POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Exploring Signal Purification against Adversarial Attacks for Speech Deepfake Detection

Tesi di Laurea Magistrale in
Music and Acoustic Engineering

Alfredo Brusca, 10936149

Abstract: Recent advances in deep learning and generative systems have enabled the creation of increasingly realistic synthetic media. As these technologies become more accessible, their misuse poses significant risks, making the ability to distinguish between real and synthetic content paramount. In the audio domain, speech deepfake generation techniques allow the synthesis of signals, which can mimic the voice of a target speaker and make them say arbitrary sentences. Speech deepfake detection techniques focus on identifying fake signals by identifying subtle artifacts typically absent in genuine human speech. However, this is an ongoing challenge, as detection systems must continually adapt to evolving threats. With detection methods now widely deployed, malicious users have begun crafting adversarial attacks explicitly designed to bypass even the most advanced methods. By subtly introducing nearly imperceptible noise into speech samples, these attacks can cause detectors to fail the detection process and misclassify synthetic speech as genuine or vice versa. To counter the use of adversarial attacks, defense mechanisms have been presented in the literature. These aim to process the input speech signals to mitigate the malicious effect caused by the adversarial attacks. In this thesis, we contribute to this research field and propose multiple adversarial defense strategies. Our approach builds incrementally, starting with simple model re-training, progressing to the use of state-of-the-art speech enhancement models, and finally adapting an input refactoring technique originally developed in the image forensics domain. The analysis of these techniques offers valuable insights, demonstrating the potential to increase the classification accuracy of the considered detectors up to 45% in balanced accuracy.

Advisor:
Prof. Paolo Bestagini

Co-advisors:
Viola Negroni
Daniele Ugo Leonzio
Davide Salvi

Academic year:
2023-2024

Key-words: Speech Deepfake Detection, Adversarial Attacks, Signal Purification

1. Introduction

We are living in an era marked by unprecedented technological advancements, with one of the most transformative innovations being in the field of Machine Learning (ML) and Artificial Intelligence (AI). These technologies have revolutionized computer science by shifting from traditional, rule-based programming to dynamic algorithms capable of learning and making data-driven decisions autonomously. Unlike standard software that follows pre-defined instructions, ML enables systems to analyze and derive patterns from input data, allowing them to classify objects, differentiate between data segments, and even generate novel content.

The advent of Deep Learning (DL) has propelled these capabilities forward. DL introduces sophisticated layers of feature extraction, mathematical modeling, and decision-making processes that empower machines to identify and prioritize relevant features without human intervention. This paradigm shift has been particularly impactful for generative algorithms, as the machine's ability to grasp complex, nuanced features of data enables it to create realistic and contextually appropriate content. Together, ML, AI, and DL, paved the way for increasingly intelligent and creative applications. The generation of synthetic content explored fields such as image generation, image style transfer, music generation and speech synthesis. Synthetic speech generation has in fact reached a level of quality that is often indistinguishable from real human recordings [22, 34, 47, 53]. This technology offers exciting advancements for companies and common citizens, such as accessibility tools for people with speech disorders [62], dubbing and artistic implementations and augmented reality applications [59]. However, these tools can also be used in harmful ways to attack security systems, individuals and companies. One example of malicious application of synthetic speech generation are speech deepfakes. Speech deepfakes are synthetic speech signals created through generative DL technology. They can be used to mimic vocal traits of the user in the generation of novel content, using them to create speech signals where arbitrary sentences are uttered. These tools are able to gather linguistic information and the content to be produced from an input text or by converting a pre-existing speech signal and to capture speech patterns, pitch, speech rate and accent of a specific speaker. This technology raises serious concerns, especially in security systems that rely on voice authentication, as well as in the realms of personal reputation and misinformation, where it could convincingly falsify evidence, spread misinformation, harm public figures and politicians, and even be used as propaganda tools from authoritarian regimes [26]. Research is already being conducted on the efficacy of the malicious application of these tools [52], which, while a worst-case scenario, could be exploited to influence elections in various countries and have already seen a significant rise in prominence ahead of major global political events.

Today, the challenge of Audio Deepfake Detection (ADD) is actively addressed by the MultiMedia Forensics (MMF) community [2, 38], with research progressing daily.

ADD focuses on the detection of deepfakes through the use of ML and DL models called Synthetic Speech Detectors (SSD). While speech deepfakes can be nearly indistinguishable from genuine speech to the human ear, they present several irregularities and artifacts that can be exploited to perform detection. The existence and continuous development of new algorithms for synthetic speech generation proved to be a threat to the robustness of classifiers with respect to unseen attacks, but through several strategies and optimization in recent years some of SSD achieved great performances. One of the most impactful initiatives that fostered the development of robust techniques for SSD has been the ASVSpoof challenge, now in its fifth iteration since its inception in 2015.

Most state-of-the-art SSDs have, however, been proven to be susceptible to adversarial attacks [55], i.e., techniques that are able to mislead a detector into incorrect classifications by adding an imperceptible noise to audio tracks. These attacks can exploit knowledge of the internal workings of the SSD to craft perturbations, in which case they are called white-box, or they can be unaware of the model to be attacked and its mechanisms and rely on other mechanisms such as masking the effect of speech synthesis, in which case they are called black-box.

White-box attacks, particularly in light of the recent spread of widely available pre-trained models, pose an elevated threat to current recognition systems, underscoring the pressing need for robust defenses [13].

In order to counter the effects of adversarial attacks, the literature has proposed several defense

mechanisms called adversarial defense. Examples of widely used adversarial defenses techniques are: adversarial detection, which tries to understand whether a sample has been tampered with through adversarial attacks; adversarial training, which aims at creating models that are natively robust against adversarial attacks; adversarial purification, which uses pre-processing techniques on the samples before entering into the model to counter the effect of any possible adversarial attack.
Despite the amount of research in the field of adversarial defense, the audio domain is under-explored with respect to image and video. Some effort has been done in transposing successful image and video approaches with varying degree of success, but ADD has seen less focus than other audio domain problems such as Automatic Speaker Verification (ASV).
This study aims to expand research in the ADD adversarial defense task. Model re-training will be analyzed to understand the extent of transferability of white-box attacks, while adversarial purification will be studied by assessing three defense mechanisms: denoising through Facebook AI research's DEMUCS denoiser [11], speech enhancement through the SepFormer model [49] and spectrogram purification through DISCO [17].
This thesis work is structured as follows. Section 2 provides informations on the background of covered topics, introducing the current state-of-the-art in synthetic speech generation, ASV and ADD, adversarial attacks and defense strategies. Section 3 presents the approach we propose, including what recognition models were used, which attacks were implemented and what our defense mechanisms are. Section 4 covers the specific setup used in our study and every parameter used. Section 5 presents and discusses the results of our analysis. Section 6 concludes this thesis, analyzing its contributions and drawing future research directions.

2. Background

This section will present a concise overview of the literature in the ADD field, as well as current state-of-the-art adversarial attacks and defenses in the audio domain.

2.1. Speech Generation Technologies

The synthesis of speech signals, a goal pursued for nearly a century, is now achievable with unprecedented realism thanks to recent advancements in DL technology. These advancements in speech synthesis have introduced new challenges, as synthesized voices can now be used to attack ASV systems, which rely on voice-based authentication to grant access to sensitive data or physical spaces. Furthermore, the misuse of speech synthesis techniques can pose risks to individuals' reputations and public trust, being employed to fabricate the so-called *deepfakes*.
The primary driver behind the success of modern speech synthesis is the development of deep learning-based neural networks specifically tailored to this purpose. The two predominant techniques in deep learning-based voice cloning are Text-to-Speech (TTS) and Voice Conversion (VC), both of which harness neural networks' advanced feature-extraction capabilities to generate highly realistic vocal outputs.
TTS aims to produce speech signals from textual input, enabling systems to generate spoken content based on written language. Over the years, TTS has seen remarkable advancements, beginning with Convolutional Neural Networks (CNNs) such as WaveNet [53]. While WaveNet was a breakthrough, recent TTS research has achieved even greater levels of quality with Variational Autoencoders (VAEs) [21] and Generative Adversarial Networks (GANs) [33]. Most TTS algorithms rely on two components:

- **A Spectrogram Generator**, which converts the input text into a spectrogram, which represents the frequency component of the audio over time.
- **A Vocoder**, which converts the spectrogram into an audio file. Vocoders operate on a source-filter model, representing the vocal tract (mouth, larynx, and vocal cords) as a system of filters applied to a signal.

VC systems differ from TTS ones in that they take an existing speech signal as input and then modify the voice's acoustic properties to resemble those of a target speaker while preserving the

original linguistic content. VC methods typically integrate several key processes, including speech analysis, speaker classification, and vocoding. Modern voice conversion systems often utilize Generative Adversarial Networkss (GANs) to output high-quality speech [54].

2.2. Speech Deepfake Detection

As synthetic speech generation becomes increasingly realistic, distinguishing genuine audio from fake is crucial to tackle its malicious use. Although high-quality synthesized speech can sound indistinguishable to the human ear, specific features and subtle and often imperceptible artifacts can be identified by specific ML and AI models tailored to detect deepfake audio. This task falls within classification problems, which can be addressed with both traditional machine learning techniques and advanced deep learning models.

Traditional ML models, such as Support Vector Machiness (SVMs), have demonstrated effectiveness in classifying known distributions of audio data. SVMs work by representing audio features as points in an n-dimensional space and identifying the optimal hyperplane that separates classes based on training data. However, while effective for data with consistent patterns, SVMs face challenges with new types of attacks and unfamiliar data distributions. Other traditional ML techniques that have been explored in the literature include the extraction of specific features from the audio samples that represent speech as an auto-regressive process, which are then used to perform the classification through the analysis of Short-Term and Long-Term (STLT) prediction traces, along with bicoherence-based features [7].

DL models have shown greater adaptability in distinguishing between genuine and spoofed samples, even with complex or nuanced datasets [27, 30, 39–44, 61]. Several SSDs have been proposed using DL methods, making use of techniques such as acoustic and semantic feature analysis [5, 36, 56]. The most popular architecture among these DL models is the CNN-based classifier [9], and the most successful is the Light Convolutional Neural Network (LCNN) proposed by Wu et al. in 2020 [60]. LCNN has achieved success in ASVspoof [24] and ADD [63] competitions, demonstrating its robustness in various ADD scenarios.

Other notable models in this area include RawNet [50, 57], ResNet [1, 9] and wav2vec [3, 46]. ResNet employs residual learning, where each layer learns differences (residuals) relative to its input. This approach improves gradient flow in deeper networks, allowing ResNet to achieve high accuracy without the computational demands of very deep CNNs, such as VGG networks [16]. RawNet, on the other hand, is an end-to-end deep CNN model that directly analyzes raw audio waveforms. This approach enables RawNet to identify relevant frequency regions adaptively, unlike fixed-feature models that rely on predefined spectral features [57]. Wav2vec is an unsupervised pre-training model originally designed to enhance the speech recognition task. Some SSDs [8, 51] used it to extract relevant features from audio samples. These features were fine-tuned on domain-specific training data to enhance performance on unseen spoofing attacks, which was shown to improves the performance by a relative 58.25% compared to traditional acoustic features extracted by digital signal processing front-ends [58].

Despite their effectiveness, these models face significant challenges, particularly due to the unpredictability of synthetic speech algorithms encountered in real-world applications. When faced with previously unseen distributions or noisy audio, classifiers may mislabel samples. Additionally, deep learning models can lack interpretability, making it difficult to understand precisely what features the model uses to classify audio and how reliable the classifier may be outside controlled settings. These vulnerabilities create opportunities for adversaries to craft audio samples specifically designed to evade detection, as will be discussed in the following section.

2.3. Adversarial Attacks and Defenses

While synthetic speech detectors can achieve high levels of accuracy, they also exhibit vulnerabilities that so-called *adversarial attacks* can exploit in order to bypass detection. Adversarial attacks aim to subtly modify audio samples in ways that are imperceptible to the human ear but alter the classification outcome of a detection model. These attacks exploit the latent space, an internal representation of hidden features created by deep learning-based detectors, that is used to make

classification decisions. The goal of adversarial attacks is to alter the model's classification without triggering human suspicion by targeting these internal representations.

In deep learning models, each classification decision relies on weighted hidden features, which the model uses as inputs to a mathematical function that estimates the probability of a sample belonging to each class. Conceptually, this can be visualized as a refined hyperplane-based decision process, where a sample's position in latent space determines its classification. If attackers can map this latent space, they can identify the minimal alteration needed to move a sample into a different classification region. By leveraging the model's gradients, the directional values that indicate how the model's weights should change to improve classification accuracy, attackers can determine precisely how to perturb a sample to alter its classification. Adding noise in the direction opposite to the gradient shifts the sample's latent position toward a region of erroneous classification.

White-box attacks assume that attackers have full access to the model's architecture and parameters, including its gradient information. Two traditional white-box adversarial attacks are the Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD) [25]. Both FGSM and PGD were initially used as adversarial attacks against AI models concerning images, but they have since been successfully transferred in the audio domain. They function by adding gradient-based noise to the sample, but they differ in complexity and effectiveness:

FGSM is a simpler attack whose only parameter is the amount of noise (ϵ) added to the track; The simplicity of FGSM makes it efficient, but higher ϵ values, which increase attack success, also raise the likelihood of detection.

PGD iteratively applies small perturbations to the input, allowing for finer control of both noise level and success rate. By adjusting the number of iterations and ϵ. Attacks with a higher ϵ, PGD can achieve high accuracy degradation with minimal perceptible noise. PGD has demonstrated effectiveness in reducing model accuracy to 0%, while maintaining a high Signal-to-Noise-Ratio (SNR) of around 32.77, making it difficult for listeners to discern the added noise [10].

In contrast to white-box attacks, black-box attacks operate without detailed knowledge of the target model's parameters or structure. Black-box attacks generally rely on trial-and-error or external optimization techniques to achieve adversarial goals. Recent black-box attack methodologies specifically focus on masking the effects of deepfake or spoofed audio to evade detection systems. One notable example is Malafide, an adversarial attack designed to compromise synthetic speech detectors in both white-box and black-box scenarios [32]. Malafide employs a linear time-invariant filter optimized to misclassify fake samples as authentic. By simulating specific spoofing conditions, Malafide has been shown to bypass several robust countermeasures and even compromise ASV systems.

Indeed, the majority of research in this area focuses on ASV, while the field of deepfake audio detection remains less explored. Many adversarial techniques and defenses for ADD borrow from ASV literature, but deepfake detection presents unique challenges. ASV systems are typically designed for machine verification, which can tolerate distortions to some extent, whereas deepfake audio must be convincing to both automated systems and human listeners. Several studies have contributed to this area [20, 28, 35, 64], some of which concentrate on countering adversarial attacks. The process of preventing misclassification due to adversarial interference, known as adversarial defense, can be pursued through multiple approaches.

One approach focuses on detecting when a sample has been modified by an adversarial attack, commonly known as adversarial detection. This method seeks to identify subtle artifacts left by adversarial perturbations without affecting clean samples. For example, Kwon et al. [23] achieved a 97% detection rate by applying carefully crafted distortions that significantly impact adversarially modified samples while leaving genuine samples unaffected. By analyzing the distinctive distortion patterns of adversarial attacks, this approach improves the ability to identify tampered data, thereby enhancing detection robustness without direct interference with the model's primary classification process.

Adversarial training is widely regarded as a promising approach, with strong support in recent research as a way to increase model robustness against adversarial attacks [2]. In adversarial training, models are trained using a diverse array of adversarially perturbed samples, which enables them to learn to resist such attacks. By adjusting objective functions and hyperparameters, this method aims to improve the model's generalizability in scenarios involving adversarial interference. However, adversarial training is computationally intensive and can lead to issues such as

reduced accuracy and convergence difficulties [4]. Recent studies in audio applications underscore the effectiveness of adversarial training. For example, Pal et al. [31] introduced a hybrid adversarial training technique for deep speaker recognition systems, which were exposed to a variety of adversarial perturbations, including well-known attacks such as FGSM and PGD attacks. This approach showed enhanced resilience, particularly against complex attacks like PGD, albeit with some limitations in convergence and scalability. Joshi et al. [18] investigated a new adversarial training method that relies on the fine-tuning of parameters in a pre-trained model, avoiding the convergency issues; the use of a pre-trained denoiser to map adversarial samples to benign ones is also analyzed. This study shows promising results when tested on FGSM and PGD.
Lastly another strategy aims to undo the effects of adversarial attacks, and it is called adversarial purification. This approach leverages additional algorithms and models to preprocess samples prior to classification. These algorithms can be developed with different focuses: one approach is pure denoising, while another takes advantage of the low-distortion constraints typical of adversarial attacks. Because adversarial attacks aim to apply minimal perturbations to shift samples into incorrect regions of latent space, adversarial purification functions by shifting points near decision boundaries back into the correct manifold. This technique is particularly advantageous in ADD due to the strict perturbation constraints involved. To the best of our knowledge, this approach has not yet been extensively studied within the ADD domain, despite its promising potential. However, it has been explored in the image domain in studies like DISCO [17], which demonstrated a performance improvement of 46.76% over existing methods in adaptive scenarios. The denoising approach considers the adversarial attack perturbation as noise that can be separated from the track. In the image field, several denoising mechanisms have been proposed [19, 29, 45]. Some problems in this approach are the negative effect on the accuracy of clean samples, as some features get treated as noise [19], but it offers several advantages such as using separate models with respect to the classifier and the possibility of cascading the denoiser an indeterminate amount of times which makes it harder for an attacker to adapt to it.

3. Proposed Method

This section will focus on the specific problems and solutions to be tested in the study.

3.1. Problem Formulation

The ADD problem is formally defined as follows. Let us consider a discrete-time input speech signal $\mathbf{x}$ with sampling frequency f_s and associated with a class $y \in \{0, 1\}$, where 0 denotes that the signal is authentic and 1 indicates that it has been synthetically generated. The goal of this task is to develop a detector $\mathcal{D}$ that estimates the class of the signal $\mathbf{x}$ as $\hat{y} \in [0, 1]$, where $\hat{y}$ is the likelihood of the signal $\mathbf{x}$ being fake. The objective of $\mathcal{D}$ is to achieve accurate classification, such that, ideally, $\mathcal{D}(\mathbf{x}) \approx y$.
An adversarial attack A attempts to compromise the performance of $\mathcal{D}$ by generating a small perturbation δ to the original sample $\mathbf{x}$, producing a perturbed sample

$$\mathbf{x}_{\text{adv}} = \mathbf{x} + \delta \tag{1}$$

that is close to $\mathbf{x}$ under a distance metric, typically

$$\|\delta\|_p \leq \epsilon, \tag{2}$$

where ϵ is the maximum perturbation allowed. The adversary's objective is to find a perturbation δ such that

$$\mathcal{D}(\mathbf{x}_{\text{adv}}) \neq \mathcal{D}(\mathbf{x}), \tag{3}$$

which could result in either a false positive (classifying authentic audio as synthetic) or a false negative (classifying synthetic audio as authentic). To counteract adversarial attacks, an adversarial purification defense mechanism is introduced as a preprocessing step through a purification module

$\mathcal{P}$, which maps an adversarial sample $\mathbf{x}_{\text{adv}}$ to a purified version $\mathbf{x}_{\text{p}} = \mathcal{P}(\mathbf{x}_{\text{adv}})$ that approximates the original unperturbed sample x. The purified input $\mathbf{x}_{\text{p}}$ should ideally enable the detector $\mathcal{D}$ to restore correct classification, such that

$$\mathcal{D}(\mathcal{P}(\mathbf{x}_{\text{adv}})) = \mathcal{D}(\mathbf{x}). \tag{4}$$

3.2. Synthetic Speech Detectors

In this study, we employ two different state-of-the-art synthetic speech detection models to perform our experiments and evaluate the transferability and robustness of adversarial attacks across different architectures.
We use a first version of a RawNet2 [50] as a victim model on which to perform our white-box attacks. This setup allows us to gain insights into the model's vulnerability when we have full access to its internal parameters and structure. We will later on refer to this model as *RawNet2-Target*.
To further investigate the transferability of these attacks within the same model architecture, we then employ a re-trained version of RawNet2, assessing what we refer to as *intra-model* transferability of such attacks. We will later on refer to this model as *RawNet2-Retrained*.
Additionally, we evaluate the *inter-model* transferability of these attacks by employing a ResNet model [1], which we configured to process raw audio inputs (1D) rather than 2D features such as spectrograms. This multi-model approach enables us to comprehensively assess the adaptability of adversarial attacks across both similar and distinct synthetic speech detectors architectures, providing valuable insights into the effectiveness and limitations of such detectors in adversarial scenarios. We will now briefly summarize the characteristics of these two synthetic speech detectors.

3.2.1 RawNet2

RawNet2 is a CNN designed to classify raw audio waveforms directly without the need for preliminary feature extraction.
Originally introduced in the ASVspoof 2019 challenge and later included as a baseline in ASVspoof 2021, RawNet2's approach leverages raw waveform input to enable the model to capture subtle, high-level patterns in audio data as the volume of available training data grows.
The model's architecture is structured as follows, from the bottom up:

- **SincNet layer**: This layer employs a SincNet model [37] that processes the raw speech waveform by generating filterbanks on the Mel scale and filtering the input signal. Only the low and high cutoff frequencies are learned directly from data, which, as experimental results show, enables the network to develop more effective and meaningful filter bank structures and outputs.
- **Residual Blocks**: Six residual blocks are stacked sequentially, each containing batch normalization, leaky ReLU activation, and convolutional layers. These blocks are designed to facilitate the propagation of information through deeper layers, which helps maintain performance and stability as the network grows in depth.
- **Attention Mechanisms**: Applied to the outputs of the residual blocks, attention mechanisms enhance the model's ability to focus on more discriminative features, thereby improving representation quality.
- **GRU layer**: A Gated Recurrent Unit (GRU) layer follows, incorporating a gating mechanism to regulate information flow. GRUs are specifically designed to address the vanishing gradient problem, where gradients diminish during backpropagation, which can otherwise impede the model's ability to learn effectively.
- **Fully Connected Layers**: Two fully connected layers are used at the top. The first layer takes the GRU output as input, and the second layer uses the output of the first layer. This sequence leads to the final Softmax layer, which produces probability scores as the model's output.

3.2.2 ResNet

The original ResNet model used as input spectrograms, 2D representation of audio signals. Spectrograms are defined as the Decibel (Db) value of the squared Short Time Fourier Transform (STFT), a Fourier transform that is used to represent the change in frequency and phase content over time. In order to obtain the STFT of a signal we need to segment the signal with arbitrary window, window length and hop length, and compute the Discrete Fourier Transform (DFT) of each segment. Squaring the results we isolate the magnitude component of the STFT, representing the frequency content of the signal in the specified window. Plotting these frequency contents as a function of time we obtained a 2D signal which, with adequate window parameters, has minimal information loss with respect to the original audio signal.
The ResNet model used in this study is a CNN modified as to take raw waveforms as input, but computing an internal pre-processing to obtain the corresponding spectrogram before going into any internal layer.
We performed such modification in order to use ResNet as our *control model* and allow adversarial white-box attacks, designed for RawNet2, to run also on this model.

- **Raw waveform processing:** The model begins by processing the input raw waveform through a series of convolutional layers and residual blocks. The convolutional layers capture local features of the raw waveform, while the residual blocks, which are a key component of the ResNet architecture, use skip connections. These skip connections allow the model to bypass certain layers, helping to preserve information as it passes through the network. This is crucial in mitigating the issues typically associated with deep neural networks, such as vanishing gradients and performance degradation when networks become very deep.
- **Leaky ReLU activations:** After each convolutional operation, a Leaky ReLU activation function is applied. This activation function is specifically chosen because it helps the model better capture and represent complex, subtle patterns in the raw audio data. Leaky ReLU avoids the issue of dead neurons, which can occur with traditional ReLU, by allowing a small, non-zero gradient when the input is negative. This ensures that information continues to propagate through the network, particularly for negative values, improving the model's ability to learn from a wider range of features in the waveform.
- **Frame-level embeddings and aggregation:** Consistent with the approach used in RawNet2, ResNet generates frame-level embeddings from the output of the residual blocks. These embeddings represent local features within small segments of the waveform. These frame-level features are then aggregated into a single, global embedding that summarizes the entire utterance, allowing the model to capture long-range dependencies and contextual information across the entire input sequence. This process helps the model understand both local details and broader patterns, which is critical for tasks like speech recognition or classification.
- **Fully connected layers for refinement:** After the residual learning blocks and the frame-level embeddings are aggregated, two fully connected layers are used to further refine the learned features. These layers allow the model to combine the information from the entire network, enhancing the overall feature representation. The final output of the model is produced through a binary log softmax activation function, which ensures that the output is in the form of probabilities that sum to one, making it suitable for binary classification tasks. This final activation produces the classification result, indicating the model's prediction based on the input audio.
- **Efficient training and convergence:** One of the key strengths of the ResNet architecture is its ability to scale to hundreds or even thousands of layers while still maintaining efficient training. The use of residual blocks with skip connections significantly improves the network's ability to learn effectively from deep models. These skip connections prevent the loss of important features and ensure that the model can converge quickly, even in deep networks. This makes ResNet an ideal choice for processing complex, high-dimensional data like raw waveforms, where the benefits of deep learning are often needed to capture intricate patterns in the data.

3.3. Considered Attacks

We employed two different kinds of attacks to assess the robustness of our synthetic speech detectors and to test the performance of the defense mechanisms: FGSM and PGD.
Originally developed in the image domain, both FGSM and PGD have become widely used as standard benchmarks in adversarial attack research across various fields, including also audio-related domains such as Automatic Speech Recognition (ASR) and ASV, but lately also ADD.
Both FGSM and PGD were initially designed to exploit vulnerabilities in image classifiers, by perturbing pixel values in ways that are imperceptible to humans but cause misclassifications by Neural Networks (NNs). Over time, these attacks have been adapted to the audio and speech domains. Adversarial attacks against speech-processing models are different from those against image- or text-processing models in that, as these can be trained by original audio signals or frequency features, perturbations against them can be divided into the time domain and frequency domain. As the name suggests, time-domain perturbations are perturbations introduced to the sampling value of the original audio, whereas frequency-domain perturbations are perturbations introduced to acoustic features, such as Mel Frequency Cepstral Coefficients (MFCCs).
In this study, in order to attack *RawNet2-Target*, we applied both FGSM and PGD on 1D raw waveforms, i.e., we generated time-domain perturbations.
We will now briefly explain to the reader how FGSM and PGD perturbations are generated.

3.3.1 FGSM

The FGSM is a widely-used white-box attack designed to mislead DL models by introducing a perturbation to the input which is minimal but targeted as to move the latent space representation of the signal towards the region of erroneous classification.
This attack perturbs the input raw waveform following the direction of the gradients of the model's loss function, but maximizing the likelihood of incorrect classification and minimizing perceptible differences to the human ear. In other words, FGSM maximizes the loss function while keeping the perturbation constrained.
Given our detection model $\mathcal{D}$, the target label y, and the loss function used to train the network $L(f(\mathbf{x}), y)$, the perturbation δ obtained through the FGSM attack with a certain step-size ϵ is:

$$\delta = \epsilon \cdot \text{sign}\left(\nabla_{\mathbf{x}} L(f(\mathbf{x}), y)\right) \tag{5}$$

By adding this computed δ to the input, we obtain an FGSM adversarial example $\mathbf{x}_{\text{adv}} = \mathbf{x} + \delta$ crafted to cause a targeted misclassification.

3.3.2 PGD

The PGD attack is an extension of the FGSM attack, where the perturbation is applied iteratively over multiple steps. This iterative process refines the adversarial perturbation with each step, resulting in a more powerful and robust attack compared to FGSM.
In each iteration, the update follows the same principle as FGSM, but with a smaller step size α. After each update, the perturbed input is "projected" back into an ϵ-bounded ball around the original input to ensure that the perturbation remains within the imperceptibility constraint. Given the perturbation:

$$\delta = \text{Clip}_{\epsilon}\left(\mathbf{x}_i + \alpha \cdot \text{sign}\left(\nabla_{\mathbf{x}} L(f(\mathbf{x}), y)\right)\right) \tag{6}$$

The update rule for each iteration t in PGD is as follows:

$$\mathbf{x}_{\text{adv}}^{(i+1)} = \mathbf{x} + \text{Clip}_{\epsilon}\left(\mathbf{x}_i + \alpha \cdot \text{sign}\left(\nabla_{\mathbf{x}} L(f(\mathbf{x}), y)\right)\right) \tag{7}$$

The pipeline for the application of the PGD attack is found in figure 1.
PGD achieves a higher attack success rate than FGSM, as it allows for more precise control over the adversarial strength by adjusting the step size α and the number of iterations. This

provides a balance between maintaining the imperceptibility of the perturbation and increasing the effectiveness of the attack.

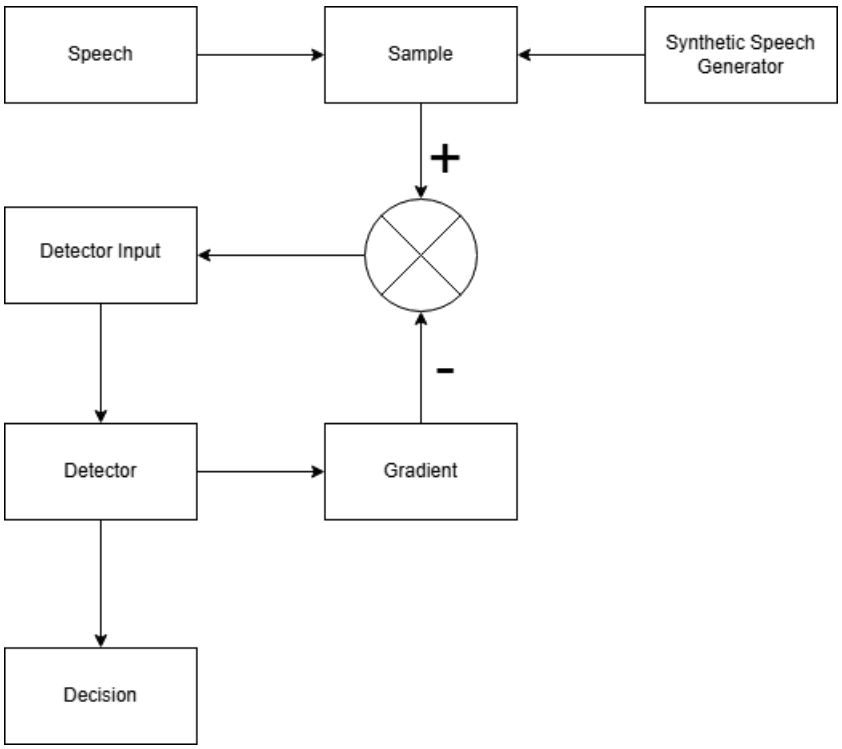

Figure 1: PGD pipeline

3.4. Adversarial Purification Methods

We propose three defense mechanisms against adversarial attacks with different paradigms: DEMUCS, SepFormer and DISCO.
These three defenses are all meant to work as different adversarial purification strategies, though they are initially conceived and trained through different methods and with different goals.

3.4.1 Denoising: DEMUCS

DEMUCS [11] is a denoiser designed to separate a speech signal from its underlying noise. Given a signal $\mathbf{x} = \mathbf{y} + n$, where $\mathbf{y}$ is the clean speech and n is the noise, DEMUCS aims to recover $\mathbf{y}$ by applying a function $f(\mathbf{x}) \approx \mathbf{y}$.
The architecture of DEMUCS is built around an encoder-decoder structure with U-Net skip connections. The encoder takes the raw waveform $\mathbf{x}$ as input and outputs a latent representation $E(\mathbf{x}) = z$. The encoder consists of multiple layers, each containing:

- A convolutional layer
- A ReLU activation function
- Another convolutional layer
- A GLU (Gated Linear Unit) activation function

This latent representation z is then passed to an LSTM network R, which performs the operation $R(z) = \text{LSTM}(z) + z$. This allows the model to capture temporal dependencies in the signal. The LSTM network consists of two LSTM layers and several hidden units. Finally, the decoder reconstructs an estimate of $\mathbf{y}$ from the output of the LSTM network. Each layer of the decoder

applies a convolution, a GLU activation, another convolution, and, except for the final layer, a ReLU activation.
We chose to employ DEMUCS as a first purification method to explore whether this approach can effectively address adversarial noise in the same way it handles real-world noise.
Our underlying hypothesis is that adversarial noise introduced by attacks may, in some cases, be treated similarly to real background noise. Specifically, we propose that adversarial noise could potentially be separated from the clean signal using denoising techniques, much like how typical background noise is handled. This hypothesis assumes that adversarial noise shares certain statistical or temporal properties with regular noise, making it possible for a denoiser like DEMUCS to model and mitigate such perturbations, thus recovering the original speech sample.

3.4.2 Denoising: SepFormer

The SepFormer [49] model was originally designed as a speech separator. A speech separator is a model that has been designed to separate speech signals from audio inputs that involve multiple distinct speakers, focusing on learning both STLT dependencies.
The model used in this study was trained to perform speech enhancement instead, which is a task that focuses on enhancing the quality of a speech signal by removing background noise.
The model architecture comprises an encoder, a masking network, and a decoder. The encoder takes the raw time-domain input signal $\mathbf{x}$ and generates an STFT-like representation k using a convolutional layer, which helps map the audio signal into a learned latent space $h = ReLU(Conv1D(\mathbf{x}))$. The masking network then chunks the input into segments and processes each one of them independently to find the speech signal and noise signal masks. It applies layer normalization and a linear transformation to the encoded representation h, followed by chunking the input and processing each chunk independently through SepFormer blocks. The SepFormer block consists of two main Transformer-based components:

- **The IntraTransformer** which models short-term dependencies within each chunk
- **The InterTransfromer** which captures long-term dependencies across chunks, using residual connections and multi-head self-attention to improve information flow and gradient propagation.

The decoder then reconstructs the separated speech signals using the masks generated by the masking network and applies an inverse transformation through a transposed convolution layer to return each separated signal $s_k = Conv1d - transpose(m_k * h)$, where m_k is the mask for the speaker k. The SepFormer leverages the parallel processing capabilities of Transformers, making it faster and less memory-intensive compared to Recurrent Neural Network (RNN)-based systems for speech separation tasks.
The use of this denoiser will test the same hypothesis as DEMUCS, but using a speech enhancement network instead of a speech separation one. These two approaches solve the task differently and with different results, and they may lead to different performances when used in the task of adversarial purification.

3.4.3 Input Refactoring: DISCO

It has been hypothesized [6, 12] that human perception is dependent on the statistical regularities in the natural world and its use to overcome the difficulty in understanding audio and video scenes under adverse conditions. Under this hypothesis, natural images and audio signals form a low-dimension manifold in signal space, denoted as the *signal manifold*, to which human perception is highly tuned. Being used to adverse conditions, such as noisy environments, and continuously learning from our surroundings, has given our senses the ability to project signals which would be barely outside the signal manifold inside it, consciously perceiving them as though they always were signals with the statistical properties necessary to be part of the signal manifold.
As we've discussed previously, adversarial attack aim at moving the latent space representation of a model into a region of erroneous classification through the shortest possible distance, which would make these attacked samples *barely outliers*; just like in the field of human perception,

adversarial purification can be the process of projecting barely outliers into the manifold. This process is a generative process, which, given a barely outlier signal, synthesizes a new signal inside the manifold.
DISCO [17] was created as an RGB image purification model with the hypothesis that manifold projection is a conditional operation: the synthesis of a natural image given the perturbed one. Assuming that an attack does not change the global structure of an image, conditional modeling could be implemented with an implicit function, by having a model learn the conditional representation of the image appearance in the neighborhood of each pixel, given a feature extracted at that pixel. To achieve this goal Ho et al. proposed a model based on an encoder-decoder structure, with an Enhanced Deep Residual Network (EDSR) encoder composed of a head with several convolutional layers, body with other convolutional layers and several residual block and a tail which upscales the resolution of the spectrogram and applies one last convolutional layer; the decoder uses several hidden layers and ReLU functions, and it is trained to predict the pixel value of the clean image using as input the output of the encoder.
Analyzing the performance of DISCO will allow us to compare a specifically trained denoiser with respect to more generic ones, and look into the transferability of the manifold projection method to the audio field.

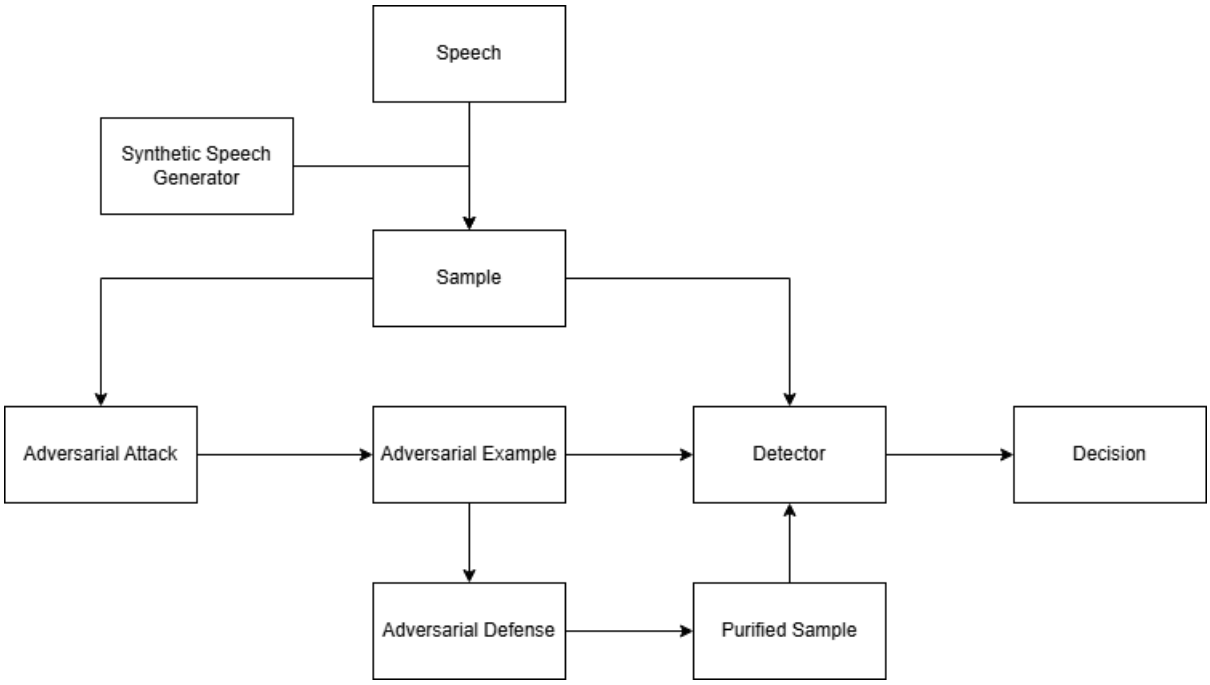

Figure 2: Testing pipeline

4. Experimental Setup

This section will cover the algorithms, data, and parameters used in the training and evaluation of the SSD, attacks, and defense mechanisms.

4.1. Considered Dataset

We utilize the ASVSpoof2019 dataset, which is designed specifically for ASV and ADD research, to train and test our SSD. We focus on the Logical Access (LA) partition, containing both real and synthesized speech signals generated through various algorithms. The LA partition comprises 3 sub-sets: training, development, and evaluation. Here is a summary of the three partitions' composition:

- Train Set: encompasses a total of 25,380 utterances, of which 2580 are real. The remaining 22800 spoofed ones belong to 6 different synthesis algorithms (named A01, A02, ..., A06), having 3800 tracks each.
- Dev Set: encompasses a total of 24844 utterances, of which 2548 are real. The generators are the same as before, each one holds a total of 3716 tracks.
- Eval Set: encompasses a total of 71237 utterances (7355 real). This partition includes samples generated with 13 techniques (A07, ..., A19), each one has 4914 samples. Note that A16 and A19 are the same spoofing systems of A04 and A06.

All SSDs, namely *RawNet2-Target*, *RawNet2-Retrained* and ResNet, were trained on the training partition of ASVspoof2019 and validated on the development partition. DISCO was trained and validated by pairing two versions of the development partition: the unprocessed development partition and the FGSM-attacked development partition, with an 80/20 split for training and validation. All tests used the evaluation partition of ASVspoof2019. For all the considered data, we assumed a sampling rate equal to 16kHz.

4.2. Synthetic Speech Detectors Training

For this study, we maintained the original RawNet2 architecture with one modification: the input size was reduced to 47,104 samples (around 3 seconds of audio) instead of the original 4 seconds. This adjustment optimizes compatibility with the spectrogram-based defense mechanisms while maintaining model performance. The training of the RawNet2 used a learning rate of 10^{-3}, a weight decay of 10^{-4}, a learning rate scheduler with patience equal to 6, and an early stopping after 18 epochs with no improvement. Cross-entropy loss is used with the Stochastic Gradient Descent (SGD) optimizer. Due to the dataset imbalance between real and fake samples, we balanced the training data.
We adopted the same experimental setup for both *RawNet2-Target* and *RawNet2-Retrained*, the only difference being in the initial seed number.

As for the ResNet SSD, we modified its architecture to take as input a raw waveform with the same sample size as the RawNet2 model. This modification ensures that our attacks, optimized using the 1-dimensional gradients of *RawNet2-Target*, can be effectively transferred to ResNet. The input is internally converted to a spectrogram using 2048 frequency bins, a hop length of 512, a rectangular window, and a window length of 2048. The training of the ResNet used a learning rate of 10^{-4} and an early stopping after 10 epochs with no improvement. The loss function considered is the cross entropy loss function and the optimizer used is the Adam optimizer. To balance the difference in real and fake data, weights of 0.9 and 0.1 were respectively used. Just like in the RawNet2 training, the starting sample of each audio was randomized.

4.3. Adversarial Examples Generation

We created an attacked copy of the development set using the FGSM attack and two attacked copies of the evaluation set using the FGSM attack for one and the PGD attack for the other. All attacks were computed leveraging the *RawNet2-Target*'s gradients and an ϵ of 0.002 was used. The number of iterations in the PGD attack was set to 15.
We applied the attacks to both genuine and synthetic speech tracks, with the goal of triggering misclassifications in both scenarios, causing authentic speech to be recognized as synthetic and synthetic speech to be recognized as authentic.
During the attack, we disabled the randomization of the starting sample in order to maintain consistency across multiple trials. This decision was made to ensure that the attack could be evaluated under controlled conditions, allowing for a more accurate assessment of the model's vulnerability to adversarial examples. By keeping the starting sample fixed, we were able to minimize any potential variation in the results that could arise from random initialization, thereby ensuring that the impact of the adversarial attacks was the primary factor influencing performance.

4.4. Defenses Configuration

4.4.1 DEMUCS Application

The pre-trained DEMUCS architecture from [11] is applied to denoise the clean, FGSM-attacked, and PGD-attacked evaluation datasets. This architecture uses the structure described in 3.4.1, initializing all model parameters using the scheme proposed by [15], which involves scaling the weights of each layer to account for the non-linear activation functions, specifically ReLU and PReLU. The scheme addresses the tendency of deep networks to either amplify or diminish signals excessively across layers, which can hinder convergence. For forward propagation, the initialization aims to maintain consistent variance across layers by setting each weight's variance to $\frac{\mathbf{n}}{\mathbf{2}}$, where $\mathbf{n}$ is the number of inputs to each neuron. This scaling helps prevent signal magnification or reduction across layers. For backward propagation, the initialization similarly scales weights to prevent gradient vanishing or explosion. By carefully deriving the variance scaling factor, this approach enables deep ReLU-based networks to converge effectively without pre-training on shallower networks or using auxiliary classifier.
The input was upsampled by a factor of 2 to increase accuracy, and the output was downsampled by the same factor. The resampling was done through a Sinc interpolation filter as part of the end-to-end training.

4.4.2 SepFormer Application

The pre-trained SepFormer architecture from [48] is applied to denoise the clean, FGSM-attacked, and PGD-attacked evaluation datasets. This version of the model was trained on the WHAMR! dataset at 16khz sampling frequency, using the same architecture as the original SepFormer. It reaches 13.5 Db Scale invariant Signal to Noise Ratio Improvement (Si-SNRi) on the test set of WHAMR! dataset.

4.4.3 DISCO Training

For this study, the DISCO architecture has been modified to process mono-channel spectrograms. The training pipeline converts audio into spectrograms on the fly, using a window length of 2048, a hop length of 512, a Hanning window, and centered padding aligning each frame $D[:, t]$ at $y[t * hop\_length]$. DISCO is trained to take the adversarial spectrogram as input and output predicted values for each time-frequency bin. The output, which we will call *predicted spectrogram*, is compared to the clean spectrogram through an L1 loss function. To update the prediction rules trying to minimize the loss function, we used a learning rate of 10^{-4} and an Adam optimizer. We trained the framework for 50 epochs while disabling sample randomization to ensure that clean and attacked spectrograms refer to the same audio segment.
Figure 3 shows the complete pipeline for the training process of DISCO.

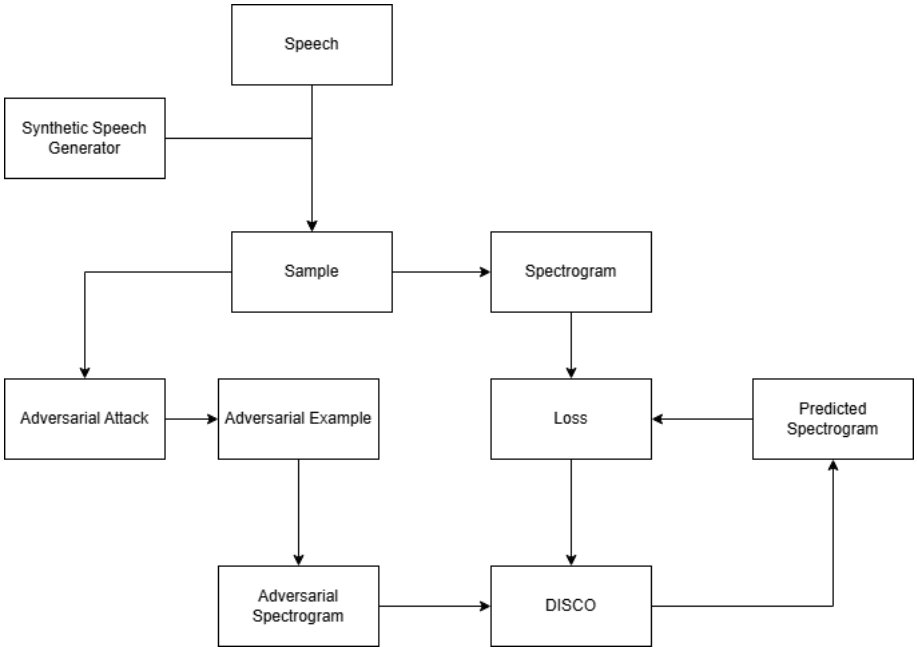

Figure 3: DISCO training pipeline

5. Results

In this section, we present and analyze the results of our experiments, discussing both the findings and their limitations. We begin by presenting the results obtained after applying attacks without any defense, followed by an analysis of how performance changes when each of the three adversarial purification methods is introduced.
The evaluation set from ASVspoof2019, including its clean, FGSM-attacked, and PGD-attacked versions, is used as the test set. The complete testing pipeline is found in figure 2.

5.1. Evaluation Metrics

Metrics for these evaluations include the Receiver Operating Characteristic (ROC) curve with Area Under the Curve (AUC) and balanced accuracy using a 0.5 threshold.
The balanced accuracy is a metric used to evaluate the performance in a binary or multiclass case classification problem. Given a decision rule for the classification, we can count the number of correctly and incorrectly classified samples in a dataset. Considering the binary case, we account for the difference in the number of elements belonging to each class (which we will call 0 and 1) by defining 4 values:

- **True Positive (TP):** This represents the number of values belonging to class 1 and being classified as 1.
- **False Positive (FP):** This represents the number of values belonging to class 0 and being classified as 1.
- **True Negative (TN):** This represents the number of values belonging to class 0 and being classified as 0.
- **False Negative (FN):** This represents the number of values belonging to class 1 and being classified as 0.

Starting from these values, we can calculate the two variables that will determine the balanced accuracy:

- **True Positive Rate (TPR):**

$$TPR = \frac{TP}{TP + FP}.$$

- **True Negative Rate (TNR):**

$$TNR = \frac{TN}{TN + FN}.$$

With these values, we can calculate the balanced accuracy B as

$$B = \frac{TPR + TNR}{2}.$$

The balanced accuracy can range from 0 to 1, with 1 being a perfect classification of every sample. The ROC curve is a graph that represents the change in the model accuracy with respect to the change in the choice for the classification threshold. In this curve, the y-axis represents the TPR, and the x-axis represents the False Positive Rate (FPR). The FPR is defined as $1 - TNR$, or:

$$FPR = \frac{FP}{FP + TN}.$$

A ROC representing a good performance will have a sharp rise. A perfect ROC would appear as a straight horizontal line situated at 1.0. The diagonal is the line that separates positive performance from poor performance; curves higher than the diagonal have good performance, under the diagonal the performance is poor and close to the diagonal we are in the random guessing region.
We can also use the ROC to derive the optimal threshold for making decisions starting from the output probability scores. The optimal point p is the one that satisfies the equation

$$TPR(p) = 1 - FPR(p)$$

Lastly, the AUC is a metric that is used to represent numerically the overall performance of the ROC. It is defined as the area underneath the ROC curve, and it ranges from 0 to 1, with 1 being the perfect classification case.

5.2. Baseline results

	Balanced accuracy	AUC
Clean	**86,7%**	**95,2%**
FGSM	29,1%	**8,1%**
PGD	**4,6%**	**0,2%**

Table 1: *RawNet2-Target* Results

	Balanced accuracy	AUC
Clean	82%	**91,4%**
FGSM	72,4%	**85,7%**
PGD	72,7%	84,2%

Table 2: *RawNet2-Retrained* Results

	Balanced accuracy	AUC
Clean	84,2%	**92,8%**
FGSM	74,7%	83,7%
PGD	74,8%	81,3%

Table 3: ResNet Results

Table 4: Balanced accuracy and AUC values of the 3 SSDs without deploying any defense strategy. Results are shown for the clean test data, the FGSM-attacked test data, and the PGD-attacked test data.

Table 4 shows the 3 SSDs results in terms of balanced accuracy and AUC values on the clean, FGSM-attacked and PGD-attacked test datasets.
On the clean evaluation set, both balanced accuracies and AUCs exhibit good values for all three models. On the other hand, both attacks degrade the *RawNet2-Target* performances drastically. This is an expected outcome, as we are performing well-crafted white-box attacks in this scenario. Nevertheless, both FGSM and PGD only cause a slight drop in the *RawNet2-Retrained* and the ResNet performances, showing poor *intra-model* and *inter-model* transferability.
While the reduced performance on *RawNet2-Retrained* and ResNet was anticipated, given that the attacks were tailored to the internals of *RawNet2-Target*, the significant impact on *intra-model* transferability is noteworthy. These findings suggest that adversarial examples are highly sensitive to even minor changes in internal model parameters, even within the same architecture trained on the same data following the same experimental setup. Evidently, even minor adjustments in weight initialization during retraining can result in new decision boundaries, which adversarial examples crafted for the original model struggle to exploit. This is because adversarial perturbations often align with specific decision boundaries that change when the model is retrained, even slightly, especially for high-dimensional audio input. Additionally, adversarial examples in SSD often rely on certain frequency or temporal patterns that may be disrupted in retrained models, leading to reduced effectiveness across versions [14].
Another interesting result to pay attention to is the difference in performance degradation between FGSM and PGD: while in the *RawNet2-Target*, the PGD attack degrades the performance by a wider margin, the drop in the *RawNet2-Retrained* model and in the ResNet model are almost identical. This suggests that these models may be less sensitive to the specific characteristics of adversarial noise and more vulnerable to a general noise addition. Indeed, the fact that the FGSM attack - which introduces higher-energy noise with respect to PGD - produces a similar performance decline as PGD itself, highlights that the issue may not lie in specific vulnerabilities to crafted perturbations but points to a broader, systemic sensitivity to noise. This aligns with a broader understanding in the field: SSDs and similar audio models often show sensitivity to noise, which has led to the widespread use of noise injection as a data augmentation technique during training. By exposing models to noise, noise injection enhances their general robustness to real-world variability, including resistance to adversarial attacks.
The ROC curves in Fig. 4 illustrate the performance differences across various evaluations: the *RawNet2-Target* model tested on the clean dataset, the *RawNet2-Target* model tested on the PGD-attacked dataset, the *RawNet2-Retrained* model tested on the PGD-attacked dataset, and the ResNet model tested on the PGD-attacked dataset.

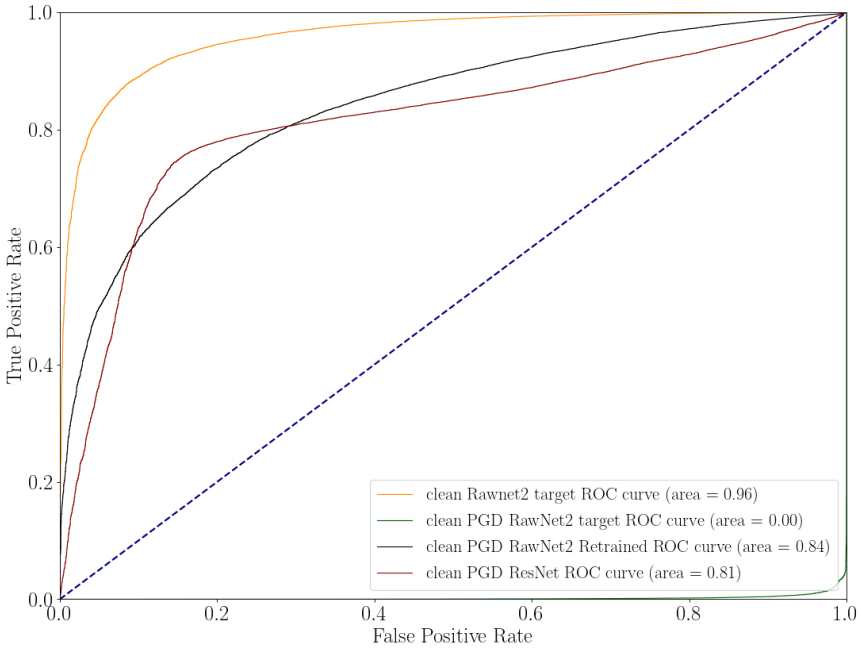

Figure 4: ROC curves of *RawNet2-Target* on clean and PGD-attacked data, and of *RawNet2-Retrained* and ResNet on PGD-attacked data only.

These results further validate our previous observations. The ROC curve for the clean *RawNet2-Target* shows a steep rise and a high AUC, indicating strong performance in the absence of adversarial noise. In contrast, the PGD-attacked *RawNet2-Target* curve remains mostly flat, reflecting a significant performance drop due to the adversarial perturbations, as shown by the low AUC. The PGD-attacked *RawNet2-Retrained* model exhibits a smoother curve than the *RawNet2-Target* tested on clean data, highlighting how simple retraining can improve adversarial robustness, though revealing a persistent vulnerability to additive noise, with a performance drop that, while far less severe than in the PGD-attacked *RawNet2-Target* case, remains noticeable.

5.3. DEMUCS results

	Balanced accuracy	AUC
Clean	**87,2%**	**94,8%**
FGSM	22%	**8,9%**
PGD	**6,5%**	**1,1%**

Table 5: DEMUCS + *RawNet2-Target* Results

	Balanced accuracy	AUC
Clean	81,9%	**91,4%**
FGSM	75,5%	**86,8%**
PGD	75,6%	**86,3%**

Table 6: DEMUCS + *RawNet2-Retrained* Results

	Balanced accuracy	AUC
Clean	80,3%	**94,8%**
FGSM	75%	82,2%
PGD	75,4%	80,7%

Table 7: DEMUCS + ResNet Results

Table 8: Balanced accuracy and AUC values of the 3 SSDs with DEMUCS denoiser deployed as a defense strategy. Results are shown for the clean test data, the FGSM-attacked test data, and the PGD-attacked test data.

Table 8 shows the results of the three SSDs on the three evaluation sets when the DEMUCS denoiser is deployed as a defense mechanism.
Comparing them to Table 4, performances seem not to change significantly in any case.
These results seem to suggest a different distribution between the noise introduced by the adversarial attacks and the background noise removed by DEMUCS. This finding is further corroborated by the similar performances in the *RawNet2-Retrained* and ResNet adversarial cases, as they show balanced accuracy values that are very close: while the drop between the clean and adversarial cases for the baseline scenario is about 10%, we only see a 1-3% increase for DEMUCS, which means that only about 1 in 10 adversarial examples exhibit noises with characteristics that are similar to proper background noise.
Fig. 5 shows the ROC curves for the clean and PGD cases on the *RawNet2-Target*, the PGD case on the *RawNet2-Retrained* and the PGD case on the ResNet. The results are almost equal to the baseline scenario depicted in Fig. 4, further showcasing the neutral behavior of DEMUCS with respect to the ADD task.
Overall, these results suggest that more sophisticated and tailored signal purification strategies might be required to improve robustness against adversarial additive noise, such as FGSM and PGD, whose characteristics are fundamentally different from background noise.

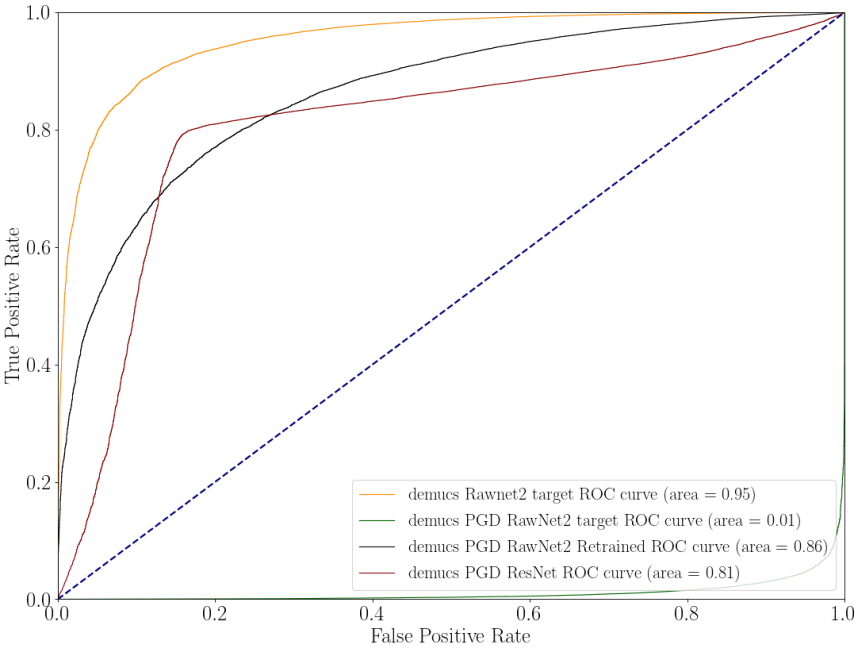

Figure 5: ROC curves of *RawNet2-Target* on clean and PGD-attacked data, and of *RawNet2-Retrained* and ResNet on PGD-attacked data only, with DEMUCS deployed as a defense mechanism.

5.4. SepFormer results

	Balanced accuracy	AUC
Clean	77,5%	**87,6%**
FGSM	60,4%	64,5%
PGD	50,2%	51,2%

Table 9: SepFormer + *RawNet2-Target* Results

	Balanced accuracy	AUC
Clean	78,3%	**85,3%**
FGSM	73,7%	80,6%
PGD	71,5%	77,6%

Table 10: SepFormer + *RawNet2-Retrained* Results

	Balanced accuracy	AUC
Clean	65,3%	71,2%
FGSM	54,1%	59,3%
PGD	54,4%	59,6%

Table 11: SepFormer + ResNet Results

Table 12: Balanced accuracy and AUC values of the 3 SSDs with SepFormer deployed as a defense strategy. Results are shown for the clean test data, the FGSM-attacked test data, and the PGD-attacked test data.

Table 12 presents some noteworthy findings, regarding the use of SepFormer as a defense strategy during the evaluation of the three SSDs.
When SepFormer was applied to the attacked samples, *RawNet2-Target* showed a significant improvement in performance on the attacked datasets, with a notable increase of up to 45% in balanced accuracy compared to the baseline results. On the other hand, *RawNet2-Retrained* did not benefit from SepFormer, with its performance on the attacked data remaining unchanged or slightly worse than the baseline, particularly in terms of AUC on PGD-attacked data. For ResNet, the effect of SepFormer was clearly detrimental, with accuracy dropping by approximately 20% and AUC decreasing by 20% to 25% on both FGSM and PGD-attacked data.
Even more surprisingly, all three models showed a decline in performance on clean data when preprocessed with SepFormer. Ideally, a defense strategy would not affect clean data performance, making this result unexpected.
In general, SepFormer had a much more pronounced impact on SSDs compared to DEMUCS. The notable differences between SepFormer and DEMUCS can be attributed to their distinct mechanisms: speech separation isolates the speech signal from noise, while speech enhancement uses filtering and noise-masking techniques over a broader frequency range. This more aggressive filtering approach in speech enhancement may help reduce adversarial perturbations, but it can also introduce distortions that negatively affect the classification of clean samples. ResNet, in particular, seems more sensitive to these distortions, likely due to the spectrogram conversion, which accentuates frequency content changes.

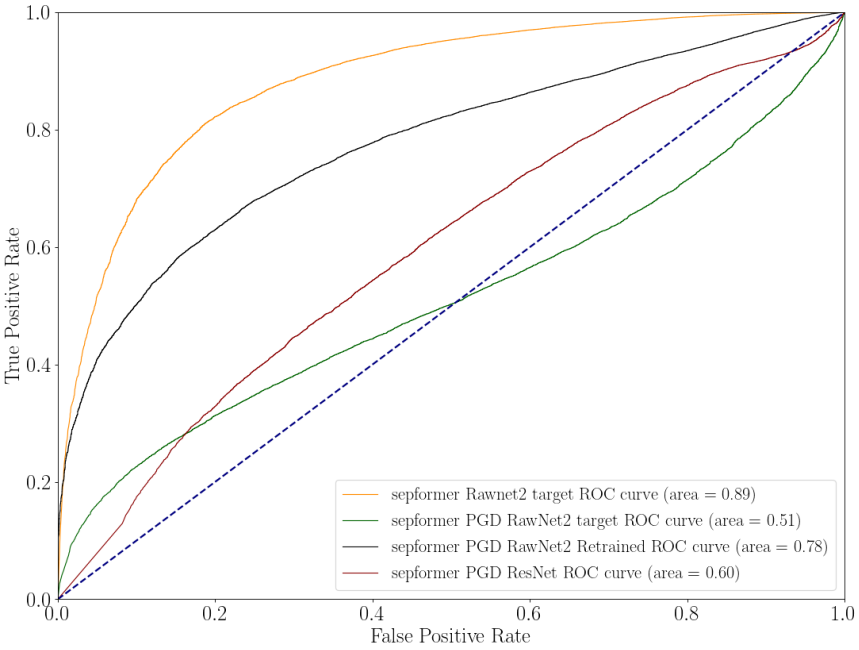

Figure 6: ROC curves of *RawNet2-Target* on clean and PGD-attacked data, and of *RawNet2-Retrained* and ResNet on PGD-attacked data only, with SepFormer deployed as a defense mechanism.

In Fig. 6, we report the ROC curves for the clean and PGD cases on the *RawNet2-Target*, the PGD case on the *RawNet2-Retrained* and the PGD case on the ResNet. Matching previous observations, the results differ significantly from the baseline and DEMUCS cases, and the most noticeable difference is in the *RawNet2-Target* PGD case, where the curve stays in the random guessing range. The other ROC curves are smoother than the baseline case, especially for the ResNet.

5.5. DISCO results

	Balanced accuracy	AUC
Clean	**84,5%**	**91,6%**
FGSM	31,9%	23,1%
PGD	20,9%	**11,5%**

Table 13: DISCO + *RawNet2-Target* Results

	Balanced accuracy	AUC
Clean	81,6%	**90,1%**
FGSM	78,7%	**87%**
PGD	78,7%	**86,9%**

Table 14: DISCO + *RawNet2-Retrained* Results

	Balanced accuracy	AUC
Clean	84,6%	**94,7%**
FGSM	80,1%	**88,4%**
PGD	79,9%	**88,3%**

Table 15: DISCO + ResNet Results

Table 16: Balanced accuracy and AUC values of the 3 SSDs with DISCO deployed as a defense strategy. Results are shown for the clean test data, the FGSM-attacked test data, and the PGD-attacked test data.

Table 16 presents the results following the application of the DISCO method for input data refactoring.
With respect to *RawNet2-Target*, while the performance improvements on attacked data are not as dramatic as those seen with SepFormer, there is still a notable increase in accuracy (up to 15%) and AUC. Overall, it is interesting to notice that the performances of the three SSDs against adversarial attacks improve in every considered scenario, even though to different extents. Even more notably, the impact on clean dataset performance is minimal, as one would ideally expect from an external defense module. These results suggest that manifold projection can represent a profitable technique also in adversarial audio purification.
The modest increase in accuracy on FGSM-attacked data with respect to the accuracy on PGD-attacked data, as regards *RawNet2-Target*, can perhaps be attributed to the carefully crafted, and yet smaller, perturbations introduced by the PGD attack, which result in adversarial samples being closer to the decision boundary of the correct class, compared to the likely larger distortions introduced by FGSM.
Given the promising results from this input refactoring defense strategy, further refining the DISCO architecture to better exploit the peculiarities of the audio domain over the image one, or training it to broaden the definition of what qualifies as a *barely outlier*, could potentially yield a more substantial improvement in adversarial robustness.

Fig. 7 shows the ROC curves for the clean and PGD cases on the *RawNet2-Target*, the PGD case on the *RawNet2-Retrained* and the PGD case on the ResNet. The results are similar to the baseline and DEMUCS cases, but tend to show an higher AUC.

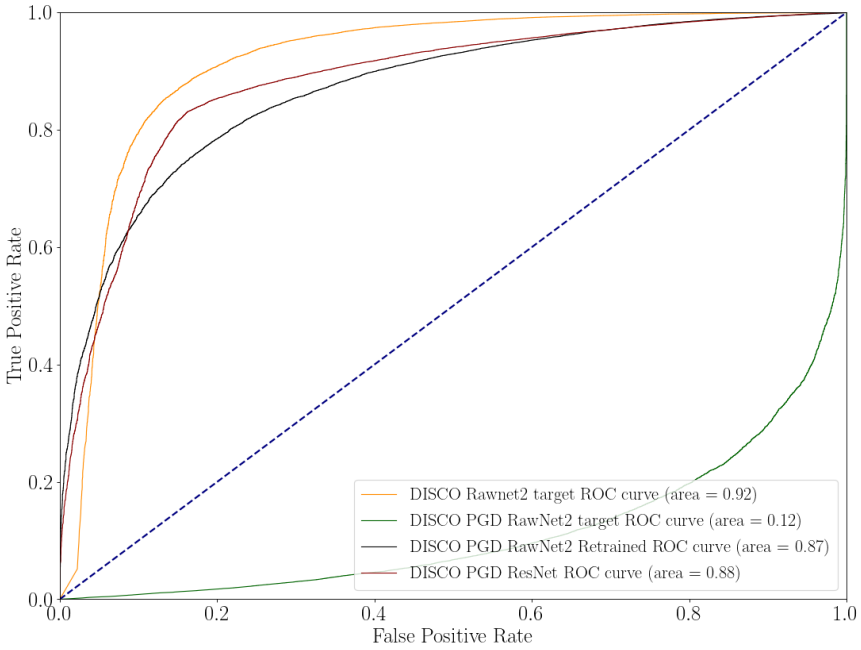

Figure 7: ROC curves of *RawNet2-Target* on clean and PGD-attacked data, and of *RawNet2-Retrained* and ResNet on PGD-attacked data only, with DISCO deployed as a defense mechanism.

6. Conclusion and Future Developments

In this study, we analyzed three defense mechanisms against adversarial attacks based on signal purification: denoising through DEMUCS and SepFormer, and input refactoring using DISCO. We studied their effect on SSDs' performance over both clean samples and adversarial examples crafted through FGSM and PGD white-box attacks. In particular, we considered three SSDs: the victim model, *RawNet2-Target*, a retrained version of it, *RawNet2-Retrained*, and a modified ResNet.
While baseline results showed poor *intra-model* and *inter-model* transferability of the considered attacks, the application of DEMUCS had no significant effect in any of the analyzed cases, with small changes in performance anytime it was applied. On the other hand, the deployment of SepFormer led to intriguing results: from huge improvements in the *RawNet2-Target* adversarial cases (up to 45%), to high drops in accuracy in the ResNet cases (around 20%).
Defense by input refactoring - using a modified version of DISCO - had a significant positive impact on all the considered SSDs, particularly in the *RawNet2-Target* case, with gains of up 15% in balanced accuracy and 0.15 AUC against PGD.
While the performance gap between DEMUCS and SepFormer is clear, a more detailed analysis of the factors driving this discrepancy would be highly valuable. Given the varied effects of these denoising mechanisms on different models—particularly in the case of SepFormer—an investigation into the specific frequency regions impacted by these methods could provide insight into their differing outcomes. Additionally, evaluating these defense mechanisms across a broader range of baseline models could help identify the features most critical for accurate classification in specific architectures. Given the obtained preliminary results, further exploration of the DISCO defense strategy within the audio domain also presents significant opportunities for refinement.

References

[1] Moustafa Alzantot, Ziqi Wang, and Mani B Srivastava. Deep residual neural networks for audio spoofing detection. 2019.

[2] Irene Amerini, Mauro Barni, Sebastiano Battiato, Paolo Bestagini, Giulia Boato, Tania Sari Bonaventura, Vittoria Bruni, Roberto Caldelli, Francesco De Natale, Rocco De Nicola, Luca Guarnera, Sara Mandelli, Gian Luca Marcialis, Marco Micheletto, Andrea Montibeller, Giulia Orru', Alessandro Ortis, Pericle Perazzo, Giovanni Puglisi, Davide Salvi, Stefano Tubaro, Claudia Melis Tonti, Massimo Villari, and Domenico Vitulano. Deepfake media forensics: State of the art and challenges ahead, 2024.

[3] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations, 2020.

[4] Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4312–4321. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Survey Track.

[5] B. T. Balamurali, Kinwah Edward Lin, Simon Lui, Jer-Ming Chen, and Dorien Herremans. Toward robust audio spoofing detection: A detailed comparison of traditional and learned features. *IEEE Access*, 7:84229–84241, 2019.

[6] H Barlow. The exploitation of regularities in the environment by the brain. *The Behavioral and brain sciences*, 24(4):602—7; discussion 652—71, August 2001.

[7] Clara Borrelli, Paolo Bestagini, Fabio Antonacci, Augusto Sarti, and Stefano Tubaro. Synthetic speech detection through short-term and long-term prediction traces. *EURASIP Journal on Information Security*, 2021, 04 2021.

[8] Zexin Cai, Weiqing Wang, and Ming Li. Waveform boundary detection for partially spoofed audio. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023.

[9] Georgia Channing, Juil Sock, Ronald Clark, Philip Torr, and Christian Schroeder de Witt. Toward robust real-world audio deepfake detection: Closing the explainability gap, 2024.

[10] Guangke Chen, Zhe Zhao, Fu Song, Sen Chen, Lingling Fan, Feng Wang, and Jiashui Wang. Towards understanding and mitigating audio adversarial examples for speaker recognition, 2022.

[11] Alexandre Defossez, Gabriel Synnaeve, and Yossi Adi. Real time speech enhancement in the waveform domain, 2020.

[12] Daniel J. Graham and David J. Field. Statistical regularities of art images and natural scenes: spectra, sparseness and nonlinearities. *Spatial vision*, 21 1-2:149–64, 2007.

[13] Onat Gungor, Tajana Rosing, and Baris Aksanli. Stewart: Stacking ensemble for white-box adversarial attacks towards more resilient data-driven predictive maintenance. *Computers in Industry*, 140:103660, 2022.

[14] Feng Guo, Zheng Sun, Yuxuan Chen, and Lei Ju. Towards the transferable audio adversarial attack via ensemble methods, 2023.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, 2015.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[17] Chih-Hui Ho and Nuno Vasconcelos. Disco: Adversarial defense with local implicit functions, 2022.

[18] Sonal Joshi, Saurabh Kataria, Yiwen Shao, Piotr Zelasko, Jesus Villalba, Sanjeev Khudanpur, and Najim Dehak. Defense against adversarial attacks on hybrid speech recognition using joint adversarial fine-tuning with denoiser, 2022.

[19] Dvij Kalaria, Aritra Hazra, and Partha Pratim Chakrabarti. Towards adversarial purification using denoising autoencoders, 2022.

[20] Piotr Kawa, Marcin Plata, and Piotr Syga. Defense against adversarial attacks on audio deepfake detection, 2023.

[21] Jaehyeon Kim, Jungil Kong, and Juhee Son. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech, 2021.

[22] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis, 2020.

[23] Hyun Kwon, Hyunsoo Yoon, and Ki-Woong Park. Acoustic-decoy: Detection of adversarial examples through audio modification on speech recognition system. *Neurocomputing*, 417:357–370, 2020.

[24] Xuechen Liu, Xin Wang, Md Sahidullah, Jose Patino, Héctor Delgado, Tomi Kinnunen, Massimiliano Todisco, Junichi Yamagishi, Nicholas Evans, Andreas Nautsch, and Kong Aik Lee. Asvspoof 2021: Towards spoofed and deepfake speech detection in the wild. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2507–2522, 2023.

[25] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019.

[26] Sapna Maheshwari. 'A.I. Obama' and Fake Newscasters: How A.I. Audio Is Swarming TikTok (Published 2023) — nytimes.com. `https://www.nytimes.com/2023/10/12/technology/tiktok-ai-generated-voices-disinformation.html`. [Accessed 02-11-2024].

[27] Daniele Mari, Davide Salvi, Paolo Bestagini, and Simone Milani. All-for-one and one-for-all: Deep learning-based feature fusion for synthetic speech detection. *arXiv preprint arXiv:2307.15555*, 2023.

[28] Mvelo Mcuba, Avinash Singh, Richard Adeyemi Ikuesan, and Hein Venter. The effect of deep learning methods on deepfake audio detection for digital investigation. *Procedia Computer Science*, 219:211–219, 2023. CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN – International Conference on Project MANagement / HCist – International Conference on Health and Social Care Information Systems and Technologies 2022.

[29] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples, 2017.

[30] Viola Negroni, Davide Salvi, Alessandro Ilic Mezza, Paolo Bestagini, and Stefano Tubaro. Leveraging mixture of experts for improved speech deepfake detection. *arXiv preprint arXiv:2409.16077*, 2024.

[31] Monisankha Pal, Arindam Jati, Raghuveer Peri, Chin-Cheng Hsu, Wael AbdAlmageed, and Shrikanth S. Narayanan. Adversarial defense for deep speaker recognition using hybrid adversarial training. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6164–6168, 2020.

[32] Michele Panariello, Wanying Ge, Hemlata Tak, Massimiliano Todisco, and Nicholas Evans. Malafide: a novel adversarial convolutive noise attack against deepfake and spoofing detection systems, 2023.

[33] Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. Grad-tts: A diffusion probabilistic model for text-to-speech, 2021.

[34] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis, 2018.

[35] Mouna Rabhi, Spiridon Bakiras, and Roberto Di Pietro. Audio-deepfake detection: Adversarial attacks and countermeasures. *Expert Systems with Applications*, 250:123941, 2024.

[36] Rishabh Ranjan, Mayank Vatsa, and Richa Singh. Uncovering the deceptions: An analysis on audio spoofing detection and future prospects, 2023.

[37] Mirco Ravanelli and Yoshua Bengio. Speaker recognition from raw waveform with sincnet. In *2018 IEEE spoken language technology workshop (SLT)*, pages 1021–1028. IEEE, 2018.

[38] Davide Salvi, Temesgen Semu Balcha, Paolo Bestagini, and Stefano Tubaro. Listening between the lines: Synthetic speech detection disregarding verbal content, 2024.

[39] Davide Salvi, Temesgen Semu Balcha, Paolo Bestagini, and Stefano Tubaro. Listening between the lines: Synthetic speech detection disregarding verbal content. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2024.

[40] Davide Salvi, Paolo Bestagini, and Stefano Tubaro. Exploring the synthetic speech attribution problem through data-driven detectors. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2022.

[41] Davide Salvi, Paolo Bestagini, and Stefano Tubaro. Reliability estimation for synthetic speech detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[42] Davide Salvi, Paolo Bestagini, and Stefano Tubaro. Synthetic speech detection through audio folding. In *ACM International Workshop on Multimedia AI against Disinformation*, 2023.

[43] Davide Salvi, Paolo Bestagini, and Stefano Tubaro. Towards Frequency Band Explainability in Synthetic Speech Detection. In *31st European Signal Processing Conference (EUSIPCO)*, 2023.

[44] Davide Salvi, Viola Negroni, Luca Bondi, Paolo Bestagini, and Stefano Tubaro. Freeze and learn: Continual learning with selective freezing for speech deepfake detection. *arXiv preprint arXiv:2409.17598*, 2024.

[45] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models, 2018.

[46] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition, 2019.

[47] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions, 2018.

[48] SpeechBrain. speechbrain/sepformer-whamr16k · hugging face — huggingface.co. `https://huggingface.co/speechbrain/sepformer-whamr16k`.

[49] Cem Subakan, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong. Attention is all you need in speech separation, 2021.

[50] Hemlata Tak, Jose Patino, Massimiliano Todisco, Andreas Nautsch, Nicholas Evans, and Anthony Larcher. End-to-end anti-spoofing with rawnet2, 2021.

[51] Hemlata Tak, Massimiliano Todisco, Xin Wang, Jee weon Jung, Junichi Yamagishi, and Nicholas Evans. Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation, 2022.

[52] Cristian Vaccari and Andrew Chadwick. Deepfakes and disinformation: Exploring the impact of synthetic political video on deception, uncertainty, and trust in news. *Social Media + Society*, 6(1):2056305120903408, 2020.

[53] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio, 2016.

[54] Tomasz Walczyna and Zbigniew Piotrowski. Overview of voice conversion methods based on deep learning. *Applied Sciences*, 13(5), 2023.

[55] Xin Wang, Hector Delgado, Hemlata Tak, Jee weon Jung, Hye jin Shim, Massimiliano Todisco, Ivan Kukanov, Xuechen Liu, Md Sahidullah, Tomi Kinnunen, Nicholas Evans, Kong Aik Lee, and Junichi Yamagishi. Asvspoof 5: Crowdsourced speech data, deepfakes, and adversarial attacks at scale, 2024.

[56] Linqiang Wei, Yanhua Long, Haoran Wei, and Yijie Li. New acoustic features for synthetic and replay spoofing attack detection. *Symmetry*, 14(2), 2022.

[57] Jee weon Jung, Hee-Soo Heo, Ju ho Kim, Hye jin Shim, and Ha-Jin Yu. Rawnet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification, 2019.

[58] Haibin Wu, Jiawen Kang, Lingwei Meng, Helen Meng, and Hung yi Lee. The defender's perspective on automatic speaker verification: An overview, 2023.

[59] Haojie Wu, Pan Hui, and Pengyuan Zhou. Deepfake in the metaverse: An outlook survey, 2023.

[60] Zhenzong Wu, Rohan Kumar Das, Jichen Yang, and Haizhou Li. Light convolutional neural network with feature genuinization for detection of synthetic speech attacks, 2020.

[61] Amit Kumar Singh Yadav, Kratika Bhagtani, Davide Salvi, Paolo Bestagini, and Edward J Delp. FairSSD: Understanding Bias in Synthetic Speech Detectors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[62] Junichi Yamagishi, Christophe Veaux, Simon King, and Steve Renals. Speech synthesis technologies for individuals with vocal disabilities: Voice banking and reconstruction. *Acoustical Science and Technology*, 33:1–5, 2012.

[63] Jiangyan Yi, Jianhua Tao, Ruibo Fu, Xinrui Yan, Chenglong Wang, Tao Wang, Chu Yuan Zhang, Xiaohui Zhang, Yan Zhao, Yong Ren, Le Xu, Junzuo Zhou, Hao Gu, Zhengqi Wen, Shan Liang, Zheng Lian, Shuai Nie, and Haizhou Li. Add 2023: the second audio deepfake detection challenge, 2023.

[64] Zirui Zhang, Wei Hao, Aroon Sankoh, William Lin, Emanuel Mendiola-Ortiz, Junfeng Yang, and Chengzhi Mao. I can hear you: Selective robust training for deepfake audio detection, 2024.

Abstract in lingua italiana

I recenti miglioramenti nei campi del deep learning e dei modelli generativi hanno reso possibile la creazione di contenuti sintetici sempre più realistici. Con l'aumento dell'accessibilità di queste tecnologie, il loro potenziale abuso crea dei rischi significativi, rendendo imperativo avere la possibilità di distinguere tra contenuto reale e sintetico. Nel campo dell'audio, le tecniche di generazione di speech deepfake consentono la sintesi di segnali che possono imitare la voce di un bersaglio e far sì che pronunci frasi arbitrarie. Le tecniche di rilevamento di speech deepfake si concentrano nell'identificazione di segnali sintetici attraverso l'analisi di artefatti tipicamente assenti nei segnali audio che rappresentano un parlato reale. Tuttavia, questo è un campo in continua evoluzione: i rilevatori devono adattarsi costantemente per contrastare minacce sempre più sofisticate. Con l'adozione diffusa dei metodi di rilevamento, gli utenti malintenzionati hanno iniziato a utilizzare i cosiddetti adversarial attacks, progettati per eludere anche i sistemi più avanzati. Introducendo del rumore quasi impercettibile nei sample considerati, questi attacchi possono far si che i rilevatori falliscano nel processo di identificazione, classsificando il parlato sintetico come genuino e viceversa. Per contrastare l'uso degli adversarial attacks, sono stati presentati nella letteratura dei meccanismi di difesa. Quest'ultimi hanno lo scopo di processare i segnali audio in input allo scopo di mitigare gli effetti degli adversarial attacks. In questa tesi, contribuiremo a questo campo di ricerca e proporremo multiple strategie di difesa. Il nostro approccio sarà incrementale, cominciando con un semplice readdestramento del modello, usando modelli di enfatizzazione del parlato allo stato dell'arte, ed infine adottando una tecnica di refactoring dell'input originariamente sviuppata per l'uso nel campo dell'analisi forense delle immagini. L'analisi di queste tecniche fornirà informazioni preziose, con un potenziale di aumento dell'accuratezza nella classificazione dei rilevatori considerati fino al 45% usando come metrica la balanced accuracy.

Parole chiave: Rilevamento di Deepfake Vocali, Adversarial Attacks, Purificazione del Segnale