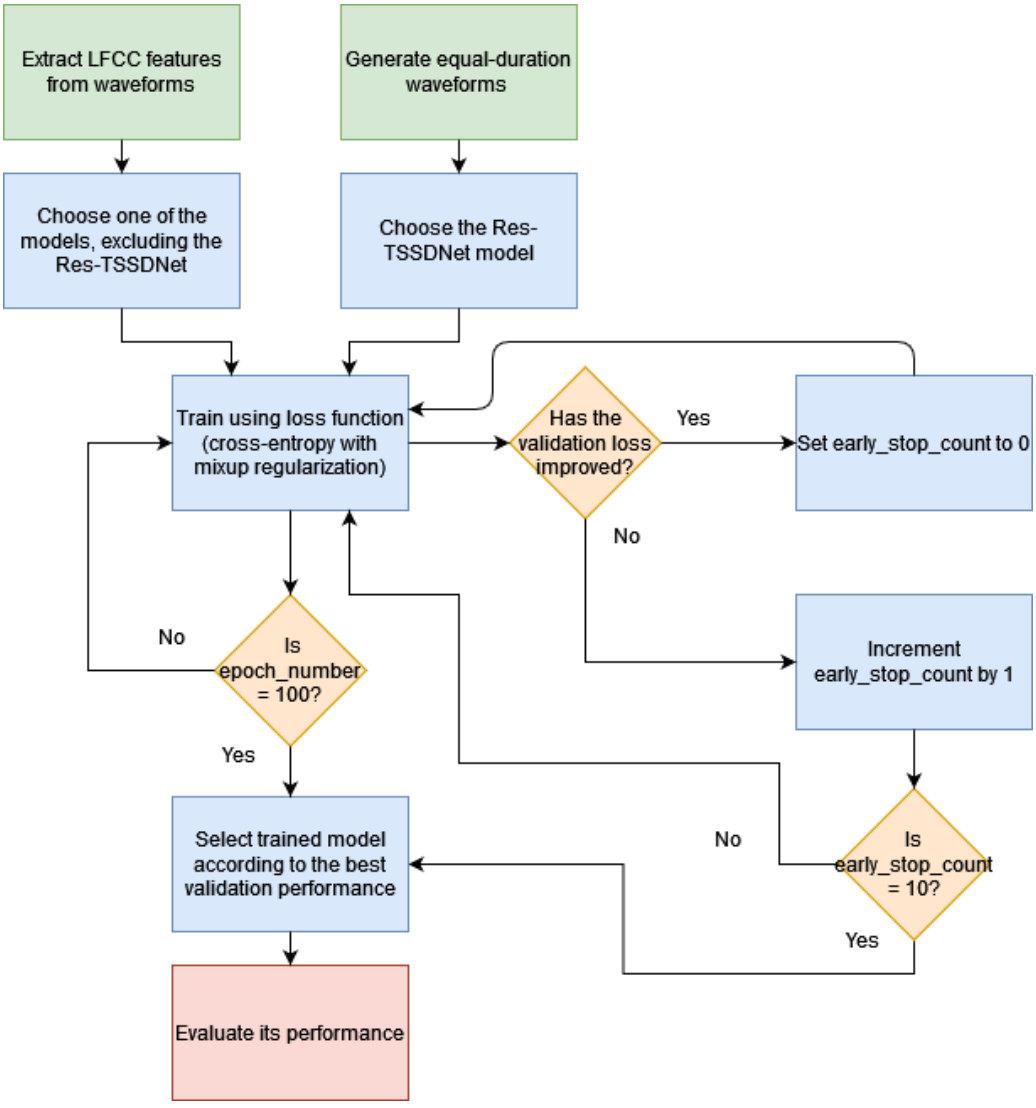

Figure 3.2: Scheme for the second experiment

For the third experiment given the new insights that will be detailed in Section 4.2, few changes were made to the overall methodology:

- Regarding the augmented dataset, we have added a second pipeline which involves the use of IRs devices [9] that is said to improve the classification of bona fide trials [8]. So, in a list of 12 IRs device, we randomly selected one for each of the audio files from the compression augmented dataset.

- Both ECAPA and ResNet-OC models with mixup, had trouble to generalize to attacks that it wasn't trained on. Due to this, we went back from using mixup to OC-Softmax loss function on these two specific models. We also kept Res-TSSDNet and TE-ResNet with mixup, in the hopes that with this new augmented dataset we can drastically improve its performance. Later, it was added TE-ResNet with OC-Softmax as well, to see if there is other factors influencing its performance or if the problem was related to the loss function.

In this final experiment, we used the empirically found optimal factors for each model architecture. Therefore, after obtaining the results for each used metric, it was time to proceed with selecting the best overall model.

3.1.3 Statistical analysis

For selecting the best classifier, we applied a series of statistical tests. These tests, which are commonly incorporated into the evaluation of algorithms, are used to determine which algorithm is better than another [12].

Statistical tests are divided in two groups: parametric and non-parametric tests, and their usage depends on the problem conditions [12]. Parametric tests are the most common techniques in computer science, but they are based on several assumptions/-conditions that must be fulfilled to use them:

- First, independence, that is two events are independent if the occurrence of one of them does not modify the probability of the occurrence of the other one.

- Normality, which is achieved when a collected data follows a normal distribution with mean μ and variance ρ.
- Homoscedasticity, i.e., equal variance for distributions in analysis.

The scheme in Figure 3.3, was used for this evaluation process. We first defined the confidence level of 95%, and the groups, which were the average of runs for both specificity and sensitivity in all datasets. Then, we checked for normality using Shapiro-Wilk test and homoscedacity with the median version of Levene's test.

After finding out whether these conditions were met, we applied either a parametric test (ANOVA) or non-parametric test such as Friedman [12]. We then proceeded for the post-hoc test [73] and finally, it was selected the best classifier according to the ranking value.

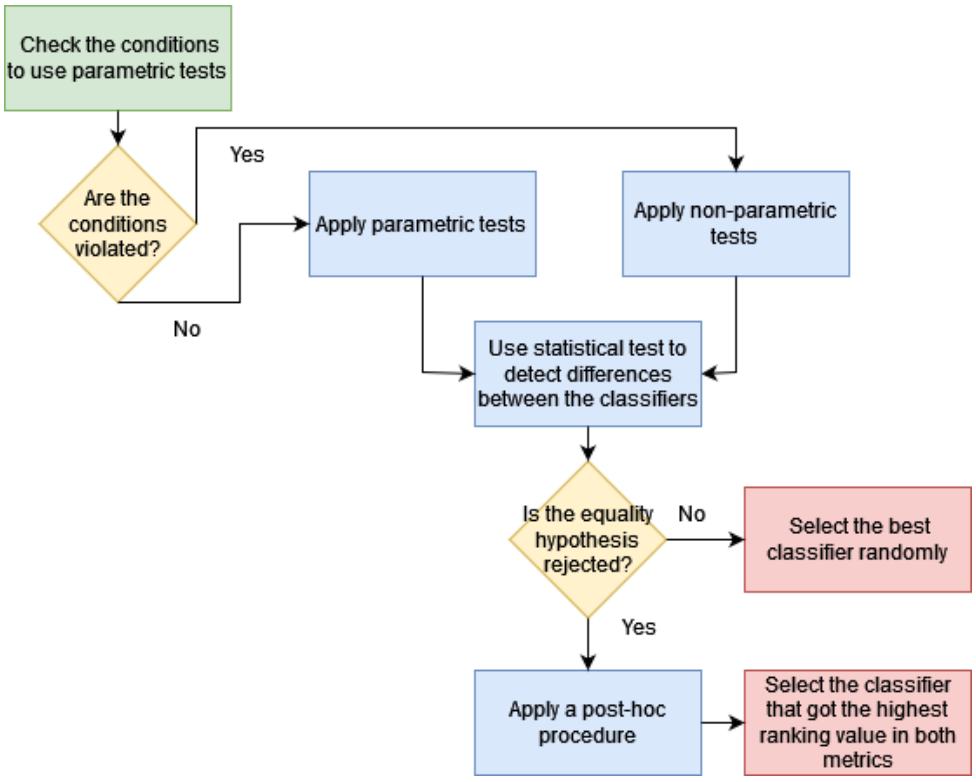

Figure 3.3: Process of selecting the best classifier (Based on [12])

3.1.4 Optimization and fine-tuning

After finding the best classifiers, it was chosen the best models of those classifiers, based on the average specificity and sensitivity on the English datasets. We then split

the PT-PT dataset into training, testing, development as shown in Table 3.2. To accomplish this we first divided the set composed by genuine, A1 and A2 attack samples, giving 70% for training and development and 30% for testing. Then, we further split these same attacks leaving 75% and 25% for the training and development set respectively. Lastly, we joined the 30% from the genuine, A1 and A2 testing set with what was left to complete it, namely the A2, A3 and A4.

Table 3.2: Divison of SSD PT-PT dataset

Subset	**Number of utterances**			**Service**
	Train	**Dev**	**Test**	
Genuine	6028	2010	3446	None
A1	957	319	547	ElevenLabs
A2	2625	875	1500	GCloud
A3	0	0	1249	GCloud+RVC
A4	0	0	2494	RVC

After this, we trained our top-performer models for 2 epochs, using the train and dev subset of the SSD PT-PT. This was done to complete our major goal of having a model capable of distinguishing in both PT-PT dataset and English-based datasets. Finally, we tested on all datasets to see how it would react to all unseen attacks.

4

Experimental results

The purpose of this chapter is to present a comprehensive analysis of the outcomes obtained from the experimental setup detailed in Chapter 3. By examining both the performance with respect to some metrics and the findings of the conducted experiments, this chapter aims to provide valuable insights and pave the way for further discussion.

For the evaluation stage, metrics such as accuracy were computed based on the confusion matrix obtained from scikit-learn. Figure 4.1 illustrates the resulting matrix. Here, true positives (TP) and true negatives (TN) represent the count of correctly classified samples as synthetic and as human/bona fide, respectively, while false positives (FP) is the number of instances where human samples are misclassified as synthetic, and false negatives (FN) the number of samples where synthetic samples are misclas-

sified as human [13, 74].

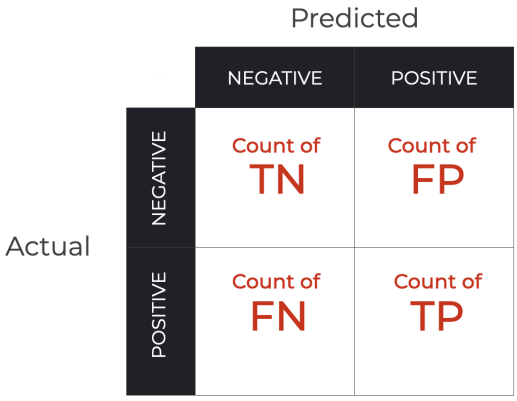

Figure 4.1: Confusion matrix (Reproduced from [13])

With that said, the chosen metrics for evaluation are the following:

- The EER measures the trade-off between security and usability, so a lower EER, in this case, indicates a more reliable SSD [75]. It's defined by the following formula [76]:

$$EER = \frac{FAR + FRR}{2}, \tag{4.1}$$

where

$$FAR = \frac{FP}{H}, \tag{4.2}$$

and

$$FRR = \frac{FN}{S}. \tag{4.3}$$

Here, H correpond to the number of the human instances, S is the number of all synthetic instances respectively. Also, false acceptance rate (FAR) and false rejection rate (FRR) are scalars that minimize abs(FAR - FRR).

- The accuracy measures the proportion of correct predictions among all predictions made by the model. The formula is:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{4.4}$$

- The specificity and sensitivity (also known as recall) is computed to obtain a more reliable score given that most datasets are unbalanced. Sensitivity measures the proportion of true positive predictions among all actual positive instances in the dataset, while specificity does the same for true negatives among all actual negative instances [77]:

$$Specificity = \frac{TN}{TN + FP}, \tag{4.5}$$

$$Sensitivity = \frac{TP}{TP + FN}. \tag{4.6}$$

4.1 Preliminary results

For these first experiments, after training and retrieving the models, we then tested with no predetermined order for the testing sequence. Regarding the hyperparameters crucial to the training and evaluation of these models, these were chosen based on the specifications outlined in the respective papers corresponding to each model described in the prior Section 3.1.1.

Figure 4.2: Preliminary experiment - ASVspoof2019 LA evaluation set

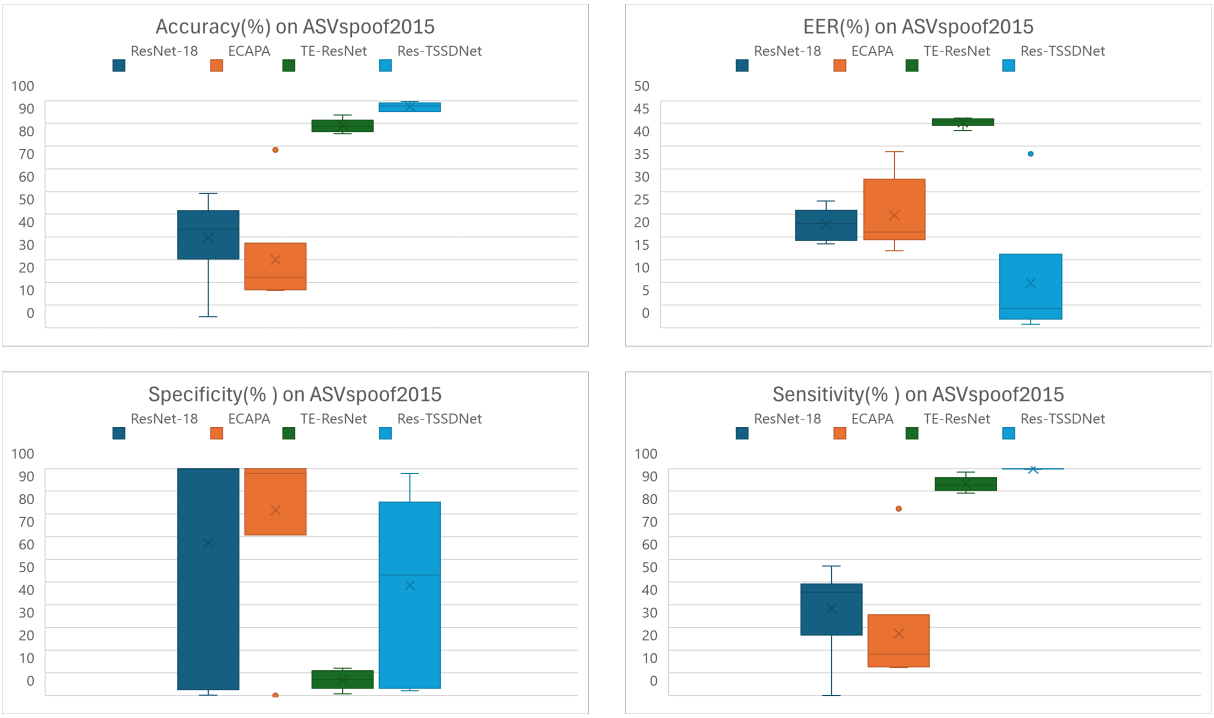

Figure 4.3: Preliminary experiment - ASVspoof2015 evaluation set

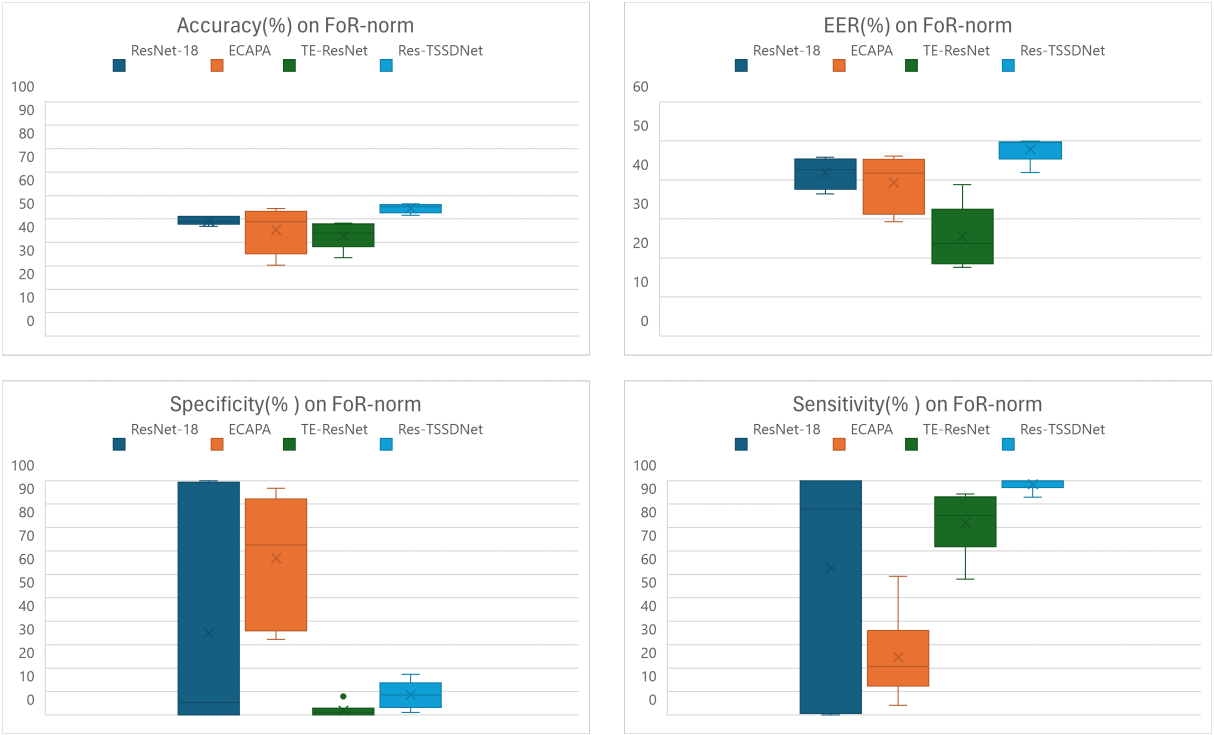

Figure 4.4: Preliminary experiment - FoR-norm evaluation set

Figure 4.5: Preliminary experiment - SSD PT-PT evaluation set

Starting off and according to the conditions detailed in Subsection 3.1.1, the results obtained from each model can be found in Figures 4.2, 4.3, 4.4 and 4.5.

Looking at Figures 4.2 and 4.3, Res-TSSDNet seems to perform best in ASVspoof2015 and 2019 but showed worst performance in FoR-norm and SSD PT-PT datasets. This discrepancy is likely attributed to the differing characteristics of samples [44, 55]. Since the Res-TSSDNet was exclusively trained on ASVspoof2019 without any data augmentation techniques, apart from mixup which was employed to aid the generalization process [78], the model appears to have overfit [79] to the ASVspoof2019 LA data complicating the task of detecting synthetic samples effectively for the other datasets.

Regarding the ResNet-OC, unlike the previous model (Res-TSSDNet), it has demonstrated a lot of variability, sometimes classifying only the bona fide samples well and other times the opposite, as we can see in the columns denoted by TN and TP from Tables 4.1, 4.2, 4.3 and 4.4 where we have values, e.g, ranging from 20 to 7355 and 0 to 59754 in the ASVspoof2019 LA evaluation set. This inconsistency show that the ResNet-OC model might not be as robust as the others. Also, and as suggested in [8], the channel-augmented dataset showed some evidence that it might help in the

classification on bona fide trials, given the specificity results on all datasets.

Table 4.1: ResNet-OC results for the ASVspoof2019 LA evaluation set

	TN	**FP**	**TP**	**FN**
Run 1	498	6857	27495	36387
Run 2	7150	205	59754	4128
Run 3	7329	26	53947	9935
Run 4	20	7335	9766	54116
Run 5	7355	0	0	63882
Run 6	7328	27	55539	8343

Table 4.2: ResNet-OC results for the ASVspoof2015 evaluation set

	TN	**FP**	**TP**	**FN**
Run 1	329	9075	85221	98779
Run 2	9380	24	105043	78957
Run 3	9399	5	65381	118619
Run 4	5	9399	82455	101545
Run 5	9404	0	0	184000
Run 6	9399	5	85440	98560

Table 4.3: ResNet-OC results for the FoR-norm evaluation set

	TN	**FP**	**TP**	**FN**
Run 1	10	2254	2219	151
Run 2	0	2264	2370	0
Run 3	233	2031	1938	432
Run 4	2246	18	22	2348
Run 5	2264	0	0	2370
Run 6	0	2264	2370	0

Table 4.4: ResNet-OC results for the SSD PT-PT evaluation set

	TN	**FP**	**TP**	**FN**
Run 1	1427	10057	4560	6006
Run 2	987	10497	10563	3
Run 3	7335	4149	9347	1219
Run 4	9451	2033	49	10517
Run 5	11484	0	0	10566
Run 6	2027	9457	10375	191

Moving on to the ECAPA-TDNN, it seems that most synthetic samples were incorrectly classified, when compared to the other models for both ASVspoof 2015 and LA 2019 evaluation set, as we can see by the median value in the sensitivity metric plot (Figures 4.2 and 4.3). This is likely attributed to the fact that we have added some modified samples using compression techniques. Nevertheless, it seems that this particular data augmentation process might have a positive impact given the fact that this model outperforms the others in discriminating both spoofed (TP) and bona fide (TN) samples for FoR-norm and SSD PT-PT datasets. This is evident from the Tables 4.5 and 4.6 when comparing to ResNet-OC, as well as from the Figures 4.4 and 4.5.

Table 4.5: ECAPA-TDNN results for the FoR-norm evaluation set

	TN	**FP**	**TP**	**FN**
Run 1	840	1424	1403	967
Run 2	2052	212	397	1973
Run 3	730	1534	672	1698
Run 4	1940	324	585	1785
Run 5	2190	74	98	2272
Run 6	1348	916	355	2015

Table 4.6: ECAPA-TDNN results for the SSD PT-PT evaluation set

	TN	**FP**	**TP**	**FN**
Run 1	10820	664	5934	4632
Run 2	11373	111	696	9870
Run 3	2961	8523	8101	2465
Run 4	11345	139	1832	8734
Run 5	11135	349	1463	9103
Run 6	10761	723	322	10244

Finally, regarding the TE-ResNet model, we can see that the EER is too high overall across all box plots (Figures 4.2, 4.3, 4.4 and 4.5). This might be related to the number of frames in LFCC, because to end up with a fixed length of only 63 frames, most of the samples were trimmed, leading to a faster model training in exchange of a greater performance. Despite this and looking at the rest of the metrics, it seems that the performance was really similar to the Res-TSSDNet in terms of the accuracy across all datasets. Given that only cross-entropy was employed, maybe with mixup, it can get a little better due to mixup's capability to generalize and create samples.

So, in summary, two major observations can be taken from this experiment:

1. The selection of the loss function is crucial. While OC-Softmax demonstrated reasonable performance for both Resnet-OC and ECAPA-TDNN, the mixup could potentially yield better results. This is attributed to its capability to generate a larger number of samples, which is particularly advantageous given the higher prevalence of spoofed samples compared to bona fide ones in the original dataset.

2. The choice of the data augmentation procedure is also important. Since the performance of the ECAPA on all datasets wasn't the best, we propose to instead of using separately, the compression codecs or the IRs, the use of a pipeline composed by both, similar to what was done in [9], to remove the dataset's bias and consequently improve its generalization ability towards unseen utterances.

4.2 Additional results

In this section, we present the results for the remaining experiments, separating them from the preliminary tests aimed at understanding the performance of various models under different conditions. Our objective here is to apply the knowledge acquired from the previous Section 4.1, identify the optimal conditions for each model and determine which model performed best overall across these diverse datasets. Also, finding if its possible to have a language agnostic model is one of our priorities too.

For the second experiment (Figures 4.6, 4.7, 4.8 and 4.9), we tried to study and confirm the impact of the mixup loss function as well as of the data augmentation technique (compression codecs).

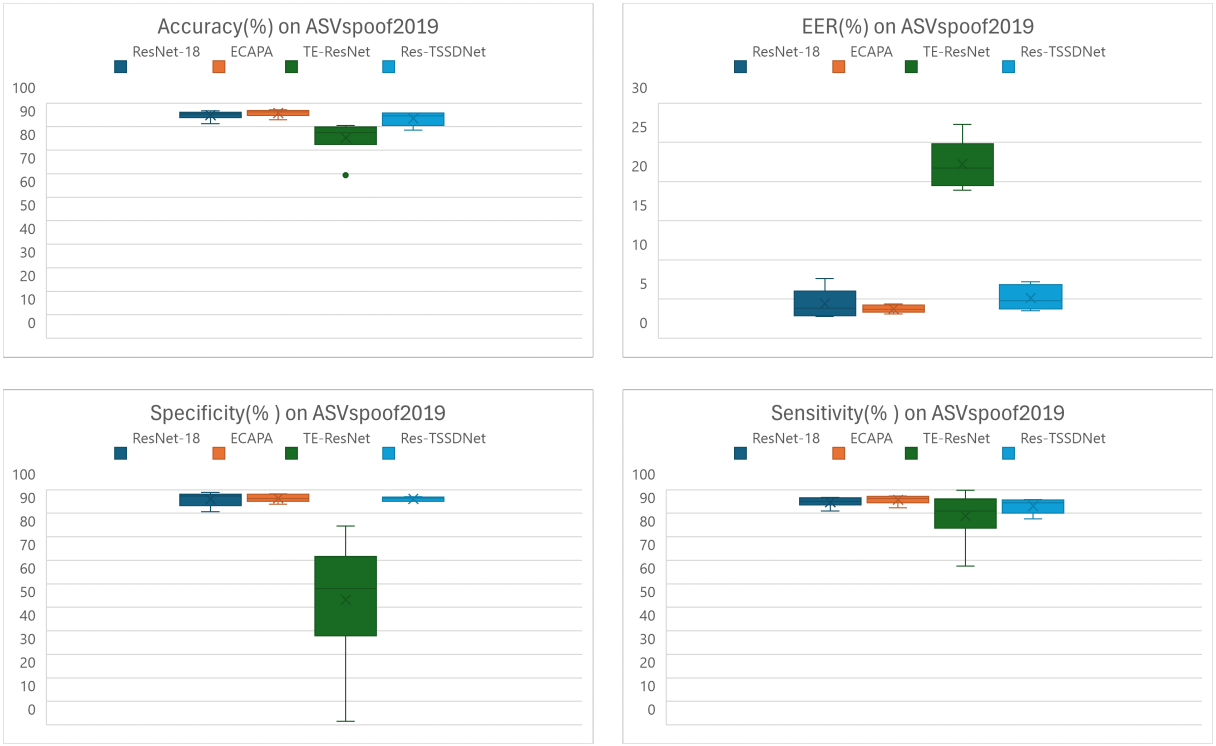

Figure 4.6: Second experiment - ASVspoof2019 LA evaluation set

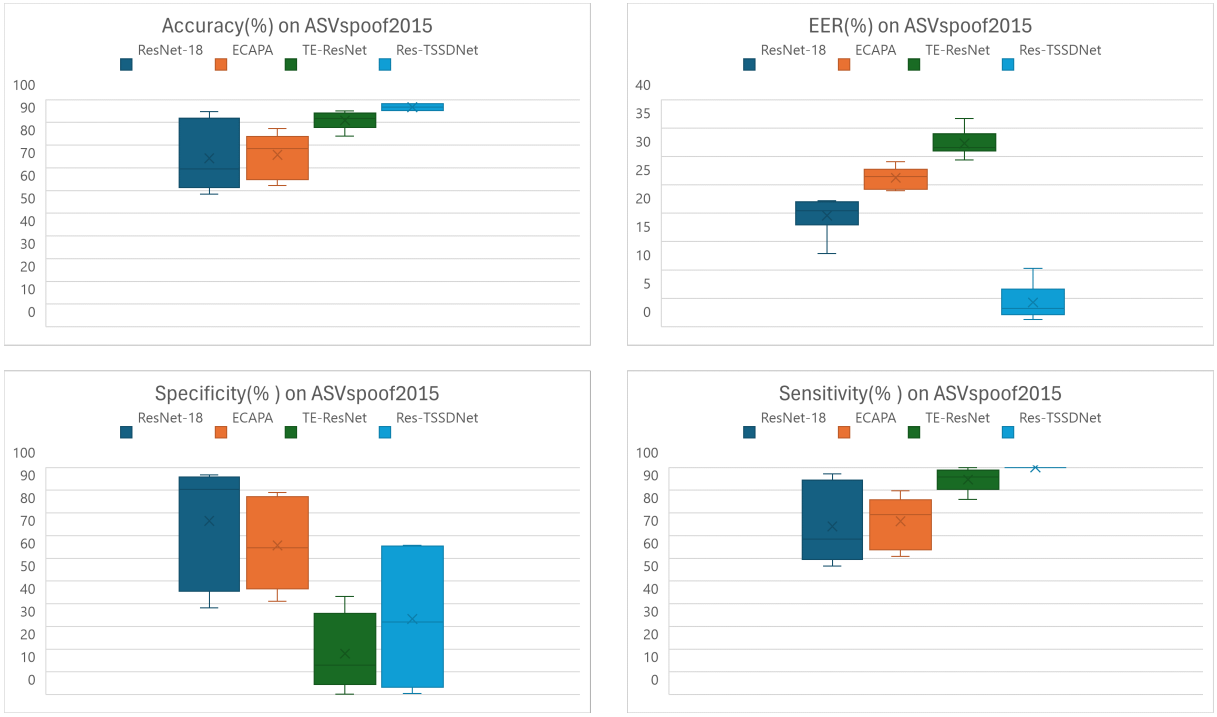

Figure 4.7: Second experiment - ASVspoof2015 evaluation set

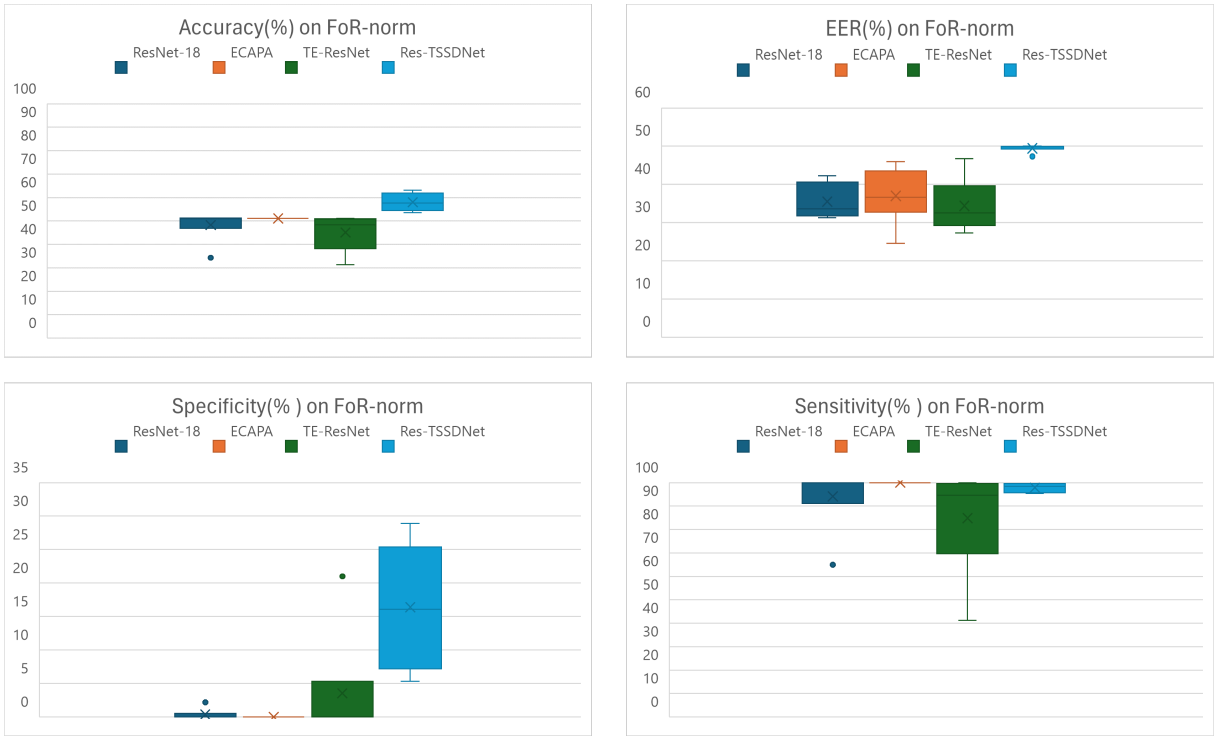

Figure 4.8: Second experiment - FoR-norm evaluation set

Figure 4.9: Second experiment - SSD PT-PT evaluation set

From these figures, we can see that using mixup, ECAPA and ResNet18-OC are prone to overfitting, because while both have an amazing performance on ASVspoof2019 and 2015, on the other datasets, they lack on specificity and thus accuracy as well. For the Res-TSSDNet some improvements can be visible on the specificity particularly for the FoR-norm dataset, meaning that more bona fide samples are being classified correctly. Therefore, it seems that using this particular data augmentation procedure might help a little in the generalization process. For the TE-ResNet, using more frames, a different loss function and the data augmentation technique seem to not have impacted its performance.

In the third experiment the obtained results can be found in Figures 4.10, 4.11, 4.12 and 4.13. These show that compared to the second experiment, both performances for ResNet18-OC and ECAPA model on all datasets have improved when replacing mixup with OC-Softmax. Moreover, by comparing the results of the ECAPA model to those from the preliminary experiments, it becomes apparent that applying device IRs in conjunction with compression techniques has further enhanced its performance on all datasets, given the higher specificity and sensitivity observed, e.g, in the For-

norm dataset (Figures 4.4, for the preliminary results, and 4.12). For the ResNet-OC model, this new augmented pipeline seems to have stabilized a little its performance when compared to the preliminary results. Although no significant improvements were recorded for the ASVspoof2019 and 2015 datasets, an analysis of both specificity and sensitivity metrics in Figures 4.4 versus 4.12, suggests an enhanced generalization capability as evidenced by the median values. Regarding the remaining models, there were minimal or no improvements observed across all datasets, preventing them from reaching the performance levels of the ECAPA and ResNet18 models, as clearly demonstrated by the figures related to the third experiment. Also, preferring OC-Softmax over mixup did not result in significant changes for the TE-ResNet model. Another key observation is that the augmentation procedures did not result in a significant increase in SSD PT-PT accuracy for any model, maintaining an accuracy level of around 50%. This suggests that these processes do not impact language characteristics, indicating that data augmentation alone is not sufficient to achieve a language-agnostic model. Consequently, fine-tuning the model for the specific language is necessary.

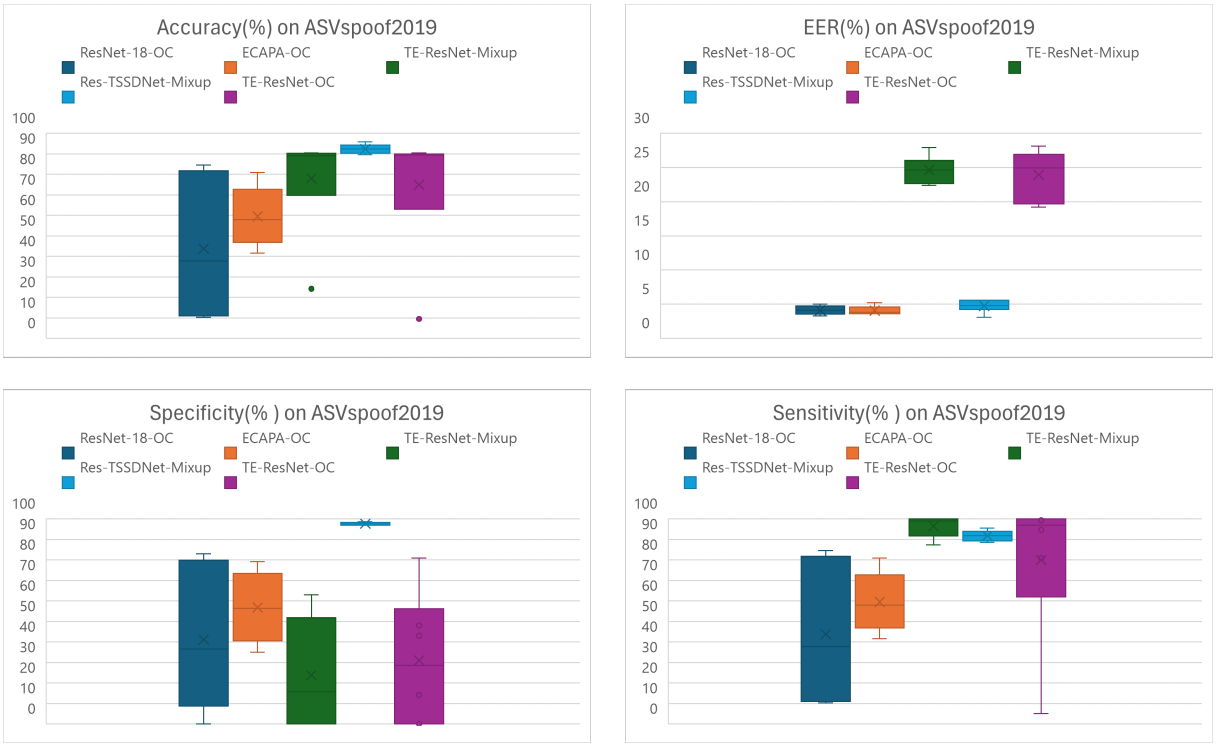

Figure 4.10: Third experiment - ASVspoof2019 LA evaluation set

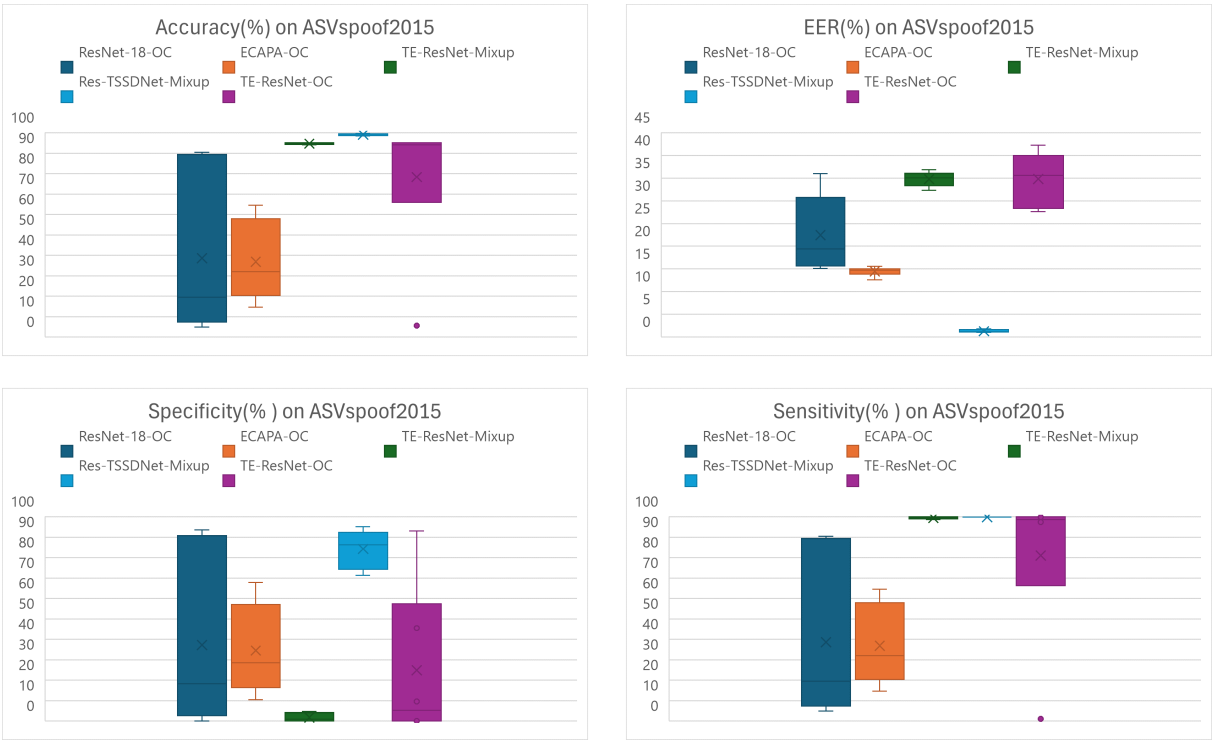

Figure 4.11: Third experiment - ASVspoof2015 evaluation set

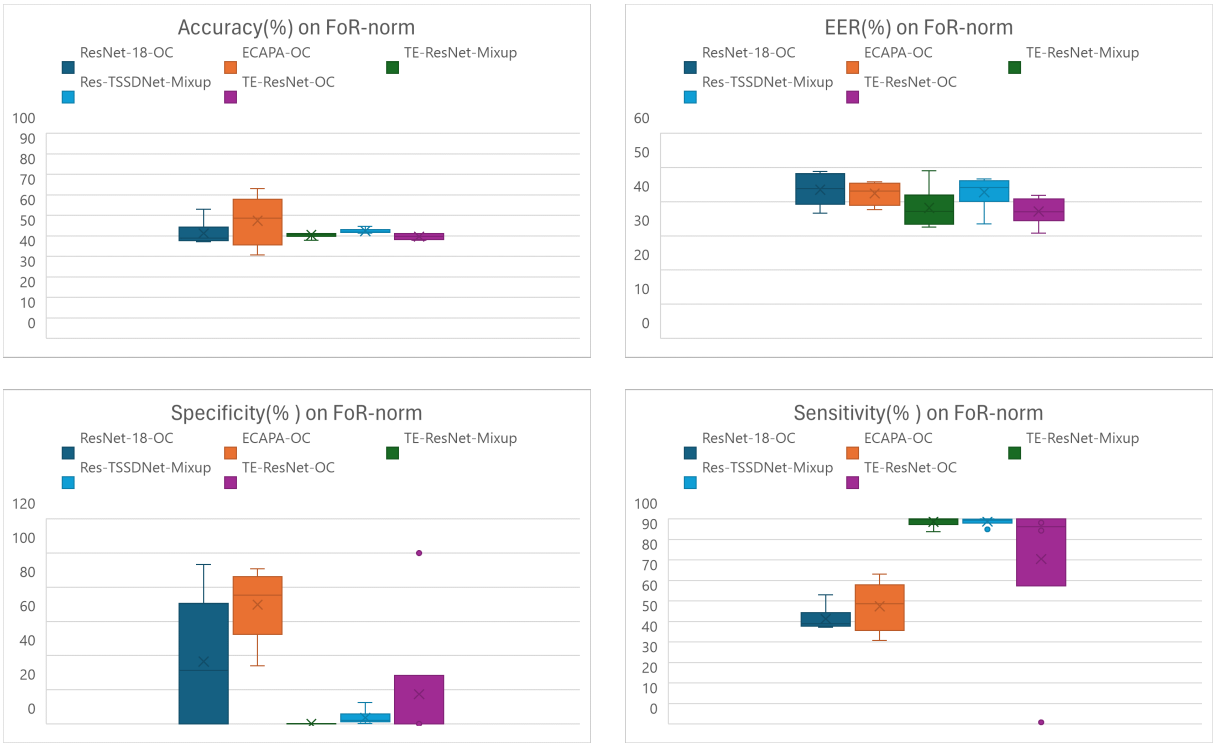

Figure 4.12: Third experiment - FoR-norm evaluation set

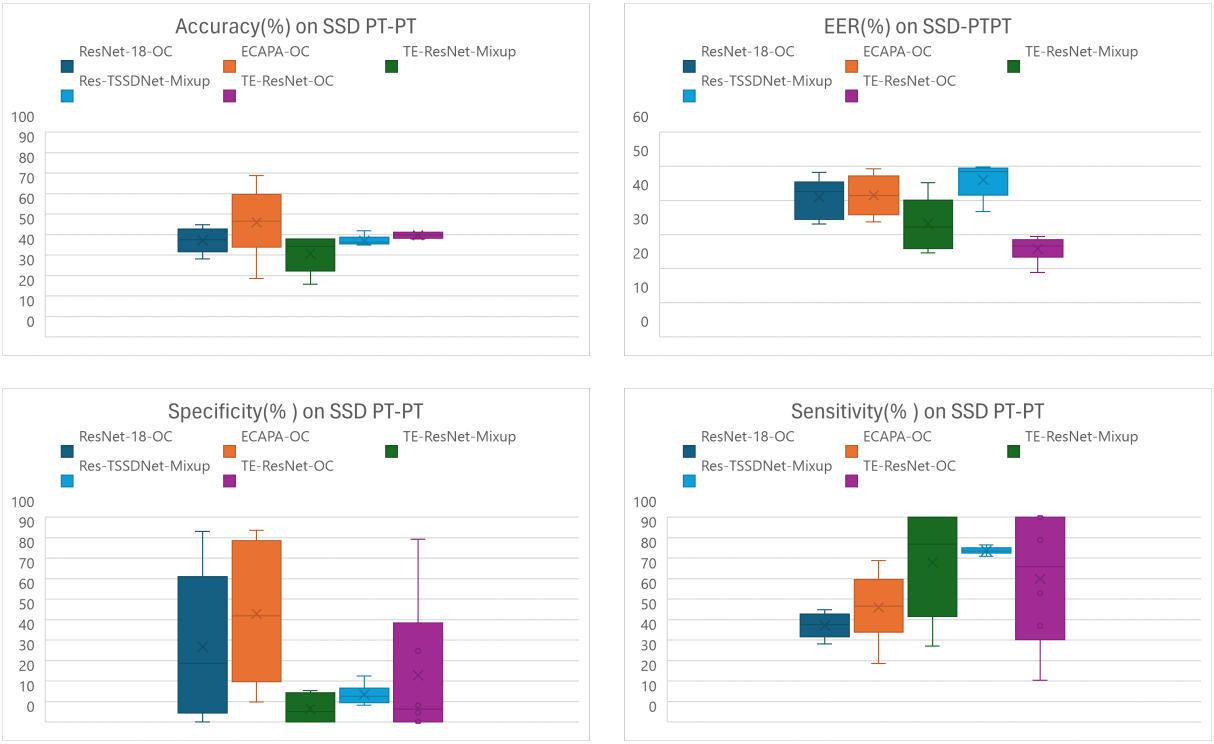

Figure 4.13: Third experiment - SSD PT-PT evaluation set

Before advancing to the fine-tuning stage, it is essential to determine which classifier performed best in the third experiment since this phase utilized the most effective techniques observed. For that we relied on the scheme illustrated in the previous Figure 3.3.

The p-values for Shapiro-Wilk test can be found in Tables 4.7 and 4.8. For specificity, the p-values were higher than 0.05, indicating that we cannot reject the null hypothesis, and thus, the data tested for normality. However, when applying the median version of Levene's test to check for equal variance, the null hypothesis was rejected since 0.0001 < 0.05. For sensitivity, models such as TE-ResNet using mixup and OC-Softmax had their null hypotheses rejected, meaning the data is not normally distributed. Given these conditions, we proceeded with non-parametric tests and used the Friedman test. The results in Table 4.9, show that the null hypothesis was rejected, meaning that differences among the classifiers were found for both metrics. Subsequently, we ranked the classifiers and selected the best performer according to the critical difference diagrams [80]. These diagrams arrange the average ranks of the classifiers on the x axis in order to facilitate performance comparisons between them [80]. The group of classifiers that

could not be deemed as statistically different are linked by a horizontal crossbar [80].

From a specificity perspective, Figure 4.14, it was found that ECAPA ranks significantly higher than TE-ResNet using mixup. On the other hand, from a sensitivity perspective, Figure 4.15, TE-ResNet using mixup ranks significantly higher than ResNet18-OC. However, this latter conclusion is somewhat misleading. Analysis of the sensitivity and specificity box plots, particularly in Figures 4.12 and 4.13, reveal that the two models — TE-ResNet and Res-TSSDNet — act as "dumb" classifiers, only detecting spoofing samples when attempting to generalize. Given this fact, we dismissed these models and focused on the ECAPA and ResNet18 models.

Table 4.7: Shapiro-Wilk's test p-values for the specificity metric

ResNet18-OC	**ECAPA-TDNN-OC**	**TE-ResNet-mixup**	**ResTSSDNet-mixup**	**TE-ResNet-OC**
0.4957	0.0971	0.2489	0.1915	0.6004

Table 4.8: Shapiro-Wilk's test p-values for the sensitivity metric

ResNet18-OC	**ECAPA-TDNN-OC**	**TE-ResNet-mixup**	**ResTSSDNet-mixup**	**TE-ResNet-OC**
0.2195	0.3826	0.0111	0.5425	0.0117

Table 4.9: Friedman's test p-values for both metrics

Specificity	**Sensitivity**
0.0056	0.0087

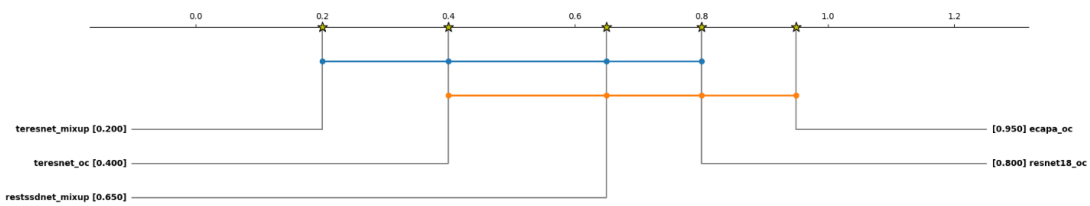

Figure 4.14: Critical difference diagram of average ranks in percentile for specificity metric

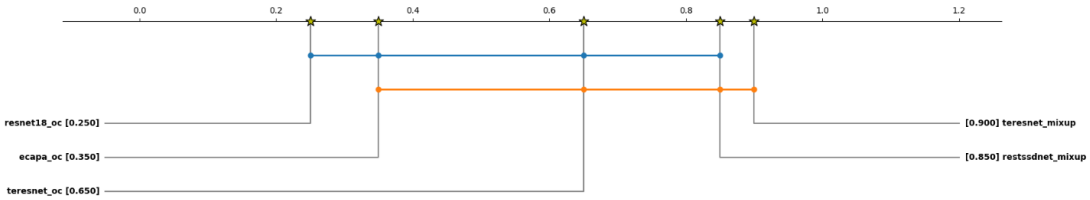

Figure 4.15: Critical difference diagram of average ranks in percentile for sensitivity metric

Next, we fine-tuned a model chosen from both classifiers. This selection was made according to the best average specificity and sensitivity metrics on the three English datasets only (Figures 4.16 and 4.17), because the prior performance on the Portuguese dataset would be improved in any case. So, for the ResNet-OC we selected the model from the first run while for the ECAPA, we chose the model from the second run.

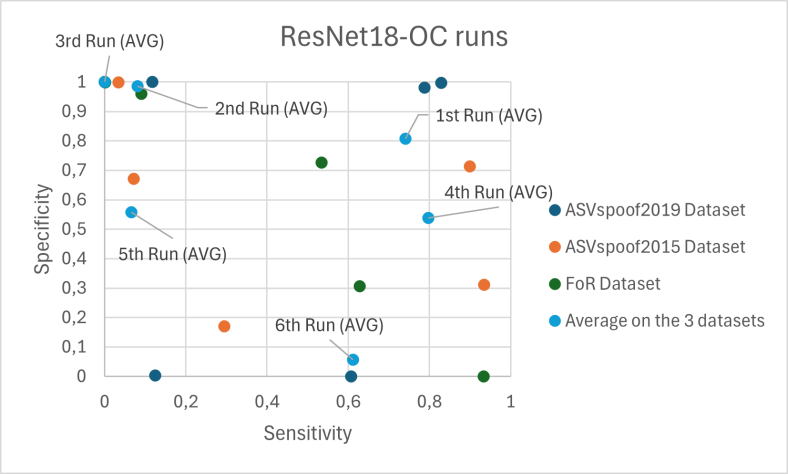

Figure 4.16: ResNet18-OC models

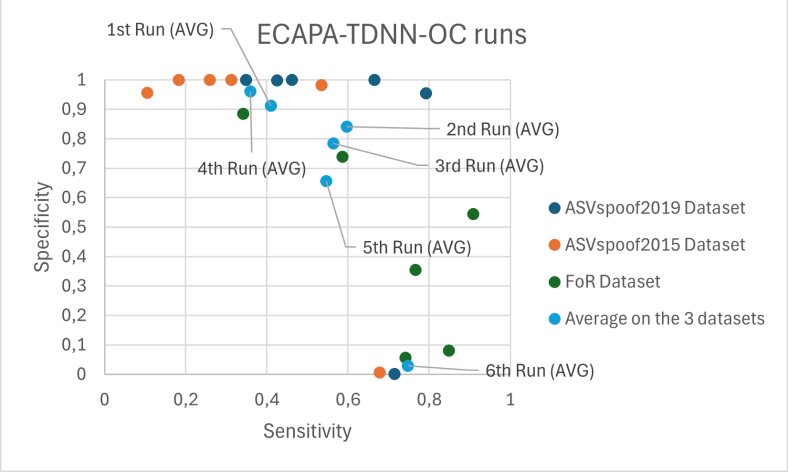

Figure 4.17: ECAPA-OC models

After conducting the fine-tuning process, as detailed in the previous Subsection 3.1.4, we selected the best model from the 2 epochs. The results are presented in Tables 4.10, 4.11, 4.12, and 4.13. These tables display the metrics for the models both before and after fine-tuning (ft), and evaluated on the split PT-PT test set (test) as well as the entire dataset without splitting (all).

Table 4.10: Accuracy of the fine-tuned SSD

Subset	**Evaluation datasets**				
	ASV2019 LA	**ASV2015**	**FoR**	**SSD PT-PT (all)**	**SSD PT-PT (test)**
ECAPA	41.6%	55.6%	73.1%	28.5%	24.2%
ft-ECAPA	72.6%	73.1%	74%	90.9%	79.7%
ResNet18	80.8%	89.1%	62.9%	42.7%	42.9%
ft-ResNet18	75.1%	55.7%	62.8%	89.6%	77.2%

Table 4.11: Specificity of the fine-tuned SSD

Subset	**Evaluation datasets**				
	ASV2019 LA	**ASV2015**	**FoR**	**SSD PT-PT (all)**	**SSD PT-PT (test)**
ECAPA	99.9%	98.2%	54.5%	33.7%	32.8%
ft-ECAPA	99.9%	97.8%	56.9%	98.6%	98.6%
ResNet18	98.0%	71.3%	72.7%	35.3%	33.7%
ft-ResNet18	98.5%	96.9%	43.6%	98.3%	98.1%

Table 4.12: Sensitivity of the fine-tuned SSD

Subset	**Evaluation datasets**				
	ASV2019 LA	**ASV2015**	**FoR**	**SSD PT-PT (all)**	**SSD PT-PT (test)**
ECAPA	34.9%	53.4%	90.8%	22.9%	19.1%
ft-ECAPA	69.5%	71.9%	90.4%	82.5%	68.5%
ResNet18	78.8%	90.0%	53.5%	50.6%	50.0%
ft-ResNet18	72.5%	53.6%	81.1%	92.8%	86.8%

Table 4.13: EER of the fine-tuned SSD

Subset	**Evaluation datasets**				
	ASV2019 LA	**ASV2015**	**FoR**	**SSD PT-PT (all)**	**SSD PT-PT (test)**
ECAPA	3.64%	14.7%	45.8%	46.5%	41.4%
ft-ECAPA	14.5%	19.9%	36.6%	6.4%	9.9%
ResNet18	4.64%	15.1%	40.1%	40.9%	40.1%
ft-ResNet18	17.1%	22%	33%	2.7%	3.5%

Overall, the results in Tables 4.10 to 4.13, for the fine-tuned version of the models suggest that it is feasible to develop a model capable of distinguishing between English and Portuguese language attacks. By decreasing the learning rate from 0.0005 to 0.0001 in the fine-tuning process, the ECAPA appears to enhance the overall accuracy, e.g., on the ASVspoof2015 dataset there has been an increase of 17.5% between the two versions of the same model (Table 4.10), consequently also gaining significant improvements for the specificity and sensitivity, as it is possible to verify in Tables 4.11 and 4.12. On the other hand, the ResNet18 seems to have only improved in terms of accuracy for the PT-PT dataset when compared to the non-fine-tuned version of it (Figure 4.10). This can possibly be attributed to the architecture itself, more specifically, the fact that this model is less complex with less parameters, and consequently with

less capability to learn. Also the EER decreased overall across all datasets (Table 4.13), meaning that both models are making fewer errors while classifying.

4.3 Discussion

Despite encountering some challenges that could have directly impacted the results section, such as the scarcity of datasets in PT-PT — with most available datasets being in English or Brazilian Portuguese — this study overcame those obstacles and uncovered significant findings.

The two best models, ResNet18-OC and ECAPA, delivered good performances on both English and PT-PT language datasets, suggesting the potential for a language-agnostic approach. Further studies are required to test this premise in other languages. Also, the preference between the two fine-tuned models is in favor to the ECAPA which accomplished an accuracy of over 70% in all evaluation datasets.

Another observation has to do with the PT-PT dataset. Given the limited availability of reliable and high-quality TTS and STS services for PT-PT, only four types of attacks could be obtained. To improve the models' performance and prepare for other unseen attacks, it is crucial to acquire more samples in this language.

Regarding the developed model (TE-ResNet), its performance did not meet our expectations. Despite various attempts to enhance its architecture, including adding more features, experimenting with different loss functions, and employing data augmentation techniques, the model still fell short. This suggests that further optimizations might be necessary to match the performance of other models. Moreover, based on our findings, replacing the ResNet34 blocks with ResNet18 blocks could potentially enhance the model's performance.

5

Conclusions and future work

In this thesis, we addressed the main problem of developing a synthetic speech detector for the PT-PT and English language. By employing state-of-the-art models such as the ECAPA-TDNN and by developing a promising model (TE-ResNet), we were able to gather valuable insights regarding the optimal combination of parameters to achieve the best performance. Also, some important milestones were achieved. Through the use of the acoustic simulator, we could apply compression codecs and IRs devices to improve the robustness over channel variability, mitigating this way the performance degradation and therefore reaching a good performance across all English datasets. However, applying this same data augmentation technique didn't seem to improve the performance in the SSD PT-PT dataset with accuracy remaining around 50%. Therefore, we incorporated the fine-tunning process for both ECAPA-TDNN and ResNet18-

OC, to be able to significantly improve performance for the SSD PT-PT dataset, achieving and maintaining an accuracy of over 70% for the English datasets. In contrast to this, the developed model (TE-ResNet) fell short of its performance, indicating a need for further optimization.

In the future, to fully test the language-agnosticism hypothesis, it is necessary to explore datasets in other languages for the task of SSD. Also, efforts will concentrate on feature selection and hyperparameter tuning to further enhance the models' performance. Additionally, a recent study [81], suggests that relying solely on ReLU may not yield the best results. Therefore, plans for the next phase of research include modifying these classifiers to incorporate multiple activation functions. Also, continuous learning approaches have been gaining popularity [82], therefore experimenting with these new frameworks might also help to improve our results.

Ultimately, our major goal for the future is to implement the fraud detection model detailed in Section 2.8, to create a comprehensive and robust framework capable of defending against various fraudulent attempts, thereby enhancing user safety.

References

[1] B. News, "Fake Obama created using AI tool to make phoney speeches." https://www.bbc.com/news/av/technology-40598465, 2017. Accessed on 24/11/2023.

[2] Y. Leviathan and Y. Matias, "Google duplex: An ai system for accomplishing real-world tasks over the phone," 2018.

[3] MATT, "THE FUTURE OF VOICE SYNTHESIS WITH AI." https://autogpt.net/the-future-of-voice-synthesis-with-ai/. Accessed on 18/01/2024.

[4] A. Organisers. https://www.asvspoof.org/. Accessed on 09/01/2024.

[5] G. Hua, A. B. J. Teoh, and H. Zhang, "Towards end-to-end synthetic speech detection," *IEEE Signal Processing Letters*, vol. 28, pp. 1265–1269, 2021.

[6] J. Center for Language Speech Processng (CLSP), "Audio Deepfake Detection: An Overview and its Challenges – Zhizheng Wu (Facebook)." https://www.youtube.com/watch?v=8MNmP5M4q10. Accessed on 10/05/2024.

[7] B. Sisman, J. Yamagishi, S. King, and H. Li, "An overview of voice conversion and its challenges: From statistical modeling to deep learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 132–157, 2020.

[8] Y. Zhang, G. Zhu, F. Jiang, and Z. Duan, "An empirical study on channel effects for synthetic voice spoofing countermeasure systems," *arXiv preprint arXiv:2104.01320*, 2021.

[9] X. Chen, Y. Zhang, G. Zhu, and Z. Duan, "Ur channel-robust synthetic speech detection system for asvspoof 2021," *arXiv preprint arXiv:2107.12018*, 2021.

[10] Z. Zhang, X. Yi, and X. Zhao, "Fake speech detection using residual network with transformer encoder," in *Proceedings of the 2021 ACM workshop on information hiding and multimedia security*, pp. 13–22, 2021.

[11] R. Zhong, X. Dong, R. Lin, and H. Zou, "An incremental identification method for fraud phone calls based on broad learning system," in *2019 IEEE 19th International Conference on Communication Technology (ICCT)*, pp. 1306–1310, IEEE, 2019.

[12] F. Pacheco, J. V. de Oliveira, R.-V. Sánchez, M. Cerrada, D. Cabrera, C. Li, G. Zurita, and M. Artés, "A statistical comparison of neuroclassifiers and feature selection methods for gearbox fault diagnosis under realistic conditions," *Neurocomputing*, vol. 194, pp. 192–206, 2016.

[13] J. Ebner, "Sklearn confusion_matrix, Explained." https://www.sharpsightlabs.com/blog/sklearn-confusion_matrix-explained/, 2023. Accessed on 29/09/2024.

[14] H. Delgado, N. Evans, T. Kinnunen, K. A. Lee, X. Liu, A. Nautsch, J. Patino, M. Sahidullah, M. Todisco, X. Wang, *et al.*, "Asvspoof 2021: Automatic speaker verification spoofing and countermeasures challenge evaluation plan," *arXiv preprint arXiv:2109.00535*, 2021.

[15] A. Lieto, D. Moro, F. Devoti, C. Parera, V. Lipari, P. Bestagini, and S. Tubaro, "Hello? who am i talking to?" a shallow cnn approach for human vs. bot speech classification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2577–2581, IEEE, 2019.

[16] X. Li, N. Li, C. Weng, X. Liu, D. Su, D. Yu, and H. Meng, "Replay and synthetic speech detection with res2net architecture," in *ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 6354–6358, IEEE, 2021.

[17] L. Zhou, J. Gao, D. Li, and H.-Y. Shum, "The design and implementation of xiaoice, an empathetic social chatbot," *Computational Linguistics*, vol. 46, no. 1, pp. 53–93, 2020.

[18] K. Houser, "Google's Call-Making AI Has Some Competition From Microsoft." https://futurism.com/xiaoice-microsoft-google-duplex. Accessed on 10/01/2024.

[19] H. Gao, H. Liu, D. Yao, X. Liu, and U. Aickelin, "An audio captcha to distinguish humans from computers," in *2010 Third International Symposium on Electronic Commerce and Security*, pp. 265–269, IEEE, 2010.

[20] H. Tu, A. Doupé, Z. Zhao, and G.-J. Ahn, "Sok: Everyone hates robocalls: A survey of techniques against telephone spam," in *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 320–338, IEEE, 2016.

[21] D. Gritzalis, Y. Soupionis, V. Katos, I. Psaroudakis, P. Katsaros, and A. Mentis, "The sphinx enigma in critical voip infrastructures: Human or botnet?," in *IISA 2013*, pp. 1–6, IEEE, 2013.

[22] R. K. Das, X. Tian, T. Kinnunen, and H. Li, "The attacker's perspective on automatic speaker verification: An overview," *arXiv preprint arXiv:2004.08849*, 2020.

[23] C. L. Information, "Senate Bill No. 1001." https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180SB1001#, 2018. Accessed on 24/11/2023.

[24] M. R. Kamble, H. B. Sailor, H. A. Patil, and H. Li, "Advances in anti-spoofing: from the perspective of asvspoof challenges," *APSIPA Transactions on Signal and Information Processing*, vol. 9, p. e2, 2020.

[25] Wikipedia contributors, "Cepstral mean and variance normalization — Wikipedia, the free encyclopedia," 2019. Accessed on 11/01/2024.

[26] T. M. Inc. https://www.mathworks.com/help/audio/ref/mfcc.html. Accessed on 16/01/2023.

[27] Z. K. Abdul and A. K. Al-Talabani, "Mel frequency cepstral coefficient and its applications: A review," *IEEE Access*, 2022.

[28] X. Wang, Y. Xiao, and X. Zhu, "Feature selection based on cqccs for automatic speaker verification spoofing.," in *Interspeech*, pp. 32–36, 2017.

[29] T. M. Inc. https://www.mathworks.com/discovery/feature-extraction.html. Accessed on 16/01/2023.

[30] G. Lavrentyeva, S. Novoselov, A. Tseren, M. Volkova, A. Gorlanov, and A. Kozlov, "Stc antispoofing systems for the asvspoof2019 challenge," *arXiv preprint arXiv:1904.05576*, 2019.

[31] G. Chu, J. Wang, Q. Qi, H. Sun, S. Tao, H. Yang, J. Liao, and Z. Han, "Exploiting spatial-temporal behavior patterns for fraud detection in telecom networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 6, pp. 4564–4577, 2022.

[32] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: A survey," *speech communication*, vol. 66, pp. 130–153, 2015.

[33] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, and A. Sizov, "Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," in *Sixteenth annual conference of the international speech communication association*, 2015.

[34] J. Yi, C. Wang, J. Tao, X. Zhang, C. Y. Zhang, and Y. Zhao, "Audio deepfake detection: A survey," *arXiv preprint arXiv:2308.14970*, 2023.

[35] P. Taylor, *Text-to-speech synthesis*. Cambridge university press, 2009.

[36] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, *et al.*, "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[37] Z. Wu, Z. Xie, and S. King, "The blizzard challenge 2019," in *Proc. Blizzard Challenge Workshop*, vol. 2019, 2019.

[38] S. H. Mohammadi and A. Kain, "An overview of voice conversion systems," *Speech Communication*, vol. 88, pp. 65–82, 2017.

[39] F. Fang, J. Yamagishi, I. Echizen, and J. Lorenzo-Trueba, "High-quality nonparallel voice conversion based on cycle-consistent adversarial network," in *2018 IEEE International conference on acoustics, speech and signal processing (ICASSP)*, pp. 5279–5283, IEEE, 2018.

[40] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

[41] K. Tanaka, H. Kameoka, T. Kaneko, and N. Hojo, "Atts2s-vc: Sequence-to-sequence voice conversion with attention and context preservation mechanisms," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6805–6809, IEEE, 2019.

[42] T. Kaneko, H. Kameoka, K. Hiramatsu, and K. Kashino, "Sequence-to-sequence voice conversion with similarity metric learned using generative adversarial networks.," in *Interspeech*, vol. 2017, pp. 1283–1287, 2017.

[43] "Voice Conversion Challenge." https://www.vc-challenge.org/. Accessed on 13/05/2024.

[44] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee, *et al.*, "Asvspoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Computer Speech & Language*, vol. 64, p. 101114, 2020.

[45] Y. University, "The Fake-or-Real Dataset." https://bil.eecs.yorku.ca/datasets/. Accessed on 28/11/2023.

[46] VoxForge, "Portuguese Speech Files." https://www.voxforge.org/home/downloads/speech/portuguese-speech-files. Accessed on 06/05/2024.

[47] C. Voice, "Conjuntos de dados." https://commonvoice.mozilla.org/pt/datasets. Accessed on 06/05/2024.

[48] H. Face, "Dataset Card for MultiLingual LibriSpeech." https://huggingface.co/datasets/multilingual_librispeech. Accessed on 06/05/2024.

[49] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5206–5210, IEEE, 2015.

[50] LibriVox, "Free public domain audiobooks." https://librivox.org/. Accessed on 06/05/2024.

[51] P. Gutenberg, "Welcome to Project Gutenberg." https://www.gutenberg.org/. Accessed on 06/05/2024.

[52] J. Iranzo-Sánchez, J. A. Silvestre-Cerdà, J. Jorge, N. Roselló, A. Giménez, A. Sanchis, J. Civera, and A. Juan, "Europarl-st: A multilingual corpus for speech translation of parliamentary debates," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8229–8233, 2020.

[53] C. Wang, M. Riviere, A. Lee, A. Wu, C. Talnikar, D. Haziza, M. Williamson, J. Pino, and E. Dupoux, "VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, (Online), pp. 993–1003, Association for Computational Linguistics, Aug. 2021.

[54] Anjok07, "Ultimate Vocal Remover GUI v5.6." https://github.com/Anjok07/ultimatevocalremovergui. Accessed on 07/05/2024.

[55] R. Reimao and V. Tzerpos, "For: A dataset for synthetic speech detection," in *2019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, pp. 1–10, IEEE, 2019.

[56] "Retrieval-based-Voice-Conversion-WebUI." https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI. Accessed on 28/04/2024.

[57] Dream-High, "RMVPE." https://github.com/Dream-High/RMVPE. Accessed on 08/05/2024.

[58] llElevenLabs, "Generative Voice AI." https://elevenlabs.io/. Accessed on 07/05/2024.

[59] G. Cloud, "Text-To-Speech." https://cloud.google.com/text-to-speech. Accessed on 28/04/2024.

[60] A. Prognosticator, "Text-to-Speech: The Difference Between Standard and WaveNet Google Voices." https://aiprognosticator.substack.com/p/tts-the-difference-between-standard-andwavenet. Accessed on 08/05/2024.

[61] G. Cloud, "Tipos de vozes." https://cloud.google.com/text-to-speech/docs/voice-types?hl=pt-br. Accessed on 08/05/2024.

[62] jcoutsousa, "voice-clone." https://github.com/jcoutsousa/voice-clone/blob/main/voiceclone.py. Accessed on 08/05/2024.

[63] moonlight_18, "Ariana Grande AI." https://www.weights.gg/pt/models/clm733dx31isqcctci0ynb5sw. Accessed on 09/05/2024.

[64] Mantrax, "Rincón de Varo." https://www.weights.gg/pt/models/clufxwly20vgw118w9n3ru05f. Accessed on 09/05/2024.

[65] L. II, "Mordecai." https://www.weights.gg/pt/models/cltdbc4ch08ip856z0aes8lrt. Accessed on 09/05/2024.

[66] M. Ferras, S. Madikeri, P. Motlicek, S. Dey, and H. Bourlard, "A large-scale open-source acoustic simulator for speaker recognition," *IEEE Signal Processing Letters*, vol. 23, no. 4, pp. 527–531, 2016.

[67] Y. Zhang, F. Jiang, and Z. Duan, "One-class learning towards synthetic voice spoofing detection," *IEEE Signal Processing Letters*, vol. 28, pp. 937–941, 2021.

[68] hkproj, "pytorch-transformer." https://github.com/hkproj/pytorch-transformer/tree/main. Accessed on 24/11/2023.

[69] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[70] N. Brümmer and E. De Villiers, "The bosaris toolkit: Theory, algorithms and code for surviving the new dcf," *arXiv preprint arXiv:1304.2865*, 2013.

[71] P. Yadav, "Beware! Elon Musk deepfake livestream scams thousands with fake crypto scheme." https://www.cnbctv18.com/technology/elon-musk-deepfake-youtube-live-scam-warning-cryptocurrency-online-fraud-19465 htm. [Accessed 29/08/2024].

[72] yzyouzhang, "Empirical-Channel-CM." https://github.com/yzyouzhang/Empirical-Channel-CM?tab=readme-ov-file. Accessed on 24/11/2023.

[73] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine learning research*, vol. 7, pp. 1–30, 2006.

[74] S. Narkhede, "Understanding Confusion Matrix." https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62, 2018. Accessed on 16/01/2024.

[75] "Equal Error Rate." https://www.sciencedirect.com/topics/engineering/equal-error-rate. Accessed on 26/08/2024.

[76] "Calculate EER from FAR and FRR?." https://stats.stackexchange.com/questions/221562/calculate-eer-from-far-and-frr. Accessed on 26/08/2024.

[77] Wikipedia contributors, "Sensitivity and specificity." Accessed on 26/08/2024.

[78] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[79] IBM, "What is overfitting?." https://www.ibm.com/topics/overfitting. Accessed on 16/01/2024.

[80] scikit posthocs, "Tutorial." https://scikit-posthocs.readthedocs.io/en/latest/tutorial.html. Accessed on 29/09/2024.

[81] W. H. Kang, J. Alam, and A. Fathan, "Investigation on activation functions for robust end-to-end spoofing attack detection system," *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, pp. 83–88, 2021.

[82] X. Zhang, J. Yi, and J. Tao, "Evda: Evolving deepfake audio detection continual learning benchmark," *arXiv preprint arXiv:2405.08596*, 2024.