*electronics*

MDPI

*Article*

# The Generalization Gap: Do Audio Deepfake Detectors Actually Protect Against Modern Vishing?

**Victoria García Martínez-Echevarría [1], Rafael Palacios [1,2,*], Gregorio López [1] and Amar Gupta [3,4]**

[1] Instituto de Investigación Tecnológica, Universidad Pontificia Comillas, 28015 Madrid, Spain; victoria.garcia@alu.comillas.edu (V.G.M.-E.); gllopez@comillas.edu (G.L.)
[2] Cybersecurity at MIT Sloan (CAMS), Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[3] Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology, Cambridge, MA 02139, USA; agupta@mit.edu
[4] Institute of Applied AI Innovation, The University of Texas at El Paso, El Paso, TX 79968, USA
* Correspondence: rafael.palacios@iit.comillas.edu

## Abstract

Voice phishing, commonly known as vishing, has become one of the fastest-growing threats in social engineering. The rapid advancement and accessibility of AI voice cloning tools have enabled attackers to produce highly convincing synthetic speech at minimal cost, driving a sharp increase in impersonation fraud. Accordingly, automatic detection of synthetic voices could contribute, as one component of a broader defense, to mitigating vishing attacks. This paper studies the automatic detection of AI-generated speech, with a particular focus on how well such detectors generalize beyond their training data to modern, unseen synthesis methods. Two detection approaches are evaluated: a Residual CNN (convolutional neural network) trained as a binary classifier on three different time–frequency representations and a one-class learning strategy with a ResNet-18 backbone, yielding four models in total. Models were trained on the well-known ASVspoof 2019 Logical Access dataset and tested on its standard partitions. Then, models were tested on the SONAR benchmark, which gathers voices generated with state-of-the-art synthesis techniques unseen during training. Experimental results show that, on the modern systems gathered in SONAR, all four configurations fall close to chance. The LFCC one-class detector generalizes comparatively best, but the apparently higher accuracy of some models reflects a tendency to label most speech as spoofed. These findings indicate that the evaluated detectors can provide, at most, a partial security layer against vishing driven by current and emerging speech-synthesis technologies, although continuous model updates are recommended.

**Keywords:** AI-generated speech; spoofing detection; residual CNN (convolutional neural network); one-class learning; generalization; vishing

Check for updates

Academic Editor: Aryya Gangopadhyay

Received: 18 May 2026
Revised: 25 June 2026
Accepted: 26 June 2026
Published: 30 June 2026

**Copyright:** © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

## 1. Introduction

In the second half of 2024, vishing attacks surged 442% relative to the first half of the year, as adversaries increasingly leveraged AI-generated voices to conduct impersonation fraud at scale [1]. The financial sector is especially concerned: Organizations face average annual losses of $14 million per vishing incident [2], and business email compromise attacks that incorporate vishing elements accounted for nearly $2.8 billion reported losses in the United States during 2024 [3]. These figures reflect a structural shift in the threat

landscape, one in which the barrier to executing a convincing voice impersonation has effectively collapsed.

Over the past decade, advances in artificial intelligence have enabled increasingly realistic synthetic content across multiple domains, including text, images, and speech. The study presented in this article is a continuation of previous research related to synthetic media and security in household devices, such as smart personal assistants [4], wearable devices [5], and biometric identification [6,7], and their impact on social networks [8,9]. In particular, modern speech synthesis and voice conversion techniques can now generate audio samples that closely reproduce the characteristics and nuances of human speech [10], supporting applications such as accessibility tools and conversational assistants.

However, as AI-generated audio becomes harder to distinguish from bona fide speech, concerns about authenticity and trust in voice-mediated interactions have intensified. From a security perspective, synthetic voices can be weaponized for impersonation and social-engineering attacks against voice-based services, including biometric authentication and smart personal assistants [11,12]. Real-world case reports have highlighted the potential of voice deepfakes to bypass security measures and undermine confidence in digital transactions [13]. Moreover, recent evidence indicates that criminals may deploy partial deepfakes in real-world scams, deceiving both humans and automated systems [14].

These threats have motivated focused research on audio deepfakes and countermeasures (e.g., Gao, 2022) [15–17] and standardized evaluations such as the ASVspoof challenges [18,19]. Despite substantial progress, robust detection remains challenging in practical settings, particularly when models are confronted with unseen synthesis methods or mismatched audio conditions [20,21].

In this work, we study the automatic detection of AI-generated speech by retraining two reference countermeasures from ASVspoof 2019 Logical Access: a residual CNN trained as a binary classifier [22] and a one-class learning approach based on OC-Softmax [23]. To reduce duration-based shortcuts, we filter the dataset to include only utterances between two and four seconds and evaluate generalization to unseen spoofing techniques, including the modern synthesis systems gathered in the SONAR benchmark.

This article builds directly upon the first author's Bachelor's thesis (*Automatic Detection of AI-Generated Audio*), openly archived in the institutional repository, from which the present study inherits its problem formulation, the choice of reference countermeasures, and part of the experimental infrastructure. That earlier work centered on assembling a detection pipeline and contrasting it with human perception; the focus here is deliberately different, shifting toward generalization. The central research question is therefore not whether a given architecture can detect known attacks—which is already a largely solved problem on in-domain data—but how far detectors trained on a single benchmark (ASVspoof in this case) remain useful once the synthesis landscape moves on.

## 2. Related Work

Synthetic speech detection has emerged as a critical area of research in recent years due to the rapid development of high-quality text-to-speech (TTS) and voice cloning technologies [16,17,24,25]. Modern AI-generated voices can closely replicate the characteristics of human speech, leading to security risks (e.g., spoofing voice authentication systems) and misinformation concerns [26]. In light of this, the speaker recognition community has established shared benchmarks—such as the ASVspoof challenges—to promote the design and deployment of effective countermeasures against synthetic speech attacks. The ASVspoof 2019 challenge, in particular, provided a comprehensive, large-scale dataset of both logical access (LA) attacks (i.e., synthetic or converted speech) and physical access (PA) replay attacks, created with state-of-the-art TTS and voice conversion (VC) systems [27].

Research conducted since around 2017 has increasingly focused on distinguishing bona fide human speech from such AI-generated voices, building on these standardized datasets [28].

Early approaches to synthetic speech detection often relied on compact signal models and handcrafted acoustic features. For example, the inaugural ASVspoof 2015 challenge [18] applied Gaussian mixture models (GMMs) using features like Mel-frequency cepstral coefficients (MFCCs) as baseline detectors. While the methods could recognize some obvious signal distortions, their performance against more sophisticated attacks was limited. Beyond GMM-based baselines, classical machine learning pipelines (including Random Forest and Support Vector Machines, among others) using MFCC descriptors have also been explored for deepfake audio detection [29]. In the ASVspoof 2019 Logical Access track, a GMM with linear frequency cepstral coefficients (LFCCs) achieved an Equal Error Rate (EER) of 13.54%, and even a stronger CQCC-GMM baseline (using constant-Q cepstral coefficients) only reached an EER of 11.04% [30]. Such results showcased the need for more powerful classifiers and richer feature representations as synthetic voices became more realistic. Over the past few years, there has been a dramatic shift in this field towards deep learning-based methods that significantly outperform traditional approaches [24].

Modern state-of-the-art systems predominantly use deep neural networks (DNNs) to automatically learn discriminative speech representations—although spectrogram-based machine learning approaches have also been proposed as lightweight alternatives for deepfake audio detection [31]. Convolutional neural networks (CNNs) are widely used to process spectrograms and other time–frequency features, capturing subtle patterns that can differentiate between real and synthetic speech better than handcrafted features, and recent works have proposed efficient residual CNN variants that combine max-feature-map activations with depthwise separable convolutions [32]. Indeed, the first adoptions of deep models already showed promise: A CNN-RNN hybrid model by Zhang et al. (2017) [33] achieved then-best results on the ASVspoof 2015 dataset by combining convolutional feature extraction with recurrent layers for temporal modeling. Later studies explored more complex architectures, including residual networks (ResNets) or lightweight CNNs with gating and recurrent units. An example of the latter is the Light Convolutional GRU-based RNN (Recurrent Neural Network) proposed by Gómez-Alanis et al. (2019) [34], which achieved an EER of 6.28% on the ASVspoof 2019 LA evaluation set. Compared to GMM baselines, these deep models demonstrated a significant performance boost, with EERs dropping about an order of magnitude in known scenarios.

One notable line of research has focused on one-class learning approaches and anomaly detection techniques. Instead of training a binary classifier on real and synthetic samples, these methods only learn the characteristics of real (bona fide) speech, aiming to detect spoofed speech as deviations from this learned distribution. Zhang et al. (2020) [23] introduced a one-class learning framework that uses a ResNet-18 backbone and a novel loss function (One-Class Softmax) to concentrate the embeddings of real speech while pushing away those of synthetic samples. This approach excelled at detecting unseen attack types: Without any data augmentation, it achieved an EER of 2.19% on the ASVspoof 2019 LA evaluation set, surpassing all prior single-model systems on this benchmark. Other recent architectures have integrated innovative network modules, such as attention mechanisms [35] or channel-wise gated Res2Nets [36], to further enhance detection performance. These studies demonstrate the community's trend towards specialized deep architectures (often ensembles of CNNs, ResNets, LSTMs, etc.) [37] finely tuned for identifying synthetic speech patterns.

The choice of input features is also a key factor that has evolved over time. Classic low-level representations such as Mel-frequency cepstral coefficients (MFCCs), linear predictive cepstral coefficients (LPCCs), or spectral centroid features were common in early

systems [38]. In later challenges, more effective features were introduced to target the specific characteristics of synthetic audio. The ASVspoof 2019 baseline models, for instance, integrated two other types of cepstral coefficients, constant-Q cepstral coefficients (CQCCs) and linear-frequency cepstral coefficients (LFCCs), paired with GMM classifiers [27]. Then, as deep learning gained traction, many systems began to feed raw audio waveforms or spectrograms directly into neural networks. Time–frequency representations of high resolution, such as Short-Time Fourier Transform (STFT) magnitude spectrograms, Mel-spectrograms, and Constant Q Transform (CQT) spectrograms [39], allow CNN-based models to learn the relevant feature filters automatically. Overall, the field has seen a shift from handcrafted features to automated feature learning, though hybrid approaches (combining multiple types) are still common.

Multiple datasets have been released to support research and benchmarking in this area. The ASVspoof series provides the foundational collection: ASVspoof 2015 [18] focused on common voice conversion and text-to-speech attacks of that era, while ASVspoof 2019 [19] greatly expanded the scale and diversity of attacks by adding new techniques. In particular, the ASVspoof 2019 Logical Access (LA) partition includes spoofed and bona fide utterances from 107 speakers (male and female) across a wide range of both TTS and VC algorithms, all derived from the Voice Cloning Toolkit (VCTK) corpus [40]. Beyond the ASVspoof data, other datasets have aimed to cover different languages and more diverse generation methods. The Fake-or-Real (FoR) dataset [41] is one remarkable example: It contains over 87,000 synthetic samples generated by a variety of modern TTS systems and more than 110,000 genuine utterances. This collection includes highly natural-sounding fakes (often indistinguishable by humans) and is sufficiently large to train complex deep models (e.g., an Inception-v3 CNN). Also, the CFAD (Chinese Fake Audio Detection) dataset [42] provides a Mandarin Chinese benchmark with 12 different audio generation methods, and its audio samples include further augmentations with real-world degradations like background noise and reverberation. More recently, Li et al. [24] introduced SONAR, an evaluation framework sourced from nine diverse TTS providers, which span both commercial APIs and state-of-the-art open-source models, and it unifies traditional and foundation model-based approaches under a common evaluation protocol. Their results showed that while existing detectors perform well on established benchmarks, they suffer substantial accuracy drops when confronted with the most advanced synthesis systems, reinforcing the need for continuously updated evaluation protocols.

Beyond the challenge of detecting unseen spoofing techniques, further research on AI-based cybersecurity systems highlights a fundamental tension between robustness and resource efficiency. Moskalenko et al. [43] proposed a model architecture and training method for resource-constrained AI systems that integrates dynamic neural networks with resilience-aware meta-learning, achieving simultaneous improvements in robustness to adversarial attacks, fault injections, and concept drift while reducing computational cost by over 30%. Their framework explicitly addresses the distribution shift problem, which is a concern closely related to the generalization gap observed in synthetic speech detectors when confronted with synthesis architectures unseen during training. This perspective suggests that improving countermeasure generalization may require not only richer feature representations or larger training sets but also training strategies explicitly designed to anticipate and absorb domain perturbations.

Another important aspect in the literature is the evaluation of human ability to detect synthetic speech. Surprisingly, even as algorithmic detectors have improved, studies show that human listeners often struggle to detect AI-generated voices. For instance, a recent experiment with over 500 participants found that humans correctly identified deepfake speech only about 73% of the time on average [44]. This was true even when comparing

English and Mandarin speech, and providing a brief training exposure to deepfakes only resulted in a slight improvement. Namely, high-quality synthetic voices can fool humans almost one out of four times. Moreover, partial fake speech (i.e., where only a few words are synthetically replaced) has been shown to be even harder to detect: human listeners identified it correctly only about 17% of the time, and existing speaker recognition systems were deceived with success rates of up to 97% [14].

Concurrently, work on authentication robustness has shown that the problem extends beyond human perception: Hong et al. [21] demonstrated that state-of-the-art speaker verification systems can be easily bypassed by open-source voice cloning models trained on as little as ten minutes of target speech. However, they also found that anti-spoofing detectors trained in-domain fail to generalize to unseen synthesis architectures, exposing a critical gap between benchmark performance and real-world security.

## 3. Proposed Methodology

To investigate the automatic detection of AI-generated audio under controlled conditions, this work adopts and retrains reference models from the ASVspoof 2019 Challenge, focusing on the Logical Access (LA) partition of the dataset. In particular, the data preprocessing stage was modified to filter the dataset and only include audio samples with durations between two and four seconds. This constraint was introduced to mitigate potential biases linked to audio length and to ensure that the models learn to discriminate based exclusively on acoustic properties. A key decision was to compare four distinct types of audio features—spectrograms, MFCCs, CQCCs, and LFCCs—each tested independently to isolate their influence on classification performance.

### *3.1. Technical Overview*

The automatic detection pipeline begins with data preprocessing, followed by feature extraction to obtain substantial audio representations. Each representation is then passed to a classification model trained to distinguish between real and synthetic speech. This work considers four different types of audio features, each obtained from the same audio samples: spectrograms, Mel-frequency cepstral coefficients (MFCCs), linear frequency cepstral coefficients (LFCCs), and constant-Q cepstral coefficients (CQCCs). These features differ in their signal representation approach, frequency resolution, noise sensitivity, and capacity to capture synthetic speech artifacts.

The spectrograms used in this project correspond to the logarithmic magnitude of the Short-Time Fourier Transform (STFT) of the audio signal. Specifically, the STFT is computed using Hamming windows of size 2048 with 25% overlap [45]. The magnitude of each frequency component is then calculated and converted to the logarithmic scale. This results in a time–frequency matrix that captures the detailed spectral characteristics of the input waveform. Unlike handcrafted features (such as MFCCs or CQCCs), this format is relatively raw; however, it exploits the representation learning capability of deep neural networks, which can learn higher-level features directly from the spectrogram within their hidden layers [22].

Mel-frequency cepstral coefficients (MFCCs) are computed by first applying the STFT to the audio signal, and the obtained frequency spectrum is then mapped—through a filter bank—onto a Mel scale that approximates the way humans perceive sound. Finally, a discrete cosine transform (DCT) is applied to decorrelate the resulting coefficients [46]. In this work, the first 24 coefficients are extracted to represent each audio frame [22]. MFCCs yield a compact representation that highlights relevant frequency bands, although they may be less sensitive to the subtle distortions introduced by advanced speech synthesis techniques, which can reduce their effectiveness under noisy or varying channel conditions.

Constant-Q cepstral coefficients are based on the Constant-Q Transform (CQT), which uses geometrically spaced frequency bins instead of the regularly spaced bins used by the STFT. This method provides higher frequency resolutions at lower frequencies and higher temporal resolutions at higher frequencies, aligning well with the human perception of pitch and timbre. To compute CQCCs, the CQT is first applied to the audio signal, followed by the calculation of a power spectrum and its conversion to the logarithmic scale. Next, uniform resampling is performed, and finally, a DCT is applied to produce the cepstral coefficients [22]. CQCCs have been shown to outperform traditional cepstral coefficients in multiple spoofing detection challenges by successfully highlighting synthesis artifacts that appear both in low and high frequencies [47,48].

Linear frequency cepstral coefficients (LFCCs) share a similar computational process with MFCCs but differ mainly in the use of a linear frequency scale rather than the Mel scale. This linear spacing ensures uniform resolution across the entire frequency spectrum, retaining high-frequency details that may contain discriminative cues for synthetic audio. LFCCs have demonstrated stronger performance in spoofing detection tasks, particularly against voice conversion attacks that introduce anomalies in the high-frequency range [48].

Each of these feature types offers specific advantages and trade-offs. Spectrograms retain rich temporal dynamics and detailed spectral information but require more computational resources and larger training datasets. MFCCs and LFCCs offer compact representations with varying frequency resolutions, balancing efficiency and classification performance. CQCCs stand out for their frequency scaling similar to human perception and their ability to detect subtle artifacts.

### *3.2. Model Design*

This work implemented and compared two distinct models for synthetic speech detection, both inspired by high-ranking systems in the ASVspoof 2019 Challenge. Each model uses different input features and architectural designs, enabling an assessment of how different representations and learning approaches affect generalization to unseen spoofing techniques.

The first model is based on the system proposed by Alzantot et al. (2019) [22] for the ASVspoof 2019 competition. It employs deep residual convolutional neural networks (ResNets) trained on three types of input features—spectrograms, MFCCs, and CQCCs—which are represented as two-dimensional matrices (along time and frequency axes), reshaped as image-like tensors, and processed through a stack of residual blocks. Each residual block (see complete structure in Figure 1) contains convolutional layers followed by normalizations (batch for spectrograms and group for MFCCs and CQCCs), leaky-ReLU activations, and dropout layers (to avoid overfitting), together with skip connections that facilitate gradient flow during training and prevent issues like vanishing gradients.

Before the final classification layer (see global architecture in Figure 2), the output of the last residual block is flattened and fed directly to two fully connected layers that produce the final binary prediction, indicating whether the input corresponds to human or spoofed speech. During training, the model minimizes the standard cross-entropy loss between the predicted class labels (bona fide vs. spoof) and the ground truth.

While the three variants of this model (trained on spectrograms, MFCCs, and CQCCs) share an almost identical structure, they differ in the shape of their input tensors due to the specific dimensionality of each feature type. As a result, the dimensionality of the first fully connected layer varies according to the input feature type.

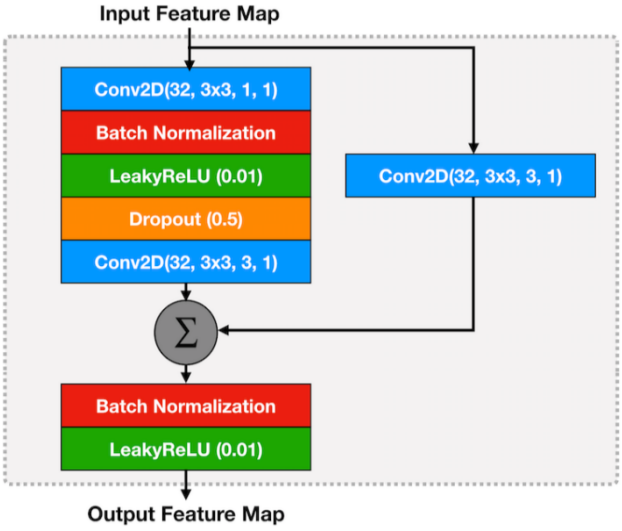


**Figure 1.** Detailed architecture of each residual block [22].

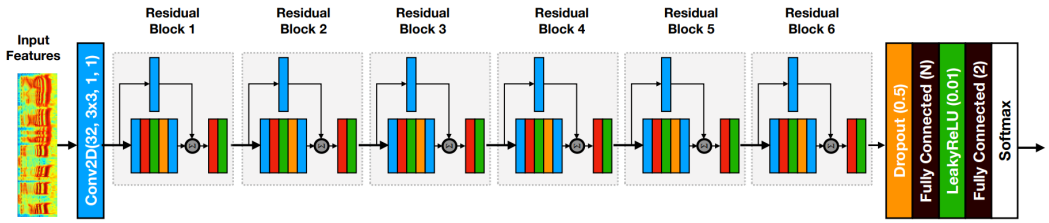


**Figure 2.** Model architecture proposed by Alzantot et al. [22].

The second model follows the one-class learning approach presented by Zhang et al. (2020) [23]. Instead of treating spoof detection as a binary classification problem, this method focuses exclusively on modeling the bona fide class and seeks to separate unseen spoofing attacks in the embedding space. This is achieved by applying a ResNet-18 backbone to LFCC audio features as input. The model is trained with an innovative OC-Softmax (One Class Softmax) loss function—also proposed in [23]—which encourages compact clustering of bona fide embeddings while simultaneously pushing away potential spoofed samples at inference time. Figure 3 illustrates this by showing the embedding space and how clusters are formed when applying different loss functions.

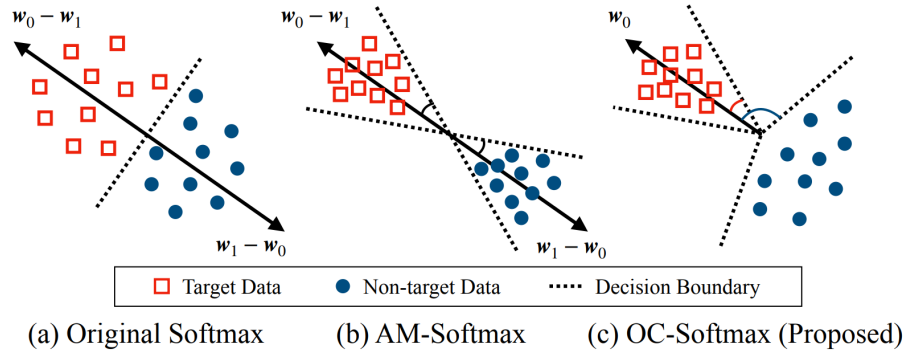


**Figure 3.** Embedding space division applying three different loss functions [23].

The OC-Softmax loss function applies a cosine-based projection of embeddings onto a unit hypersphere, introducing an angular margin to penalize deviations from the class center. This makes the system especially robust against unseen attack types, which is a truly beneficial property for the ASVspoof evaluation set, as it includes spoofed samples generated with algorithms not present in the training set.

All models were trained and tested on a subset of the ASVspoof 2019 Logical Access dataset, containing audio clips between 2 and 4 s long. Evaluation was performed on the corresponding filtered evaluation set to ensure consistency in input length.

## 4. Implementation and Experiments

The experiments in this project consisted of retraining spoofing detection models on a subset of the ASVspoof 2019 Logical Access dataset and then evaluating them on both the development and evaluation partitions of said subset, as well as on an independent set of external audio samples generated by third-party tools. The main goal was to assess the models' generalization capacity to classify unseen spoofing techniques.

### *4.1. Dataset*

The ASVspoof 2019 challenge is a broadly adopted benchmark for synthetic speech detection. Its Logical Access (LA) set has a large collection of both real (bona fide) and spoofed audio recordings. The latter are generated using a variety of advanced text-to-speech (TTS) and voice conversion (VC) techniques, which makes it a suitable dataset for training and evaluating spoofing detection models. The data is split into three partitions—training, development, and evaluation—and each of them contains a diverse set of speakers, balanced between male and female participants [27].

A key preprocessing decision in this work was to filter the dataset to only include audio clips with durations between two and four seconds. This choice was intended to reduce the risk that models attend to superficial temporal or length-based features rather than focusing on the acoustic properties of the audio samples. Although the absolute number of samples in each partition is reduced, the class balance of the retained data stays close to that of the full dataset (Table 1).

**Table 1.** Counts and ratios (SP-BF) before and after filtering the data by duration.

| Set | Class | Original Count | Original % | Filtered Count | Filtered % |
|---|---|---|---|---|---|
| train | SP | 22,800 | 89.83% | 13,208 | 87.63% |
| | BF | 2580 | 10.17% | 1865 | 12.37% |
| dev | SP | 22,296 | 89.74% | 12,350 | 88.2% |
| | BF | 2548 | 10.26% | 1653 | 11.8% |
| eval | SP | 63,882 | 89.68% | 33,401 | 86.17% |
| | BF | 7355 | 10.32% | 5360 | 13.83% |

The original dataset presents a wide range of utterance lengths, with some samples extending beyond 10 s in some cases and a notable skew towards shorter durations. Figure 4 shows the distribution of audio lengths by class for each partition of the dataset, both before and after filtering. Once the filtering is applied, the distribution of audio lengths becomes more uniform and tightly constrained within the targeted range, effectively removing outliers and reducing variance.

Beyond making the length distribution more uniform, restricting the data to the 2–4 s range serves a second, more important purpose, which becomes clear when looking at how the two classes are distributed across durations rather than at the aggregate counts. In the training partition, the shortest and longest clips are almost entirely spoofed: Around 96.7% of the clips under two seconds and 94.9% of those between five and ten seconds belong to the spoofed class, so duration alone becomes a strong cue for the class a clip belongs to. Within the 2–4 s band, by contrast, the proportion of spoofed clips (about 87.6%) is considerably more stable and close to the global ratio of the full dataset. Keeping only this band therefore preserves a class balance comparable to the one used in the original

work—where the dataset is taken as a whole, without accounting for possible duration biases—so that the results remain comparable while preventing the models from using duration as a shortcut to infer the class instead of attending to the acoustic properties of the audio.

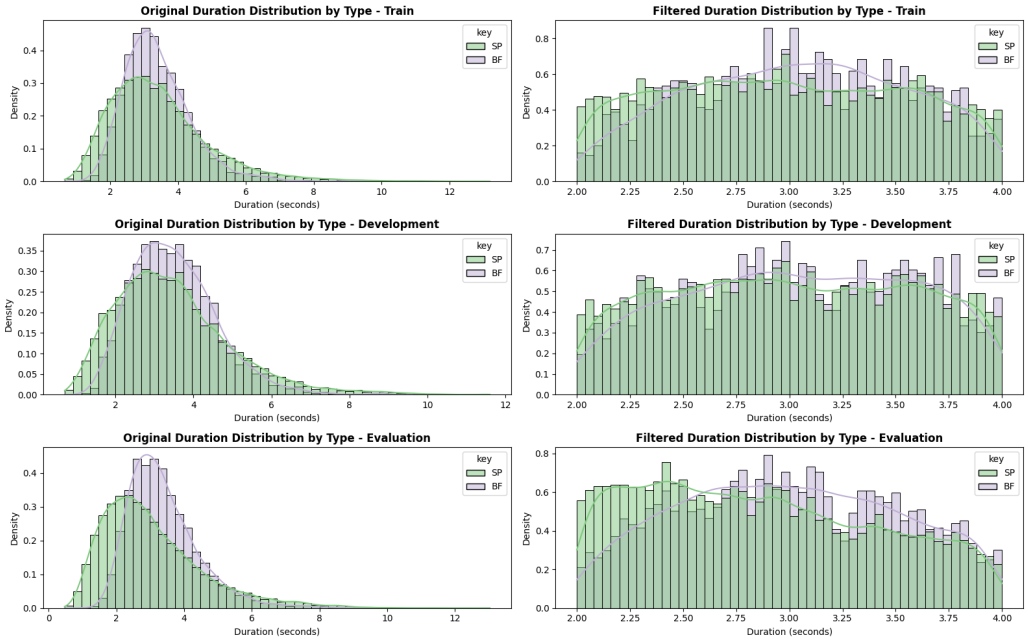


**Figure 4.** Distribution of audio lengths by class for each partition before and after filtering.

In addition to the ASVspoof 2019 data, this work evaluates the trained models on the external SONAR benchmark [24], a recent framework designed to test deepfake speech detectors against modern synthesis. SONAR gathers spoofed speech from nine diverse state-of-the-art text-to-speech (TTS) systems—including commercial APIs and open-source/foundation-model generators such as VALL-E, Seed-TTS, VoiceBox, NaturalSpeech3, xTTS, FlashSpeech, PromptTTS2, AudioGen, and OpenAI TTS—alongside bona fide human recordings. This set contains 9077 clips (2274 bona fide and 6803 spoofed). Because every generator in SONAR post-dates the ASVspoof 2019 attacks and was unseen during training, this set provides a far stronger test of generalization to current and emerging synthesis than in-domain evaluation. All clips are processed through the same preprocessing, feature-extraction, and scoring pipeline used for ASVspoof 2019 so that the protocol, metrics, and output format remain identical across sets.

### *4.2. Setup and Configuration*

The preprocessing and data handling stages were implemented in separate modules corresponding to each model architecture. In both cases, these scripts perform loading, batching, and basic audio transformations in order to prepare inputs to be fed into the models.

Feature extraction was performed externally using MATLAB R2023b scripts—specifically for CQCCs and LFCCs—which involved computationally intensive processes and resulted in large `.mat` files. This approach aligns with the original papers' methodologies and ensures accurate calculations of these specialized features. One advantage is that preprocessing audio samples can be done using parallel processing techniques [49].

The values of the training hyperparameters were set to those recommended as optimal by the original authors of the models. For the first model, Alzantot et al. (2019) [22] suggested using a learning rate of 0.0001 and a batch size of 32, while Zhang et al. (2020) [23] recommended a learning rate of 0.0003 (with 50% decay every 10 epochs) and a batch size of 64 for the second model. In terms of experimentation, the only variable that was modified

was the number of training epochs, as multiple versions of each model were trained to observe their performance differences and identify the best-performing configurations.

In some cases, early stopping was implemented with a considerably small threshold ($\epsilon = 1 \times 10^{-100}$). For instance, the MFCC model trained for 91 epochs (see Section 4.3) was stopped early due to this condition. However, the internal configurations and structures of the models—such as optimizers, schedulers, and other architectural details—were kept consistent with the original designs, ensuring that any differences in performance could be attributed to the number of epochs rather than to changes in model setup.

### *4.3. Training and Validation*

For each feature type (spectrograms, MFCCs, CQCCs, and LFCCs), several models were trained with different numbers of epochs in order to identify the most effective configurations. All models were trained on a DGX server using an NVIDIA H200 Tensor Core GPU with 141 GB of high-bandwidth memory. The resulting training times, along with the remaining computational-cost figures, are reported in Section 4.5.

As mentioned above, the proposed training pipelines were minimally modified to adapt to the filtered dataset, keeping the original architectures and hyperparameters as close as possible to the ones described in the papers. In terms of loss functions, the first model used a weighted cross-entropy loss to compensate for the class imbalance in the training data. Thus, the ratio of weights assigned to the bona fide and spoofed classes was 9:1, respectively [22], given that the training set contains approximately 90% spoofed samples and 10% bona fide samples (see exact percentages in Table 1). For the second model, the tailored OC-Softmax loss function was used, which focuses only on the real speech distribution and aims to separate unseen spoofing attacks in the embedding space [23].

In terms of evaluation, the primary performance metrics considered were accuracy and Equal Error Rate (EER). The former measures the fraction of correctly classified samples, while the latter is calculated by identifying the point where the false acceptance rate (FAR) equals the false rejection rate (FRR). These metrics are commonly used in spoofing detection tasks and together provide a comprehensive view of the models' performance across both classes. It should be noted, however, that the t-DCF (tandem-detection cost function) [50] suggested by the ASVspoof 2019 organizers was not used in this project. The t-DCF metric, designed for speaker verification tasks, incorporates factors like user identity, which are not relevant to this work's focus on detecting whether an audio is real or spoofed.

The models performed significantly well on the training and development sets; notwithstanding, performance on the evaluation set dropped, as discussed in the next section.

### *4.4. Performance Analysis*

The performance of the models was first analyzed in-domain on the training, development, and evaluation partitions of ASVspoof 2019. These in-domain accuracies are the basis on which a representative version of each feature type was selected for the detailed analysis of Section 5. The metrics considered in this section are accuracy—to measure the proportion of correctly classified instances—and the Equal Error Rate (EER)—to compare the performance against the original models.

The first analysis is based on the performance of the models on the training and development sets. As expected, the accuracy is higher in the training set than in the development set for all models (Figure 5). However, for most models, the difference in accuracy between the two sets is barely noticeable (see numerical values in Table 2), except when using MFCCs as input features. Additionally, it appears that increasing the number of epochs does not notably affect accuracy, indicating that the model is not learning more with additional epochs (nor is it overfitting). But again, this trend does not hold for

the models trained on MFCCs, where increasing the number of epochs actually results in a drop in development accuracy.

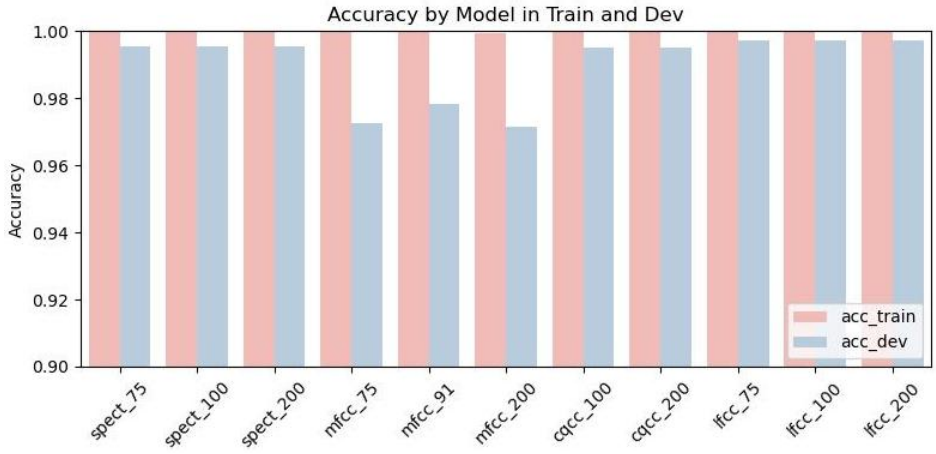


**Figure 5.** Accuracy bar plot of each model on the training and development sets.

The perfect training accuracy can be attributed to the models' ability to memorize the underlying details of the training samples. This suggests that, after a certain point, the models are not learning new patterns but rather focusing on the characteristics of the training set.

**Table 2.** Accuracy and EER values of each model on the training and development sets.

| Model | Train Accuracy | Dev Accuracy | Dev EER | Original Dev EER |
|---|---|---|---|---|
| spect_75 | 1.000000 | 0.995358 | 0.004344 | 0.0011 |
| spect_100 | 1.000000 | 0.995358 | 0.004263 | 0.0011 |
| spect_200 | 1.000000 | **0.995430** | 0.002929 | 0.0011 |
| mfcc_75 | 0.999668 | 0.972577 | 0.043822 | 0.0334 |
| mfcc_91 | 0.999602 | **0.978290** | 0.040631 | 0.0334 |
| mfcc_200 | 0.999536 | 0.971649 | 0.090397 | 0.0334 |
| cqcc_100 | 0.999934 | 0.995001 | 0.006728 | 0.0001 |
| cqcc_200 | 1.000000 | **0.995215** | 0.008567 | 0.0001 |
| lfcc_75 | 1.000000 | 0.997143 | 0.007395 | 0.0020 |
| lfcc_100 | 1.000000 | **0.997286** | 0.005475 | 0.0020 |
| lfcc_200 | 1.000000 | 0.997143 | 0.007314 | 0.0020 |

**Bold** values indicate the best development accuracy within each feature type group.

The EERs calculated for each model (shown in Table 2) are generally higher than those reported in the original papers for the development stage. This is not surprising given the reduced dataset size after the filtering process (training on a smaller dataset may result in less specialized models). Furthermore, it is worth noting that the original models might have indirectly incorporated the duration of the audio samples as a feature, which could have contributed positively to their performance on the original dataset. While it is difficult to determine the exact impact of this hypothesis, it is a plausible factor to consider.

The evaluation set, which contains spoofed audio generated using techniques different from those used to generate the training set, shows a decrease in accuracy for all models (Table 3). The spectrogram-based models suffer the greatest drop in performance, while the LFCC model maintains the highest accuracy. This suggests a better generalization ability of the one-class approach in handling spoofing techniques not seen during training.

**Table 3.** Accuracy and EER values of each model on the evaluation set.

| Model | Eval Accuracy | EER | Original Eval EER |
|---|---|---|---|
| spect_75 | 0.788447 | 0.139396 | 0.0968 |
| spect_100 | **0.797348** | 0.125166 | 0.0968 |
| spect_200 | 0.780940 | 0.137352 | 0.0968 |
| mfcc_75 | 0.842393 | 0.147221 | 0.0933 |
| mfcc_91 | 0.832538 | 0.133189 | 0.0933 |
| mfcc_200 | **0.843941** | 0.136575 | 0.0933 |
| cqcc_100 | **0.833338** | 0.104786 | 0.0769 |
| cqcc_200 | 0.757462 | 0.119310 | 0.0769 |
| lfcc_75 | **0.934573** | 0.036393 | 0.0219 |
| lfcc_100 | 0.931426 | 0.035325 | 0.0219 |
| lfcc_200 | 0.932948 | 0.034899 | 0.0219 |

Green cells indicate models that outperform the all-spoof baseline (≈86% accuracy); yellow cells indicate models that do not. **Bold** values indicate the best evaluation accuracy within each feature type group.

When comparing model performance to a simplistic baseline model that classifies all audio samples as spoofed, the models shaded in yellow in Table 3 perform worse than the baseline, which would achieve an accuracy of around 86% due to the class distribution in the evaluation set (see Table 1). The only model that surpasses this baseline is the LFCC model (shaded in green) in all three versions (75, 100, and 200 epochs).

The EERs of the models on the evaluation set are also higher than those reported in the original papers, following the same trend observed in the development set.

Taken together, these in-domain results are what guided the selection of a single representative version per feature type, which is the basis for the comparison carried out in the next section. The selection relied primarily on accuracy rather than on the EER, because in our experiments, the EER proved to be considerably more sensitive to the number of training epochs. For instance, the development the EER of the MFCC model rises from 0.041 at 91 epochs to 0.090 at 200 epochs—more than doubling—whereas its development accuracy stays within a 0.7-point band (0.972–0.978) over the same range (Table 2). As accuracy offered a more stable and reproducible criterion for comparing configurations, it was adopted as the main basis for choosing, for each feature type, the version that best balances performance and training cost. Accordingly, the selected versions are as follows: 100 epochs for the spectrogram-based model, 75 epochs for MFCCs, 100 epochs for CQCCs, and 75 epochs for LFCCs.

### *4.5. Computational Cost*

Since the system is motivated by a practical, potentially real-time security setting, the four selected models were assessed according to criteria relevant to deployment feasibility: parameter count, multiply–accumulate operations (expressed as GFLOPs per clip), per-clip inference latency and throughput on GPU, model size on disk, the real-time factor (RTF), and total training time. Table 4 summarizes these metrics. All measurements were obtained on the same DGX server used for training (NVIDIA H200 Tensor Core GPU with 141 GB of high-bandwidth memory; see Section 4.3).

The three residual classifiers are extremely lightweight (around 0.3 M parameters, at most 3.4 GFLOPs per clip, roughly 1 ms per clip on GPU, and about 1 MB on disk), whereas the LFCC-based one-class model is substantially larger (12.5 M parameters, 15.8 GFLOPs per clip, 47.6 MB) owing to its ResNet-18 backbone. Despite this, all four models run far faster than real time (RTF $\leq 3 \times 10^{-4}$), so inference latency is not a bottleneck for any of them. Two caveats apply when comparing the RTF across the two families: The

LFCC clips cover approximately 7.5 s of audio, compared with 4 s for the binary models, and the reported LFCC latency includes only the neural backbone, excluding the external MATLAB-based feature extraction step required by the CQCC and LFCC pipelines. This step would dominate end-to-end latency in a real deployment, and it should therefore be considered when comparing feasibility against the spectrogram and MFCC pipelines, which compute their features on the fly.

**Table 4.** Computational cost of the four selected models. The RTF is the inference time divided by the audio duration; values below 1 indicate faster-than-real-time processing.

| Model | Params (M) | GFLOPs/Clip | Latency (ms) | Throughput (Clips/s) | Disk (MB) | RTF | Train Time |
|---|---|---|---|---|---|---|---|
| spect_100 | 0.316 | 3.350 | 1.157 | 864 | 1.24 | 0.00029 | 26.8 min |
| mfcc_75 | 0.256 | 0.235 | 1.066 | 938 | 1.01 | 0.00027 | 4.5 min |
| cqcc_100 | 0.311 | 1.084 | 0.759 | 1318 | 1.23 | 0.00019 | 10.4 min |
| lfcc_75 | 12.450 | 15.765 | 1.150 | 870 | 47.6 | 0.00015 | 21.1 min |

## 5. Results

This section analyzes in detail the behavior of the four selected models—one representative version per feature type—across the four data splits, first comparing their accuracy, then their full set of metrics, and finally their behavior on each individual spoofing technique.

Figure 6 compares the accuracy of these four models across the training, development, and evaluation partitions of ASVspoof 2019 and the external SONAR benchmark.

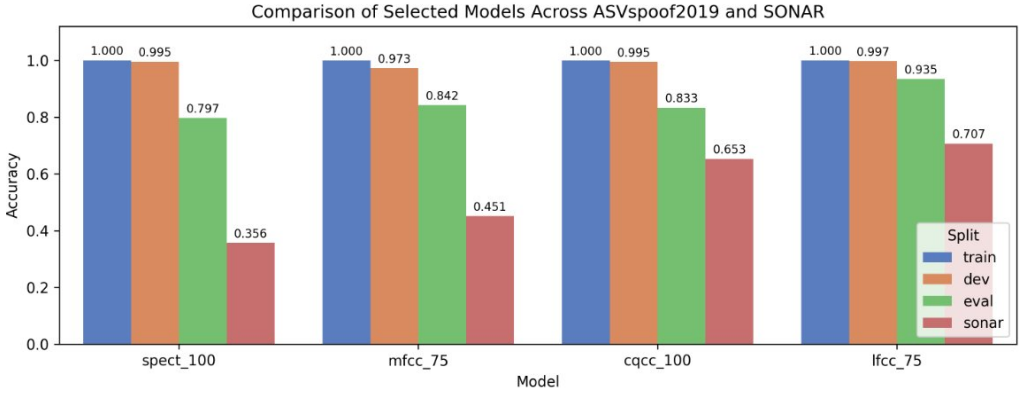


**Figure 6.** Accuracy of the four selected models across the ASVspoof 2019 partitions (train, development, and evaluation) and the external SONAR benchmark.

The figure shows a progressive loss of performance as the evaluation setting becomes more demanding. All four models memorize the training set almost perfectly (accuracy $\approx 1.0$) and maintain very high accuracy on the development set (0.97–1.0). On the evaluation set, which introduces spoofing techniques unseen during training, accuracy drops to between 0.80 (spectrogram) and 0.94 (LFCC), with the one-class LFCC model clearly being the most robust. On SONAR, accuracy falls much further (to 0.36–0.71), but these values should not be interpreted in isolation, especially given the class imbalance in this benchmark. Across every split, the ranking is preserved, with the LFCC model ahead and the spectrogram model behind.

To look beyond raw accuracy, Tables 5 and 6 report a fuller set of metrics, each with its 95% confidence interval written as a $\pm$margin. The balanced accuracy averages the per-class accuracies and is therefore insensitive to the class imbalance; the F1 score summarizes precision and recall on the minority bona fide class; the AUC measures ranking quality independently of the decision threshold; the EER is the operating point where false-

acceptance and false-rejection rates coincide; APCER and BPCER (reported for SONAR) are the error rates on spoofed and bona fide clips, respectively.

**Table 5.** Metrics with 95% confidence interval (±margin) on the evaluation set for the four selected models. Intervals are Wilson score (accuracy) and percentile bootstrap (remaining metrics).

| Model | Accuracy | Balanced Acc. | F1 | AUC | EER |
|---|---|---|---|---|---|
| spect_100 | 0.797 ± 0.004 | 0.872 ± 0.003 | 0.571 ± 0.005 | 0.950 ± 0.003 | 0.125 ± 0.004 |
| mfcc_75 | 0.842 ± 0.004 | 0.857 ± 0.005 | 0.606 ± 0.007 | 0.921 ± 0.003 | 0.147 ± 0.005 |
| cqcc_100 | 0.833 ± 0.004 | 0.892 ± 0.003 | 0.618 ± 0.006 | 0.957 ± 0.002 | 0.105 ± 0.004 |
| lfcc_75 | **0.935 ± 0.003** | **0.958 ± 0.002** | **0.807 ± 0.006** | **0.990 ± 0.001** | **0.036 ± 0.002** |

**Bold** values indicate the best result across all models.

On the evaluation set (Table 5), the confidence intervals are very narrow, largely because this partition contains 38,761 clips. Therefore, the observed differences between models are unlikely to be explained by sampling noise. Indeed, all pairwise differences are statistically significant after applying McNemar's test with Holm correction ($p < 10^{-5}$). The LFCC model achieves the best performance across all metrics and is the only configuration for which its accuracy is significantly above the all-spoof baseline of 0.862 (recall class proportions per split from Table 1); the other three models fall significantly below this reference value. However, this sub-baseline accuracy should not be interpreted as a collapse into the trivial "always-spoof" rule. Their balanced accuracy remains above 0.85, indicating that they still exploit acoustic evidence to distinguish between classes. Instead, the low raw accuracy reflects errors on the novel attacks included in this partition, an effect that is partly hidden by the strong imbalance toward spoofed samples. The gap between accuracy and F1 (e.g., 0.833 versus 0.618 for CQCC) likewise reflects a high error rate on the minority bona fide class.

**Table 6.** Metrics with 95% confidence interval (±margin) on the external SONAR benchmark. APCER and BPCER are the error rates on spoofed and bona fide clips, respectively.

| Model | Accuracy | Balanced Acc. | AUC | APCER | BPCER |
|---|---|---|---|---|---|
| spect_100 | 0.356 ± 0.009 | 0.497 ± 0.010 | 0.533 ± 0.014 | 0.785 | 0.221 |
| mfcc_75 | 0.451 ± 0.010 | 0.490 ± 0.011 | 0.500 ± 0.014 | 0.589 | 0.431 |
| cqcc_100 | 0.653 ± 0.008 | 0.491 ± 0.009 | **0.574 ± 0.014** | 0.184 | 0.834 |
| lfcc_75 | **0.707 ± 0.006** | **0.513 ± 0.008** | 0.558 ± 0.014 | **0.098** | **0.875** |

**Bold** values indicate the best result for each metric.

On SONAR (Table 6), the picture changes completely. The balanced accuracy of all four models falls to 0.49–0.51, and the AUC remains between 0.50 and 0.57. In balanced terms, performance on SONAR is close to chance and not practically reliable: None of the models reaches the all-spoof baseline of 0.75. The higher raw accuracy obtained by CQCC (0.653) and LFCC (0.707) is therefore not evidence of reliable detection. Instead, it is mainly caused by a strong tendency to predict "spoof": APCER remains low (0.18 and 0.10), but BPCER is very high (0.83 and 0.88), meaning that most bona fide clips are incorrectly flagged as spoofed.

This collapse is consistent with the nature of SONAR: its spoofed clips are generated by recent zero-shot and codec- or language-model-based systems, including VALL-E, Seed-TTS, VoiceBox, and NaturalSpeech3, for which their artifacts differ from those present in ASVspoof 2019. At the same time, its bona fide recordings also come from a different domain, so the distribution shift affects both classes. Even under this harder setting, the LFCC one-class model still achieves the best accuracy and balanced accuracy (and the lowest APCER), while CQCC obtains the highest AUC. Thus, the two best models differ depending on the metric, but both remain far from usable on modern synthesis.

These trends are summarized by the ROC curves in Figure 7: the area under the curve, between 0.92 and 0.99 on the evaluation set, collapses to 0.50–0.57 on SONAR for every model.

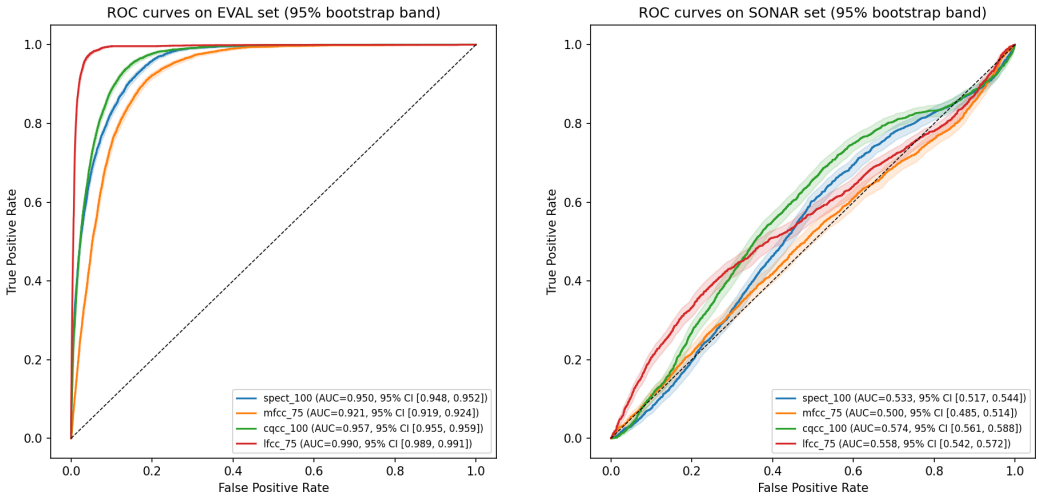


**Figure 7.** ROC curves of the four models on the evaluation set (**left**) and on the external SONAR benchmark (**right**). The dashed diagonal line represents random chance (AUC = 0.5).

Beyond the global metrics, since the spoofed clips were generated with different methods, it was deemed interesting to examine which techniques are hardest for each model to detect. Figure 8 reports the per-model detection accuracy for every spoofing technique in the ASVspoof 2019 evaluation set.

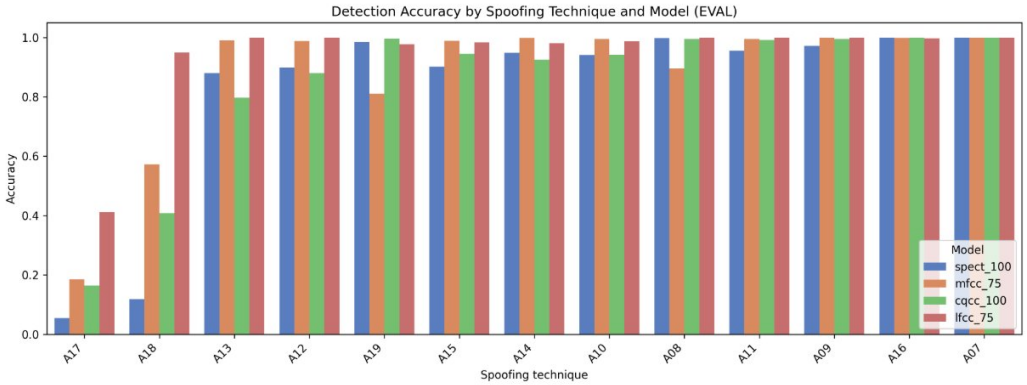


**Figure 8.** Detection accuracy by model for each spoofing technique in the ASVspoof 2019 evaluation set.

Two techniques, A17 and A18, stand out as by far the most difficult: Every model struggles with A17 (from about 0.05 for the spectrogram model to 0.41 for LFCC), and A18 is detected reliably only by LFCC (≈0.95, against 0.12–0.57 for the others), whereas the remaining attacks (A07–A16) are detected almost perfectly by all four. According to the ASVspoof 2019 documentation [27], both A17 and A18 rely on advanced voice-conversion (VC) systems that do not require parallel training data and are considered highly deceptive. A17 is built on a VAE-based voice-conversion pipeline that replaces the traditional vocoder with a generalized direct waveform-modification method [51], rated among the most effective in the Voice Conversion Challenge 2018 for its ability to replicate the spectral structure of natural speech [52]. Conversely, A18 implements a non-parallel framework grounded in i-vector/PLDA (Probabilistic Linear Discriminant Analysis) speaker representation, performing the conversion through a regression-based mapping in the i-vector space [53].

The clear advantage of the LFCC one-class model on these two attacks is consistent with its design: by modelling the bona fide distribution instead of memorizing specific attacks, it flags these unusual conversions as deviations more effectively than the binary classifiers.

The same analysis on SONAR (Figure 9) shows the detection accuracy of each model on the spoofed clips of the nine generation tools in the benchmark.

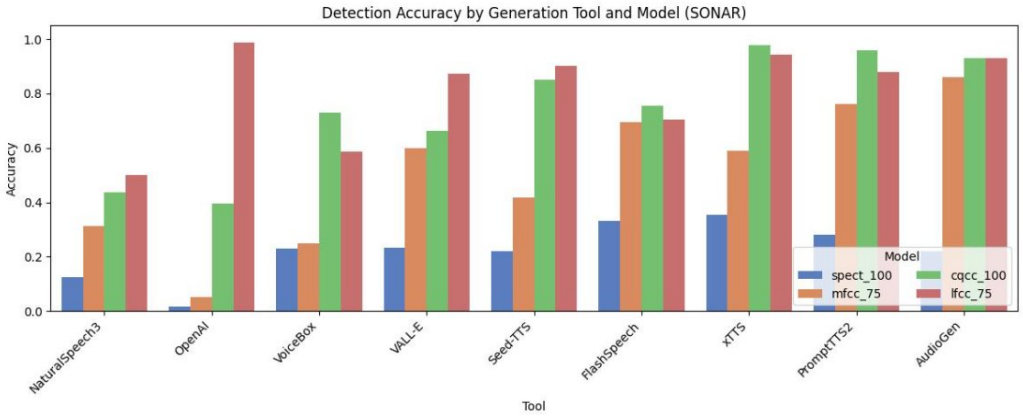


**Figure 9.** Detection accuracy by model on the spoofed clips of each generation tool included in SONAR.

These accuracies are in line with the overall trend: the spectrogram model detects almost none of the modern spoofed clips (accuracy below 0.36 for every tool), and the MFCC model detects only somewhat more, whereas the CQCC and LFCC models flag most of them, often above 0.7. The hardest sources for all models are OpenAI's TTS and NaturalSpeech3: Interestingly, OpenAI clips are almost entirely missed by the spectrogram and MFCC models yet caught by LFCC (≈0.99), while NaturalSpeech3 defeats every model (at most ≈0.50). It must be noted, however, that this figure shows only spoof recall: the same CQCC and LFCC models that look strong here misclassify the majority of bona fide clips as spoofed (BPCER above 0.83, Table 6), so their high per-tool accuracy reflects a bias toward the spoof class rather than reflecting reliable discrimination.

In summary, the four detectors behave consistently across the two unseen-attack scenarios, despite substantial differences in absolute performance. On the ASVspoof 2019 evaluation set, the LFCC-based one-class model is the only configuration that clearly surpasses the all-spoof baseline, whereas on the modern synthesis gathered in SONAR, all four models drop to chance level, with LFCC retaining the best accuracy and balanced accuracy and CQCC retaining the best AUC. This widening gap between in-domain and out-of-domain performance is the central finding of this work: detectors trained on ASVspoof 2019 do not, on their own, generalize successfully to current generation tools.

## 6. Conclusions

Throughout this project, various spoofing detection models were retrained and analyzed using a subset of the ASVspoof 2019 dataset. The models were evaluated across multiple datasets, including the training, development, and evaluation partitions of ASVspoof, and the external SONAR benchmark, which gathers modern text-to-speech systems unseen during training. This analysis provided valuable insights into the strengths and weaknesses of current spoofing detection models.

The LFCC-based model, trained using a one-class learning approach, was the strongest configuration on the in-domain ASVspoof partitions and the only one to clearly surpass the all-spoof baseline on the evaluation set (although on the modern SONAR benchmark, its advantage holds only in accuracy and balanced accuracy, not in AUC). Conversely, models based on spectrograms, MFCCs, and CQCCs—despite achieving near-perfect accuracy

during training and development—displayed significant performance drops on the evaluation set and especially on the external dataset. Crucially, on the modern synthesis systems gathered in SONAR, all four models—including the one-class detector—degraded to near-chance discrimination, and none reached the all-spoof baseline. In fact, the comparatively higher raw accuracy of some models is misleading: It stems from a strong bias toward labelling speech as spoofed (BPCER above 0.83 for CQCC and LFCC), so most genuine clips are flagged as synthetic. This suggests that detectors trained on ASVspoof 2019 do not, by themselves, generalize reliably to current generation tools and, in a real vishing-defense setting, this false-alarm behavior would itself be a serious practical limitation. Overall, the results emphasize that, while the one-class approach generalizes comparatively better, robust detection of present-day deepfake speech remains an open problem, and such systems should be understood as a limited and complementary security layer against vishing rather than a standalone solution.

Several limitations should be considered when interpreting these results. All experiments were run on clean benchmark audio, and the models were not tested under conditions that characterize real vishing calls (telephone bandwidth and codec compression, background noise, reverberation, or emotional and conversational speech), so the connection between the reported metrics and an actual phone-based attack remains to be established. A possible use case would be a real-time component that monitors a call and, when a voice is classified as likely synthetic, issues a discreet warning to the user: for example, through a vibration or tone. However, this should be combined with other security mechanisms rather than be treated as an independent defense. Finally, the comparison is limited to the architectures evaluated in this work; benchmarking against recent large pretrained audio models and transformer-based detectors and testing under degraded acoustic conditions represent natural next steps.

As future work, several promising directions can be explored to enhance spoofing detection systems. Firstly, given its comparatively better generalization in our experiments, the one-class learning approach should be further investigated along with other methods that leverage embeddings and latent space representations, such as those derived from autoencoders or variational autoencoders. Secondly, developing hybrid models that combine the strengths of different feature extraction techniques (for instance, both handcrafted and advanced neural features) could lead to improved performance across diverse spoofing methods. Additionally, a broader range of input features should be considered, potentially incorporating raw audio waveforms, learned embeddings, or phase-based features. Finally, integrating data augmentation and adversarial training strategies could help models become more resilient to emerging spoofing techniques, thus enhancing their ability to generalize to new, unseen methods.

**Author Contributions:** Conceptualization, V.G.M.-E., R.P., G.L. and A.G.; methodology, V.G.M.-E., R.P., G.L. and A.G.; software, V.G.M.-E.; validation, V.G.M.-E., R.P., G.L. and A.G.; formal analysis, V.G.M.-E., R.P., G.L. and A.G.; investigation, V.G.M.-E.; resources, R.P. and G.L.; data curation, V.G.M.-E.; writing—original draft preparation, V.G.M.-E.; writing—review and editing, R.P., G.L. and A.G.; visualization, V.G.M.-E., R.P. and G.L.; supervision, V.G.M.-E., R.P. and G.L.; project administration, R.P. and G.L.; funding acquisition, R.P. and G.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** All datasets used in this research are publicly available and the references are provided.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References


1. CrowdStrike. *2025 Global Threat Report*; Technical Report; CrowdStrike, Inc.: Austin, TX, USA, 2025.
2. Keepnet Labs. *The 2024 Voice Phishing (Vishing) Response Report*; Technical Report; Keepnet Labs Ltd.: London, UK, 2024.
3. Federal Bureau of Investigation. *Internet Crime Report 2024*; Technical Report; FBI Internet Crime Complaint Center (IC3): Washington, DC, USA, 2025.
4. Palacios-Castrillo, C.; Palacios, R.; Gesteira-Miñarro, R.; Chávez-Macías, A.; López, G. Analysis of the Security and Privacy of Smart Personal Assistants with Real and Synthetic Voices. *J. Inf. Secur. Appl.* **2026**, 104554. [CrossRef]
5. Fúster de la Fuente, J.; Solera-Cotanilla, S.; Pérez, J.; Vega-Barbas, M.; Palacios, R.; Álvarez Campana, M.; López, G. Analysis of security and privacy issues in wearables for minors. *Wirel. Netw.* **2024**, *30*, 5437–5453. [CrossRef]
6. Zen, H.; Wagh, R.; Wanderley, M.; Bicalho, G.; Park, R.; Sun, M.; Palacios, R.; Carvalho, L.; Rinaldo, G.; Gupta, A. Ensemble-Based Biometric Verification: Defending Against Multi-Strategy Deepfake Image Generation. *Computers* **2025**, *14*, 225. [CrossRef]
7. Palacios, R.; Gupta, A.; Wang, P.S. Feedback-based architecture for reading courtesy amounts on checks. *J. Electron. Imaging* **2003**, *12*, 194–202. [CrossRef]
8. Sánchez-Rebollo, C.; Puente, C.; Palacios, R.; Piriz, C.; Fuentes, J.P.; Jarauta, J. Detection of jihadism in social networks using big data techniques supported by graphs and fuzzy clustering. *Complexity* **2019**, *2019*, 1238780. [CrossRef]
9. Garrido-Merchán, E.C.; Puente, C.; Palacios, R. Fake News Detection by Means of Uncertainty Weighted Causal Graphs. In *Proceedings of the Hybrid Artificial Intelligent Systems-HAIS, Gijón, Spain, 11–13 November 2020*; Springer: Cham, Switzerland, 2020; pp. 13–24. [CrossRef]
10. Barakat, H.; Turk, O.; Demiroglu, C. Deep Learning-based Expressive Speech Synthesis: A Systematic Review of Approaches, Challenges, and Resources. *EURASIP J. Audio Speech Music Process.* **2024**, *2024*, 11. [CrossRef]
11. Dimension Market Research. Voice Recognition Security Market Size, CAGR, Trends and Forecast 2034. 2025. Available online: https://dimensionmarketresearch.com/report/voice-recognition-security-market/ (accessed on 2 June 2025).
12. Griffin, S.E.; Rackley, C.C. Vishing. In *Proceedings of the 5th Annual Conference on Information Security Curriculum Development (InfoSecCD '08)*; ACM: New York, NY, USA, 2008; pp. 33–35. [CrossRef]
13. Forrest, D. Challenges in Voice Biometrics: Vulnerabilities in the Age of Deepfakes. ABA Banking Journal. 2024. Available online: https://bankingjournal.aba.com/2024/02/challenges-in-voice-biometrics-vulnerabilities-in-the-age-of-deepfakes/ (accessed on 21 April 2025).
14. Alali, A.; Theodorakopoulos, G. Partial Fake Speech Attacks in the Real World Using Deepfake Audio. *J. Cybersecur. Priv.* **2025**, *5*, 6. [CrossRef]
15. Gao, Y. Audio Deepfake Detection Based on Differences in Human and Machine Generated Speech. Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2022. [CrossRef]
16. Li, M.; Ahmadiadli, Y.; Zhang, X.P. A Survey on Speech Deepfake Detection. *ACM Comput. Surv.* **2025**, *57*, 165. [CrossRef]
17. Zhang, B.; Cui, H.; Nguyen, V.; Whitty, M. Audio Deepfake Detection: What Has Been Achieved and What Lies Ahead. *Sensors* **2025**, *25*, 1989. [CrossRef] [PubMed]
18. Wu, Z.; Kinnunen, T.; Evans, N.; Yamagishi, J.; Hanilçi, C.; Sahidullah, M.; Sizov, A. ASVspoof 2015: The First Automatic Speaker Verification Spoofing and Countermeasures Challenge. In Proceedings of the Interspeech 2015, Dresden, Germany, 6–10 September 2015; pp. 2037–2041. [CrossRef]
19. Nautsch, A.; Wang, X.; Evans, N.; Kinnunen, T.; Vestman, V.; Todisco, M.; Delgado, H.; Sahidullah, M.; Yamagishi, J.; Lee, K.A. ASVspoof 2019: Spoofing Countermeasures for the Detection of Synthesized, Converted and Replayed Speech. *IEEE Trans. Biom. Behav. Identity Sci.* **2021**, *3*, 252–265. [CrossRef]
20. Yi, J.; Wang, C.; Tao, J.; Zhang, X.; Zhang, C.Y.; Zhao, Y. Audio Deepfake Detection: A Survey. *arXiv* **2023**, arXiv:2308.14970. [CrossRef]
21. Hong, M.; Jiang, D.; Xie, Z.; Zhao, W.; Wang, G.; Zhang, C.J. Vulnerabilities of Audio-Based Biometric Authentication Systems Against Deepfake Speech Synthesis. *arXiv* **2026**, arXiv:2601.02914. [CrossRef]
22. Alzantot, M.; Wang, Z.; Srivastava, M.B. Deep Residual Neural Networks for Audio Spoofing Detection. In *Proceedings of the Interspeech 2019*; ISCA: Kolkata, India, 2019; pp. 1078–1082. [CrossRef]
23. Zhang, Y.; Jiang, F.; Duan, Z. One-class Learning Towards Synthetic Voice Spoofing Detection. *IEEE Signal Process. Lett.* **2021**, *28*, 937–941. [CrossRef]
24. Li, X.; Chen, P.Y.; Wei, W. Where Are We in Audio Deepfake Detection? A Systematic Analysis over Generative and Detection Models. *ACM Trans. Internet Technol.* **2025**, *25*, 20. [CrossRef]
25. Borzì, S.; Giudice, O.; Stanco, F.; Allegra, D. Is Synthetic Voice Detection Research Going into the Right Direction? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*; IEEE: Piscataway, NJ, USA, 2022; pp. 71–80. [CrossRef]
26. Pedersen, K.T.; Pepke, L.; Stærmose, T.; Papaioannou, M.; Choudhary, G.; Dragoni, N. Deepfake-Driven Social Engineering: Threats, Detection Techniques, and Defensive Strategies in Corporate Environments. *J. Cybersecur. Priv.* **2025**, *5*, 18. [CrossRef]

27. Wang, X.; Yamagishi, J.; Todisco, M.; Delgado, H.; Nautsch, A.; Evans, N.; Sahidullah, M.; Vestman, V.; Kinnunen, T.; Lee, K.A.; et al. ASVspoof 2019: A Large-scale Public Database of Synthesized, Converted and Replayed Speech. *Comput. Speech Lang.* **2020**, *64*, 101114. [CrossRef]
28. Arif, T.; Javed, A.; Alhameed, M.; Jeribi, F.; Tahir, A. Voice Spoofing Countermeasure for Logical Access Attacks Detection. *IEEE Access* **2021**, *9*, 162857–162868. [CrossRef]
29. Hamza, A.; Javed, A.R.; Iqbal, F.; Kryvinska, N.; Almadhor, A.S.; Jalil, Z.; Borghol, R. Deepfake Audio Detection via MFCC Features Using Machine Learning. *IEEE Access* **2022**, *10*, 134018–134028. [CrossRef]
30. Neelima, M.; Prabha, I.S. Hybrid Feature Optimization for Voice Spoof Detection Using CNN-LSTM. *Trait. Signal* **2024**, *41*, 717–727. [CrossRef]
31. Bohara, R.; Bairwa, A.K. Detecting Deepfake Audio Using Spectrogram-Based Machine Learning Approaches. *IEEE Access* **2025**, *13*, 149478–149489. [CrossRef]
32. Kwak, I.Y.; Kwag, S.; Lee, J.; Jeon, Y.; Hwang, J.; Choi, H.J.; Yang, J.H.; Han, S.Y.; Huh, J.H.; Lee, C.H.; et al. Voice Spoofing Detection Through Residual Network, Max Feature Map, and Depthwise Separable Convolution. *IEEE Access* **2023**, *11*, 49140–49152. [CrossRef]
33. Zhang, C.; Yu, C.; Hansen, J.H.L. An Investigation of Deep-Learning Frameworks for Speaker Verification Anti-Spoofing. *IEEE J. Sel. Top. Signal Process.* **2017**, *11*, 684–694. [CrossRef]
34. Gomez-Alanis, A.; Peinado, A.M.; Gonzalez, J.A.; Gomez, A.M. A Light Convolutional GRU-RNN Deep Feature Extractor for ASV Spoofing Detection. In *Proceedings of the Interspeech 2019*; ISCA: Kolkata, India, 2019; pp. 1068–1072. [CrossRef]
35. Shen, Q.; Guo, M.; Huang, Y.; Ma, J. Attentional Multi-Feature Fusion for Spoofing-Aware Speaker Verification. *Int. J. Speech Technol.* **2024**, *27*, 529–543. [CrossRef]
36. Li, X.; Wu, X.; Lu, H.; Liu, X.; Meng, H. Channel-wise Gated Res2Net: Towards Robust Detection of Synthetic Speech Attacks. In *Proceedings of the Interspeech 2021*; ISCA: Kolkata, India, 2021; pp. 4314–4318. [CrossRef]
37. Verma, K.; Mittal, D.; Samanta, S.; Gulati, K.; Kulkarni, O.; Dar, M.A.; Biji, C.L. Deepfake Audio Detection: A Comparative Study of Advanced Deep Learning Models. *IEEE Access* **2025**, *13*, 168447–168462. [CrossRef]
38. Sahidullah, M.; Kinnunen, T.; Hanilçi, C. A Comparison of Features for Synthetic Speech Detection. In *Proceedings of the Interspeech 2015*; ISCA: Kolkata, India, 2015; pp. 2087–2091. [CrossRef]
39. Abdzadeh Ziabari, P.; Veisi, H. A Comparison of CQT Spectrogram with STFT-based Acoustic Features in Deep Learning-based Synthetic Speech Detection. *J. AI Data Min.* **2023**, *11*, 119–129. [CrossRef]
40. Veaux, C.; Yamagishi, J.; MacDonald, K. *CSTR VCTK Corpus: English Multi-Speaker Corpus for CSTR Voice Cloning Toolkit (Version 0.92)*; University of Edinburgh, The Centre for Speech Technology Research: Edinburgh, UK, 2019. [CrossRef]
41. Reimao, R.; Tzerpos, V. FoR: A Dataset for Synthetic Speech Detection. In *Proceedings of the 2019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*; IEEE: Piscataway, NJ, USA, 2019; pp. 1–10. [CrossRef]
42. Ma, H.; Yi, J.; Wang, C.; Yan, X.; Tao, J.; Wang, T.; Wang, S.; Fu, R. CFAD: A Chinese Dataset for Fake Audio Detection. *Speech Commun.* **2024**, *164*, 103122. [CrossRef]
43. Moskalenko, V.; Kharchenko, V.; Semenov, S. Model and Method for Providing Resilience to Resource-Constrained AI-System. *Sensors* **2024**, *24*, 5951. [CrossRef] [PubMed]
44. Mai, K.T.; Bray, S.; Davies, T.; Griffin, L.D. Warning: Humans Cannot Reliably Detect Speech Deepfakes. *PLoS ONE* **2023**, *18*, e0285333. [CrossRef] [PubMed]
45. Bojkovic, Z.S.; Bakmaz, B.M.; Bakmaz, M.R. Hamming Window to the Digital World. *Proc. IEEE* **2017**, *105*, 1185–1190. [CrossRef]
46. Fayek, H.M. Speech Processing for Machine Learning: Filter Banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between. 2016. Available online: https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html (accessed on 6 May 2025).
47. Todisco, M.; Delgado, H.; Evans, N. A New Feature for Automatic Speaker Verification Anti-Spoofing: Constant Q Cepstral Coefficients. In Proceedings of the Speaker and Language Recognition Workshop (Odyssey 2016), Bilbao, Spain, 21–24 June 2016; pp. 283–290. [CrossRef]
48. Todisco, M.; Delgado, H.; Evans, N. Constant Q Cepstral Coefficients: A Spoofing Countermeasure for Automatic Speaker Verification. *Comput. Speech Lang.* **2017**, *45*, 516–535. [CrossRef]
49. MathWorks. Parallel Computing Toolbox. 2025. Available online: https://www.mathworks.com/products/parallel-computing.html (accessed on 2 June 2025).
50. Kinnunen, T.; Lee, K.A.; Delgado, H.; Evans, N.; Todisco, M.; Sahidullah, M.; Yamagishi, J.; Reynolds, D.A. t-DCF: A Detection Cost Function for the Tandem Assessment of Spoofing Countermeasures and Automatic Speaker Verification. In Proceedings of the Speaker and Language Recognition Workshop (Odyssey 2018), Les Sables d'Olonne, France, 26–29 June 2018; pp. 312–319. [CrossRef]

51. Huang, W.C.; Wu, Y.C.; Kobayashi, K.; Peng, Y.H.; Hwang, H.T.; Tobing, P.L.; Tsao, Y.; Wang, H.M.; Toda, T. Generalization of Spectrum Differential based Direct Waveform Modification for Voice Conversion. In *Proceedings of the Interspeech 2019*; ISCA: Kolkata, India, 2019; pp. 2843–2847. [CrossRef]
52. Kinnunen, T.; Lorenzo-Trueba, J.; Yamagishi, J.; Toda, T.; Saito, D.; Villavicencio, F.; Ling, Z. A Spoofing Benchmark for the 2018 Voice Conversion Challenge: Leveraging from Spoofing Countermeasures for Speech Artifact Assessment. In *Proceedings of the Speaker and Language Recognition Workshop (Odyssey 2018)*; ISCA: Kolkata, India, 2018; pp. 239–246. [CrossRef]
53. Kinnunen, T.; Juvela, L.; Alku, P.; Yamagishi, J. Non-Parallel Voice Conversion Using i-Vector PLDA: Towards Unifying Speaker Verification and Transformation. In *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: Piscataway, NJ, USA, 2017; pp. 5535–5539. [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.