Deepfake Speech Detection: A Spectrogram Analysis

Anton Firc
Brno University of Technology
Brno, Czech Republic
ifirc@fit.vut.cz

Kamil Malinka
Brno University of Technology
Brno, Czech Republic
malinka@fit.vut.cz

Petr Hanáček
Brno University of Technology
Brno, Czech Republic
hanacek@fit.vut.cz

ABSTRACT

The current voice biometric systems have no natural mechanics to defend against deepfake spoofing attacks. Thus, supporting these systems with a deepfake detection solution is necessary. One of the latest approaches to deepfake speech detection is representing speech as a spectrogram and using it as an input for a deep neural network. This work thus analyzes the feasibility of different spectrograms for deepfake speech detection. We compare types of them regarding their performance, hardware requirements, and speed. We show the majority of the spectrograms are feasible for deepfake detection. However, there is no general, correct answer to selecting the best spectrogram. As we demonstrate, different spectrograms are suitable for different needs.

CCS CONCEPTS

• **Security and privacy** → *Systems security*; **Biometrics**; **Authentication**.

KEYWORDS

Deepfake, Speech, Image-based, Deepfake Detection, Spectrogram

ACM Reference Format:
Anton Firc, Kamil Malinka, and Petr Hanáček. 2024. Deepfake Speech Detection: A Spectrogram Analysis. In *The 39th ACM/SIGAPP Symposium on Applied Computing (SAC '24), April 8–12, 2024, Avila, Spain.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3605098.3635911

1 INTRODUCTION

The term *deepfake* has no technical definition. It is just a combination of words *deep learning* and *fake*. Deepfakes, thus, are an output of artificial intelligence and machine learning using deep learning that combines, manipulates, or superimposes media to look genuine. Besides positive impact in applications such as entertainment, they can be used for malicious actions such as defaming individuals, spreading fake news, and many others [2, 11].

The recent advancements in artificial intelligence and machine learning allow more people to access deepfake creation. Nowadays, no technical background is required to swap faces or synthesize speech. These advancements create new threats to computer security, especially to the security of voice biometrics systems and scenarios where humans are presented with spoofed speech.

SAC '24, April 8–12, 2024, Avila, Spain

ACM ISBN 979-8-4007-0243-3/24/04.
https://doi.org/10.1145/3605098.3635911

To further highlight the motivation for mitigating the threats posed by deepfake speech, we may look at recent incident reports involving deepfakes. According to a survey performed on 600 security professionals and 3,500 workers around the world, vishing[1] attacks were reported by 69% of companies in 2021, which has risen from 54% experienced in 2020 [15]. Moreover, a vast increase in the usage of vishing attacks is reported by the Intelligence Report from Agari[2] and PhishLabs. The use of vishing attacks in response-based scams between Q1 2021 and Q1 2022 rose by 550%. A recent incident report of a $35 Million bank heist utilizing cloned voice [4] provided an ultimate showcase of the mentioned threats.

The reliability of voice biometrics systems might be compromised by the presentation of carefully prepared synthetic speech. The current state of technology allows the creation of synthetic speech of such quality that human listeners cannot differentiate between real and spoofed speech [28]. If this issue remains unaddressed, the reliability of voice biometrics will degrade over time due to spoofing attacks [3, 11, 36]. Simultaneously, more deepfake-powered heists and scams will occur. A recent study [10] demonstrated how easily such an attack on speaker recognition (identity verification) using synthetic speech might be performed. Voice biometrics systems are not limited to customers' identity verification. Such systems are used for various use cases: *fraud detection* - in banking scenarios, to prevent fraudsters from asking for multiple loans by telephone by identifying their voice in repeated requests; *forensics purposes* - to use speech samples in court trials as evidence; or *threat detection* - to identify a phone call made by a wanted criminal or criminal organization. Thus, supporting voice biometrics systems with proper deepfake detection mechanisms is crucial to retain their reliability and suitability in the mentioned use cases.

One of the sparsely explored areas is image-based detection. The main idea behind image-based detection is that the speech is converted into a spectrogram (image) and then provided as input to a deep neural network for classification [20, 32].

In this paper, we decided to build on the existing research on image-based detection. Remaio [30] was the first to propose image-based deepfake speech detection and to assess multiple detector architectures and spectrograms regarding their feasibility for deepfake detection. Khochare et al. [20] later extended the idea and explored the behaviour of Temporal Convolutional Networks (TCNs) with Mel-Spectrogram as input. We follow up on these works by building six new detectors for deepfake speech as variations to formerly proposed TCN architecture with different spectrograms as inputs and evaluate their performance.

The main contributions of this paper may be stated as follows:

[1] A portmanteau of voice and phishing: social engineering attack analogous to phishing, but using spoofed voice.
[2] https://info.phishlabs.com/quarterly-threat-trends-and-intelligence-may-2022

- We assess the feasibility of different spectrograms as inputs for deepfake speech detection.
- We perform an in-depth analysis of these spectrograms' accuracy, resource requirements, and learning curves using available deepfake datasets.

2 RELATED WORK

This section provides an overview of areas related to image-based speech deepfake detection. We list the available deepfake speech datasets and provide an overview of speech deepfake detection methods.

2.1 Speech deepfakes

Speech deepfakes might be divided into two main categories according to the synthesis inputs: *Text-to-Speech (TTS)* synthesis and *Voice Conversion* (VC). The main difference is in the input data. As the name suggests, TTS consumes written text and an embedding recording and produces synthesized speech sounding like the particular individual from the embedding recording. VC, in turn, consumes a source voice saying the desired phrase and a target voice and outputs the source phrase spoken by the target voice.

Recently, novel deepfake speech datasets have been released for public use. One of the most influential datasets is provided by the ASVSpoofChallenge [38, 39]. In total, 19 synthesis systems are used [37]. TTS and VC technologies are used, and some synthesis technology includes Variational Autoencoders [14, 43], Gaussian mixture models [25] or LSTM-RNN models [24, 42].

WaveFake dataset [12] is a bilingual dataset containing English and Japanese speech. The synthesized speech is generated using a multitude of TTS GAN-based systems such as MelGAN [22], HiFi-GAN [21], or WaveGlow [27].

FakeAVCeleb dataset [19] contains a combination of fake video and audio. The audio was synthesized using a TTS system Real-Time-Voice-Cloning [9]. The same tool was also used to create a bilingual dataset (English and Czech) [10].

Finally, the Fake or Real (FoR) dataset [31] uses commercial TTS systems such as Deep Voice 3 or Google TTS to synthesize deepfake utterances. The dataset contains multiple subsets with trimmed, normalized or re-recorded utterances.

Additionally, SYNSPEECHDDB [44] and FMFCC-A [45] datasets were published; however, none is currently available.

2.2 Speech deepfake detection

We divide deepfake detection into two categories: *feature-based* and *image-based*. A definition and overview of the latest methods for each category is then provided.

Feature-based deepfake detection uses a numerical representation of the signal input. These features are then used as input for various classifier types, primarily neural networks. Todisco et al. [35] propose a novel feature named Q-cepstral coefficient and use a Gaussian Mixture Model (GMM) based classifier. Ahmed et al. [1] similarly use a GMM-based classifier to detect differences in the spectral power between genuine and synthetic speech. Chen et al. [8] propose a new loss function named Large Margin Cosine Loss. Similarly, Cáceres et al. [7] propose a novel loss function to be used with a linear fusion of classifiers. The classifier fusion was also proposed by Kang et al. [18].

Image-based deepfake detection uses image representation of speech as an input. The features are extracted from a speech sample and represented as an image – spectrogram. These spectrograms are then provided as input into Deep Neural Networks, mainly Convolutional Neural Networks, for classification.

The initial idea of image-based detection was proposed by R. Remaio [30, 32]. Remaio's work compares the common feature-based detectors to image-based detectors. The results suggest that image-based detection performs better, even on previously unseen samples. Kchochare et al. [20] build on Remaio's work by utilizing the Temporal Convolutional Network (TCN). The architecture is further described further in Section 4.4. The results indicate that TCN performs well in detecting deepfake speech in the form of Mel-Spectrograms. The best-achieved performance is claimed to be 92%. Unfortunately, no other input features than Mel-Spectrogram are tested.

Our work aims to extend further the scope of the paper published by Kchochare et al. [20]. We test six more deepfake detectors varying in input spectrograms such as STFT-Spectrograms, MFCC-Cepstrograms, chromagrams, and others to investigate the usability of the proposed approach to speech deepfake detection. We use the same Temporal Convolutional Network architecture, further described in Section 4.4, and compare the suitability of proposed detectors for image-based deepfake detection.

3 PROBLEM STATEMENT

Previous sections outlined the image-based approach to deepfake detection that we decided to follow up on. This section outlines the research questions.

Image-based deepfake speech detection is a new and promising approach to deepfake speech detection. However, all efforts seem to focus on using the Mel-spectrogram-based detectors. There are many different spectrograms to choose from, so we find it crucial to understand how they perform. Using a different spectrogram might significantly change the performance of an existing detector without any special change in its architecture. The results of this work thus help create better-performing deepfake speech detectors. This work aims to answer the following research questions:

RQ1: *What spectrogram delivers the best accuracy?*

RQ2: *How does the performance of the detectors vary based on the used dataset?*

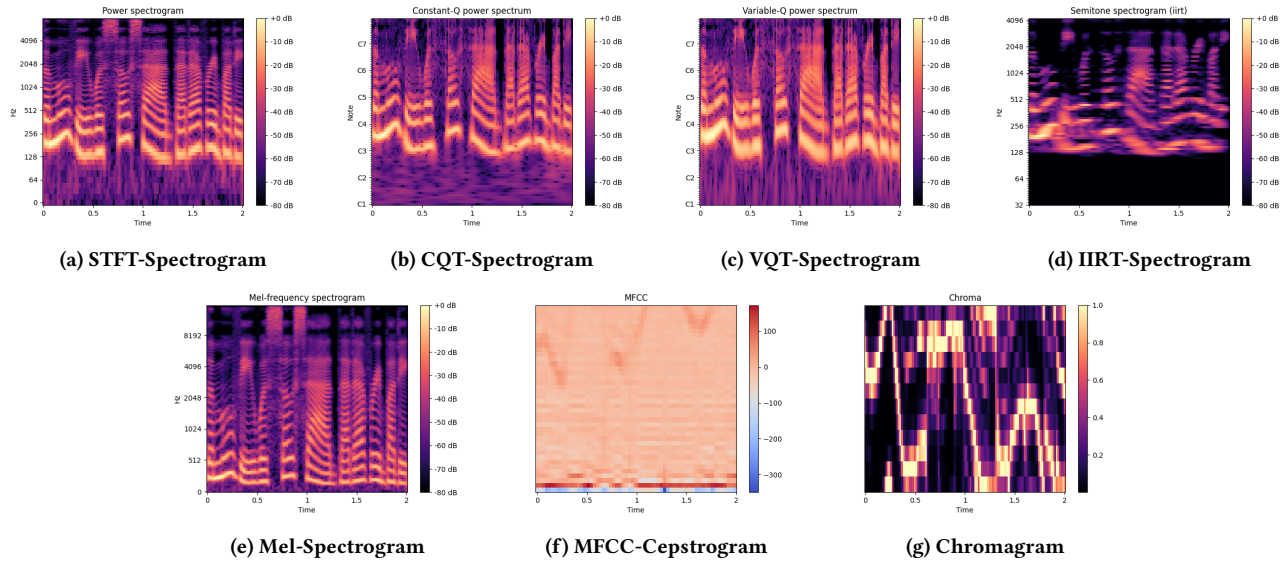
RQ3: *What is the link between accuracy and resource requirements?*

RQ4: *How does the learning curve behave for different spectrograms?*

4 EXPERIMENT DESIGN

The experiments compare the feasibility of different spectrograms as inputs for the TCN-based deepfake speech detection method (see Section 4.4). The experiments consist of two parts:

The *first part* examines the accuracy of different spectrograms. We construct seven deepfake detectors as variations of the mentioned TCN-based detector and train them using the datasets (subsets) described in Table 1. In addition to single-dataset validation,



(a) STFT-Spectrogram (b) CQT-Spectrogram (c) VQT-Spectrogram (d) IIRT-Spectrogram

(e) Mel-Spectrogram (f) MFCC-Cepstrogram (g) Chromagram

Figure 1: Examples of used spectrograms. Each image represents the same recording from the *for-2-second* dataset.

we perform cross-dataset validation by evaluating all trained models over all datasets. We evaluate the models using all evaluation datasets (excluding ones used for training). Datasets used for evaluation are either separate validation subsets of training datasets or standalone datasets with smaller sample count that makes them unsuitable for the training of deep neural networks (see Table 2). The description of these datasets is provided further in this section. We use Equal Error Rate (EER) to evaluate performance. This threshold-independent metric allows better comparison between different deepfake detectors when using the same dataset. While precision and recall are often used to measure the performance of machine learning models, EER became a standard for evaluating deepfake detection methods.

Next, in the *second part*, we examine the data requirements for each spectrogram. We compare the spectrograms on consumed storage space, consumed RAM during training, and time needed to extract each spectrogram. Additionally, we explore the learning curves of the proposed detectors. Examining the learning curves helps to understand how the available training data count affects the detector performance. As the WaveFake dataset contains the most samples, we decided to use it. We split the dataset into ten equal parts while preserving the original ratio of genuine and deepfake samples. Then, we construct ten training sets that progressively grow in size by concatenating the previously divided parts. The smallest part consists of almost 24k samples, and the entire dataset of 240k. Finally, we assume that each dataset sample contains the same information. Thus, the changes in performance are purely based on the amount of provided training data.

4.1 Spectrograms

Spectrograms display the time-frequency representation of speech signals. Since the 1940s, spectrograms have been a basic tool for understanding how sounds are produced and how the information is encoded in speech signal [29]. An example for each used spectrogram is shown in Figure 1. This section provides preliminary information on used spectrograms:

STFT-Spectrogram STFT-Spectrogram displays the magnitude of a short-time Fourier transform (STFT). We obtain STFT by windowing and taking a discrete Fourier transform of the signal. The STFT itself describes the evolution of frequency components over time. Instead of directly visualizing the complex-valued output matrix, log spectra is used. The 2-dimensional log spectra is then plotted as a heat-map (spectrogram) [6, 29]. An example is shown in Figure 1a.

CQT-Spectrogram Constant-Q transform (CQT) is similar to the Discrete Fourier Transform but with a constant centre frequency ratio to resolution, as shown in Figure 1b. This transform is used to obtain a constant pattern in the frequency domain for sounds with harmonic frequency components [5].

Table 1: Used training datasets (including their validation subset).

Dataset name	*Acronym*
for-2seconds	F2S
for-rerecorded	FREC
ASVSpoof 2019 LA	AS19
WaveFake	WF

Table 2: Used validation datasets.

Dataset name	*Acronym*
AVCeleb	AVC
ASVSpoof 2021 DF eval	AS21
Firc and Malinka	F&M

VQT-Spectrogram Variable-Q transform (VQT) is a modification of CQT (see Figure 1c) by introducing a new parameter γ, that dampens the Q factor at lower frequencies while maintaining roughly a constant Q at higher frequencies. This dampening improves the time resolutions at lower frequencies [17].
IIRT-Spectrogram IIRT-Spectrogram is a time-frequency representation using a multi-rate filter bank consisting of IIR filters [26]. An example is shown in Figure 1d.
Mel-Spectrogram Mel-Spectrogram is a particular type of spectrogram utilizing the Mel-scale, which is a logarithmic transformation of a signal's frequency. The sounds of equal distance on the Mel-scale are perceived as equal distance on the pitch to humans. Mel-Spectrogram thus visualizes the speech on the Mel-scale instead of using the magnitude of STFT as shown in Figure 1e [29].
MFCC-Cepstrogram Mel-Frequency Cepstrum Coefficients are computed by a frequency analysis based upon a filter bank with approximately critical band spacing of the filters and bandwidths. Firstly, STFT is done, and then the calculated values are grouped together in critical bands and weighted by a triangular weighting function. Next, logs of the powers at each of the mel-frequencies are taken, and the Discrete Cosine Transform of the list of mel-log-powers is computed. Finally, MFCCs are the amplitudes of the resulting spectrum [29]. As the MFCC features represent cepstral, not spectral, characteristics, we define the image representation as a cepstrogram. An example is shown in Figure 1f.
Chromagram Pitch can be decomposed into two components: *tone height* and *chroma*. The tone height refers to the octave number, and the chroma to the respective pitch spelling attribute. There are 12 different chroma values. A pitch class is defined as the set of all pitches that share the same chroma. Chroma features aggregate all spectral information related to a given pitch class into a single coefficient, as shown in Figure 1g [41].

4.2 Used Datasets

We primarily use the Fake or Real (FoR) dataset [31] as it is used in the previous work we aim to extend, thus ensuring the comparability of our results to the previous ones. Similar to the previous work, we use only specific parts of this dataset:

- *for-2-second* consists of normalized samples trimmed to two-second length. We use this subset because both Remaio and Khochare use it.
- *for-rerecorded* consists of re-recorded utterances from the *for-2-second* dataset. Re-recording was done by playing the recording using regular speakers and recording it again using a non-professional microphone. We use this subset because it is used by Remaio [30] to assess the robustness of proposed detectors.

Next, we use the ASVSpoof 2019 LA dataset, which was used for training in the two most recent ASVSpoof challenges. Additionally, we use the ASVspoof 2021 DF Eval dataset [39]. As the name suggests, this dataset consists only of evaluation samples.

WaveFake dataset [12] consists only of deepfake samples. Fortunately, the documentation of this dataset contains information on genuine datasets (LJSpeech [16], JSUT [34]) used for deepfake creation; thus, we supplemented it with real samples and used the whole set for training.

We also use the FakeAVCeleb dataset [19], extracting only the fake audio from contained samples. Unfortunately, after processing, the dataset (see section 4.3) consists only of 2800 samples, split 50 : 50 (real:deepfake). Due to this limited number of samples, the dataset is suitable only for evaluation. Finally, we use the deepfake dataset proposed in [10]; this dataset's limited size makes it usable only for evaluation.

4.3 Data preparation

The FoR dataset was left intact, as the recordings were already trimmed to a two-second length and normalized. The other datasets contain samples of variable length. The detector requires only a representation of two-second samples; all samples shorter than two seconds are discarded, and all longer samples are split into two-second segments.

The only dataset suitable for training that does not explicitly contain training, development, and validation subsets is WaveFake. We thus randomly split the dataset once with a ratio of $7 : 1 : 2$ (train:dev:val).

As the FakeAVCeleb dataset [19] contains only videos, the speech had to be separately extracted. We used `ffmpeg`[3] tool to convert the *mp4* files to *wav* and to remove silence (segments under 40 dB).

4.3.1 Spectrogram creation. To convert the audio samples in waveform audio format (WAV) to their image representations, we use the *librosa*[4] library. The parameters for Mel-Spectrogram creation were set the same as by Khochare et al. [20] to ensure comparability of results and parameters for other spectrograms to copy this setting. The window length is set to 1024, hop length to 256, Mel frequency count to 256, and the MFCC-cepstrogram's MFCC count is set to 40.

4.4 Detector architecture

The main building block of the network architecture is a Temporal Convolutional Network (TCN). TCN is a variant of the convolutional neural network. It is a time-series model capturing long-range patterns using a hierarchy of temporal convolutional filters. The network employs causal convolutions and dilations, making it suitable for sequential data such as spectrograms [23, 40].

The overall network architecture is shown in Figure 2. Khochare et al. [20] state that *CustomPooling* was used. Unfortunately, no more detail on this layer is provided; thus, we try to simplify this layer by using the provided MaxPooling and AveragePooling layers. Since our preliminary experiments have shown no significant change in performance according to the selected pooling layer, we opted for the AveragePooling layer.

4.4.1 Implementation details. The detector is implemented in Python using the TensorFlow framework. The used TCN layers are available from GitHub[5] [33]. The demo code for our implementation was also published on GitHub [6].

Each TCN differs in used kernel size but retains all other parameters. The causal mode is turned on, and the dropout rate is 0.2. The

[3] https://ffmpeg.org
[4] https://librosa.org
[5] Commit: a3d72cf58343fb4ad42f13f52532d071602dd36c
[6] GitHub link to be provided for the camera-ready manuscript.

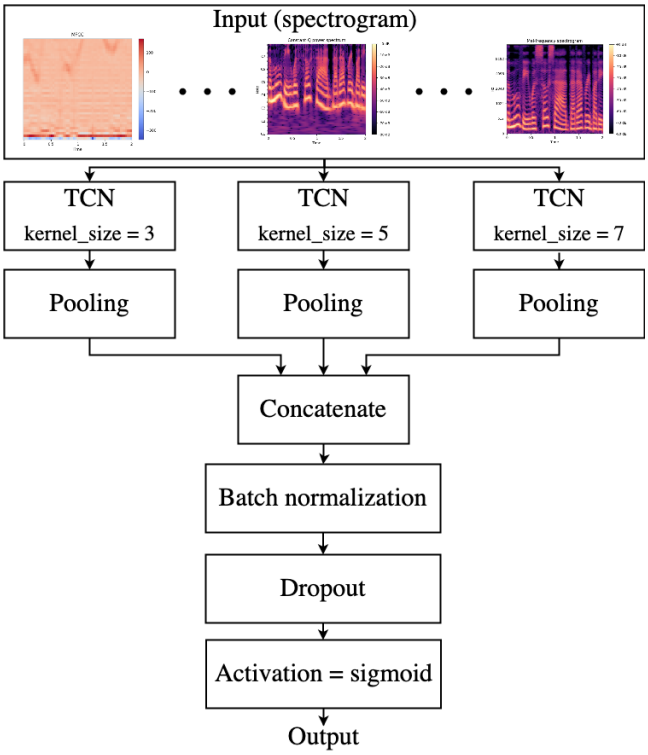


Figure 2: Deepfake detection network architecture.

pooling layer was selected as AveragePooling, with pool size and strides set to (10,5). Binary Crossentropy is used as a loss function with Adam optimizer. Early stopping is used to prevent over-fitting. We use the *sklearn* library to balance the datasets. Finally, we use a learning rate scheduler that reduces the learning rate based on validation loss.

5 EVALUATION

This section provides information on the execution of proposed experiments and answers the research questions stated previously in this paper. The structure of this section follows the logical division into parts as previously proposed. Firstly, we discuss the feasibility of different spectrograms for deepfake detection. Then, we examine the detectors' data requirements and learning curves.

Firstly, we compare the performance of different detectors (spectrograms) over the same dataset. As Figure 3 shows, a comparison of EER collected from all detectors suggests that the MFCC-cepstrogram-based detector performs the best. The STFT-spectrogram-based detector very closely follows its performance. Moreover, the overall performance is comparable with the results of the ASVspoof2021 challenge, where the detectors in the deepfake task achieved EER between 15.64 and 29.75% [39].

Following this, we conduct cross-dataset validation. This involves taking each detector (spectrogram) trained on one of the training sets (refer to Table 1) and validating it using all other validation sets (refer to Table 2), excluding the validation set from the same dataset if applicable. Subsequently, we calculate the Equal

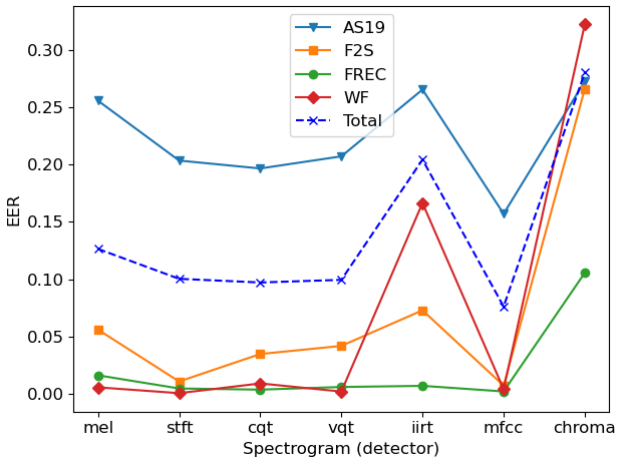


Figure 3: Equal Error Rates (EER) for each training dataset and detector. The dashed line shows the total EER for each detector.

Table 3: EER for each detector from the cross-dataset validation. The EER is collected by joining scores for each detector through all validation datasets and then calculated over the whole set.

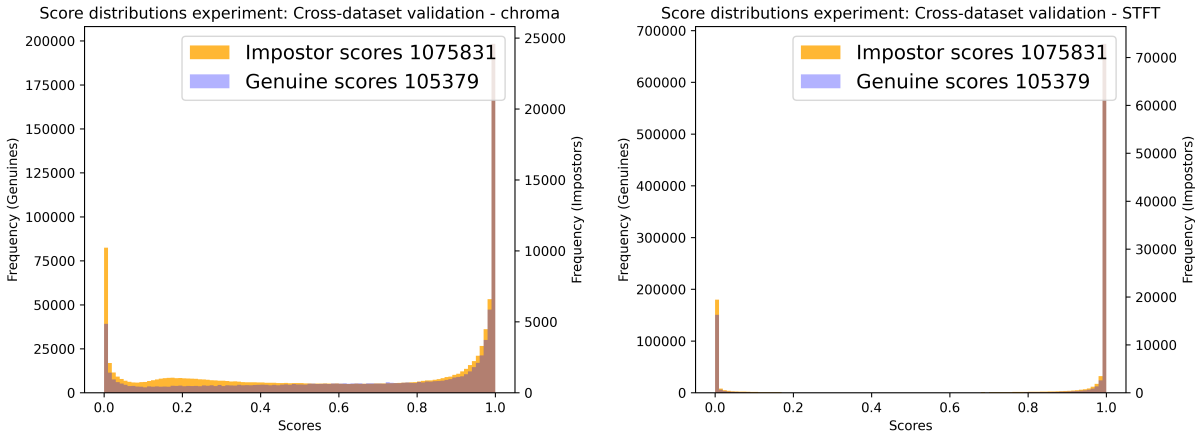
			EER (%)			
Mel	*STFT*	*CQT*	*VQT*	*IIRT*	*MFCC*	*Chroma*
45.85	46.25	46.43	45.88	49.05	48.23	46.58

Error Rate (EER) across all validation sets for each detector (spectrogram), as presented in Table 3, employing the *pyeer*[7] library. It's worth noting that an increase in EER is observed, indicating that all detectors face challenges when classifying previously unseen samples. Additionally, it's noteworthy that the detectors which performed the worst in the previous experiment now exhibit improved performance. This is attributed to the prior uncertainty of the detector, which is now reflected in the new and unknown data.

In contrast, the well-performing detectors provide scores very close to lower or higher boundaries for both classes - 0.0 or 1.0. Thus, if a well-performing detector makes a mistake, there is a large difference between the predicted and target scores. On the contrary, the difference between these scores is much smaller in a mistake made by a worse-performing detector. This means that for the worse-performing detector, these *"mistakes"* are better distributed in the possible score range, not only around 0.0 and 1.0, significantly lowering the EER rates. Figure 4 provides an example illustrating this behaviour.

Ultimately, the detector based on the STFT-spectrogram remains one of the top-performing ones. Its ability to perform well on both familiar (same training and validation dataset, separated training and validation subsets) and unfamiliar (different training and validation dataset) data suggests it may be the optimal choice.

[7] https://pypi.org/project/pyeer/



(a) Score distribution plot for Chromagram-based detector.

(b) Score distribution plot for STFT-based detector.

Figure 4: The difference in score distribution between two detectors. The left represents the scores of a poorly performing model, and the right represents a well-performing model. Notice that the scores of a poorly performing detector are distributed through the whole range. In contrast, the scores of the well-performing detector are tightly packed around 0.0 and 1.0. Ultimately, when calculating EER, this uncertainty helps the poorly-performing detector to make a mistake with smaller *cost (certainty)*, while the well-performing detector makes a costly mistake. Because of this behaviour, the performance of the poorly-performing detectors is better on unseen data.

Next, we examine the data requirements: consumed storage space, consumed RAM during training, and time needed to extract each spectrogram. The extraction time was measured as an average time to extract given features from 1413 two-second samples. As Figure 5 shows, the STFT-spectrogram is the most resource-hungry. However, the STFT-spectrogram is the fastest one for extraction. Finally, the MFCC-Cepstrgorgam and Chromagram are well-balanced regarding all monitored aspects and other spectrograms.

Finally, we explore the learning curves of the proposed detectors. Examining the learning curves helps to understand how the available training data count affects the detector performance. As the WaveFake dataset contains the most samples, we decided to use it. We split the dataset into ten equal parts while preserving the original ratio of genuine and deepfake samples. Then, we construct ten training sets that progressively grow in size by concatenating the previously divided parts. The smallest part comprises almost 24k samples and the full dataset of 240k samples. Finally, we assume that each dataset sample contains the same information. Thus, the changes in performance are purely based on the amount of provided training data.

To plot the learning curves, we again use EER. The better the performance, the lower the EER value. Thus, the learning curves should decrease over time. As Figure 6 shows, the change in performance is primarily similar for all detectors. The performance stops rapidly increasing after seeing 70k training samples. We may notice that the learning curves for CQT and VQT-spectrograms are steeper initially. Thus, these spectrograms might be more powerful in low-data scenarios.

Moreover, the abnormal initial increase in EER for MFCC and Mel-spectrogram-based detectors is solely based on the problematic calculation of the EER. As the utter majority of the score is packed

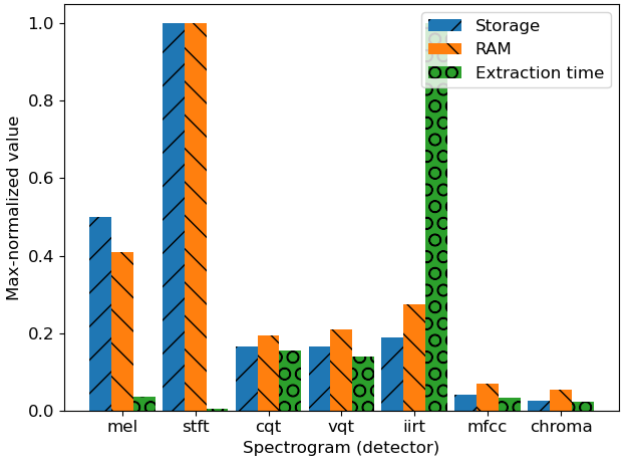


Figure 5: Comparison of data requirements for each spectrogram. *Storage* refers to the size of the extracted features, *RAM* refers to the used RAM during training, and *Extraction time* refers to the average time of extraction of the spectrogram. All values have been max-normalized for a more transparent comparison.

around 0.0, the EER fluctuates; however, this change does not reflect any change in the actual performance of the model. This behaviour can be seen by comparing Figure 7a and Figure 7b, where a very different EER is calculated for two models with similar performance.

All of the research questions were answered:

RQ1: What spectrogram delivers the best accuracy?

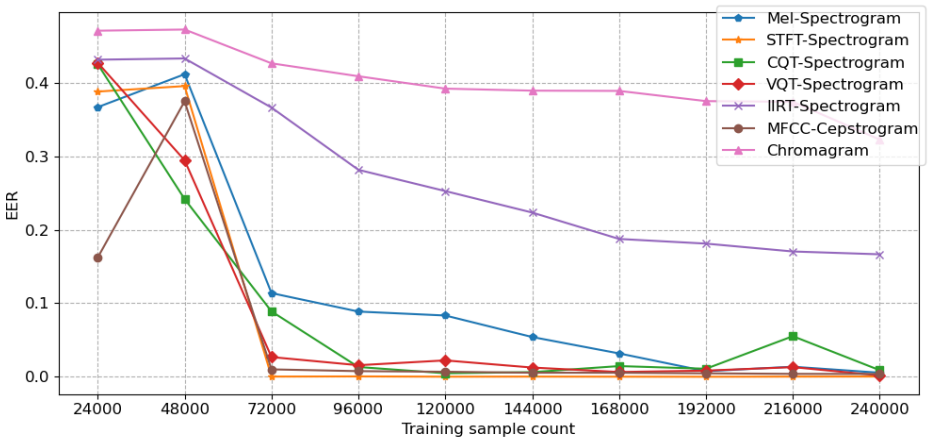
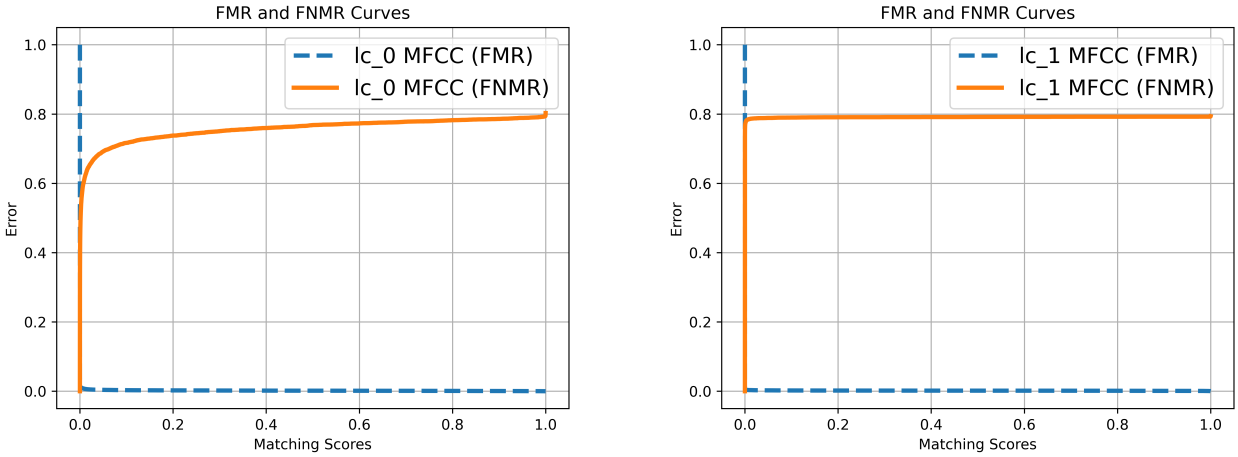


Figure 6: Learning curves of detectors.



(a) Training with 24k training samples.

(b) Training with 48k training samples.

Figure 7: Demonstration of problematic EER calculation for the MFCC-spectrogram-based detector in low training data setting. The same behaviour is observed with the Mel-spectrogram-based detector.

The accuracy over known data is the best when using the MFCC-spectrogram. The accuracy over unknown data is the best for Mel-spectrogram and VQT-spectrogram. However, if we seek an overall well-performing detector (spectrogram), the STFT-spectrogram seems the best choice.

To further extend this claim, STFT computation is a mandatory step for computing other spectral representations. This not only means fewer computations are required for feature extraction but also that the STFT-spectrogram contains the *most raw information* among other spectrograms. This raw data may work in harmony with the preference of neural networks to work with raw data [13].

RQ2: How does the performance of the detectors vary based on the used dataset?

To address this research question, we exclude details about the utilized spectrogram and focus solely on analyzing performance based on the training and evaluation dataset. Table 4 displays noteworthy performance variations depending on these datasets.

Among the training datasets, WaveFake emerges as the least effective option. Detectors trained on WaveFake struggle to accurately classify deepfake speech across virtually all evaluation datasets. Despite its substantial size, the issue appears to lie in the dataset's content. The speech synthesis tools cover only a small portion of deepfake speech - GAN-based TTS algorithms, while most algorithms are derived from each other. The large quantity of analogous data makes the detector over-fit the GAN-based TTS architecture and perform poorly on all other data.

Several other datasets also suffer from a scarcity of speech synthesizers, such as AVC or F&M. However, their limited sizes prevent over-fitting, leading to reasonable performance across diverse datasets.

Table 4: Cross-dataset validation EER (%). Training and evaluation sets do not overlap for any dataset.

		Training dataset			
		F2S	*FREC*	*AS19*	*WF*
Evaluation dataset	*F2S*	6.99	22.58	46.04	100
	FREC	39.36	2.13	47.42	100
	AS19	47.73	51.00	23.37	100
	WF	100	51.93	51.15	7.33
	AS21	47.66	51.00	31.02	47.01
	AVC	38.49	46.01	51.31	100
	F&M	55.85	50.64	48.82	100

Ultimately, the ASVSpoof 2019 LA dataset proves optimal for training deepfake speech detectors. It offers a multitude of utterances produced by various synthesizers, demonstrating well-balanced performance across different datasets.

Interestingly, the FoR dataset delivers satisfactory performance despite exclusively containing speech synthesized by commercial TTS systems. Notably, the challenge with "commercial TTS" lies in the system's synthesis of a generic speaker rather than a "clone", as done by other synthesizers. This distinction could pose issues for malicious applications requiring specific speaker emulation, such as vishing. However, as our preliminary findings suggest, this difference might not be discernible for deepfake speech detectors.

To further address the performance of detectors over previously unseen data, differences between scores for evaluation with the same dataset and different datasets might be observed in Table 4. The worst evaluation EER, in this case, slightly surpasses 23% for the AVSspoof 2019 dataset. A similarly performing detector (in terms of EER) ranked 39 out of 50 in the ASVSpoof 2019 challenge. However, looking at the evaluation scores with other datasets, an EER of 100% is not uncommon. Moreover, the differences between same-dataset and other-dataset evaluation scores suggest how badly the detector's performance suffers when dealing with previously unseen data. Because of these significant differences, deepfake speech detectors should be evaluated in a cross-dataset manner to provide reliable and as close as possible to real usage evaluation results.

RQ3: What is the link between accuracy and resource requirements?

The STFT-spectrogram is the most resource-hungry and most accurate simultaneously. However, the MFCC-cepstrogram delivers reasonable accuracy, while the data requirements are among the lowest. This means there is no direct connection between the resource requirements and final accuracy. Thus, the final accuracy does not depend on the resources used but on the quality and suitability of the extracted information.

RQ4: How does the learning curve behave for different spectrograms?

In general, more data equals better performance. If we, thus, have unlimited training data and resources, the decision seems to be the STFT-spectrogram. However, the CQT-spectrogram or VQT-spectrogram should be more suitable in low-resource scenarios. As previously seen, the learning curve for these spectrograms is much steeper in the beginning. Offering better performance in scenarios with less than 70k training samples. For all detectors, we see EER continuously decrease with more training samples.

6 CONCLUSIONS

The STFT-spectrogram provides the highest accuracy in general. The performance over known and unknown samples is balanced. The resource analysis revealed that the MFCC-cepstrogram might be more desirable in resource-limited scenarios. Low resource requirements and fast extraction times balance its lower performance. Finally, when implementing deepfake speech detectors, it is essential to consider the number of available training samples. The CQT and VQT-spectrogram-based detectors have a steeper learning curve. Thus, they outperform other tested detectors when trained with less than 70k samples.

The cross-dataset validation shows that the selection of training and validation datasets has a non-negligible impact on the final performance. Careful selection of these datasets might make the final results look better or worse than the actual performance. For a more reliable and attainable presentation of results, we highly encourage incorporating cross-dataset validation into all further development and testing of deepfake speech detection methods.

As demonstrated, there is no single answer to what detector performs the best or what is the best dataset. Numerous parameters might be taken into account, and the behaviour of the detectors significantly varies based on these parameters. It is thus crucial to understand the environment where such a detector will be deployed and set it up to deliver the best possible performance given the circumstances.

ACKNOWLEDGMENTS

This work was supported by the national project "NABOSO: Tools To Combat Voice DeepFakes" (with code VB02000060) funded by the Ministry of the Interior of the Czech Republic and the Brno University of Technology internal project FIT-S-23-8151. Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.

REFERENCES

[1] Muhammad Ejaz Ahmed, Il-Youp Kwak, Jun Ho Huh, Iljoo Kim, Taekkyung Oh, and Hyoungshick Kim. 2020. *Void: A Fast and Light Voice Liveness Detection System.* USENIX Association, USA.

[2] Jon Bateman. 2020. *Deepfakes and Synthetic Media in the Financial System: Assessing Threat Scenarios.* Technical Report. Carnegie Endowment for International Peace. i–ii pages. http://www.jstor.org/stable/resrep25783.1

[3] Jean-Francois Bonastre, Hector Delgado, Nicholas Evans, Tomi Kinnunen, Kong Aik Lee, Xuechen Liu, Andreas Nautsch, Paul-Gauthier Noe, Jose Patino, Md Sahidullah, Brij Mohan Lal Srivastava, Massimiliano Todisco, Natalia Tomashenko, Emmanuel Vincent, Xin Wang, and Junichi Yamagishi. 2021. Benchmarking and challenges in security and privacy for voice biometrics. https://doi.org/10.48550/ARXIV.2109.00281

[4] Thomas Brewster. 2022. Fraudsters cloned company director's voice in $35 million bank heist, police find. https://www.forbes.com/sites/thomasbrewster/2021/10/14/huge-bank-fraud-uses-deep-fake-voice-tech-to-steal-millions/?sh=776258a75591

[5] Judith C. Brown. 1991. Calculation of a constant Q spectral transform. *The Journal of the Acoustical Society of America* 89, 1 (1991), 425–434. https://doi.org/10.1121/1.400476 arXiv:https://doi.org/10.1121/1.400476

[6] Tom Bäckström. 2019. Spectrogram and the STFT. online. https://wiki.aalto.fi/display/ITSP/Spectrogram+and+the+STFT

[7] Joaquín Cáceres, Roberto Font, Teresa Grau, and Javier Molina. 2021. The Biometric Vox System for the ASVspoof 2021 Challenge. In *Proc. 2021 Edition of the*

Automatic Speaker Verification and Spoofing Countermeasures Challenge. 68–74. https://doi.org/10.21437/ASVSPOOF.2021-11

[8] Tianxiang Chen, Avrosh Kumar, Parav Nagarsheth, Ganesh Sivaraman, and Elie Khoury. 2020. Generalization of audio deepfake detection. In *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*. 132–137.

[9] Jemine Corentin. 2019. *Real-time Voice Cloning*. Master thesis. Université de Liège, Liège, Belgique. https://matheo.uliege.be/handle/2268.2/6801?locale=en

[10] Anton Firc and Kamil Malinka. 2022. The Dawn of a Text-Dependent Society: Deepfakes as a Threat to Speech Verification Systems. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing* (Virtual Event) *(SAC '22)*. Association for Computing Machinery, New York, NY, USA, 1646–1655. https://doi.org/10.1145/3477314.3507013

[11] Anton Firc, Kamil Malinka, and Petr Hanáček. 2023. Deepfakes as a threat to a speaker and facial recognition: An overview of tools and attack vectors. *Heliyon* 9, 4 (2023), e15090. https://doi.org/10.1016/j.heliyon.2023.e15090

[12] Joel Frank and Lea Schönherr. 2021. WaveFake: A Data Set to Facilitate Audio Deepfake Detection. arXiv:2111.02813 [cs.LG]

[13] Alejandro Gomez-Alanis, Antonio M. Peinado, José Andrés González López, and Angel M. Gomez. 2018. Performance evaluation of front- and back-end techniques for ASV spoofing detection systems based on deep features. In *Proc. IberSPEECH 2018*. 45–49. https://doi.org/10.21437/IberSPEECH.2018-10

[14] Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang. 2016. Voice conversion from non-parallel corpora using variational auto-encoder. In *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. 1–6. https://doi.org/10.1109/APSIPA.2016.7820786

[15] Laurie Iacono, Josh Hickman, and Caitlin Muniz. 2022. The rise of vishing and Smishing attacks – the monitor, issue 21. https://www.kroll.com/en/insights/publications/cyber/monitor/vishing-smishing-attacks

[16] Keith Ito and Linda Johnson. 2017. The LJ Speech Dataset. online. https://keithito.com/LJ-Speech-Dataset/

[17] Frank C. Cwitkowitz Jr. 2019. *End-to-End Music Transcription Using Fine-Tuned Variable-Q Filterbanks*. Thesis. Rochester Institute of Technology.

[18] Woo Hyun Kang, Jahangir Alam, and Abderrahim Fathan. 2021. Investigation on activation functions for robust end-to-end spoofing attack detection system. In *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*. 83–88. https://doi.org/10.21437/ASVSPOOF.2021-13

[19] Hasam Khalid, Shahroz Tariq, Minha Kim, and Simon S. Woo. 2021. FakeAVCeleb: A Novel Audio-Video Multimodal Deepfake Dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. https://openreview.net/forum?id=TAXFsg6ZaOl

[20] Janavi Khochare, Chaitali Joshi, Bakul Yenarkar, Shraddha Suratkar, and Faruk Kazi. 2021. A Deep Learning Framework for Audio Deepfake Detection. *Arabian Journal for Science and Engineering* (08 Nov 2021). https://doi.org/10.1007/s13369-021-06297-w

[21] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 17022–17033. https://proceedings.neurips.cc/paper_files/paper/2020/file/c5d736809766d46260d816d8dbc9eb44-Paper.pdf

[22] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville. 2019. MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/file/6804c9bca0a615bdb9374d00a9fcba59-Paper.pdf

[23] Colin Lea, Michael D. Flynn, René Vidal, Austin Reiter, and Gregory D. Hager. 2017. Temporal Convolutional Networks for Action Segmentation and Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1003–1012. https://doi.org/10.1109/CVPR.2017.113

[24] Li-Juan Liu, Zhen-Hua Ling, Yuan Jiang, Ming Zhou, and Li-Rong Dai. 2018. WaveNet Vocoder with Limited Training Data for Voice Conversion. In *Proc. Interspeech 2018*. 1983–1987. https://doi.org/10.21437/Interspeech.2018-1190

[25] D. Matrouf, J.-F. Bonastre, and C. Fredouille. 2006. Effect of Speech Transformation on Impostor Acceptance. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, Vol. 1. I–I. https://doi.org/10.1109/ICASSP.2006.1660175

[26] Meinard Müller. 2007. *Information retrieval for music and motion*. Springer Berlin, Heidelberg, New York.

[27] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. 2018. WaveGlow: A Flow-based Generative Network for Speech Synthesis. arXiv:1811.00002 [cs.SD]

[28] Daniel Prudký, Anton Firc, and Kamil Malinka. 2023. Assessing the Human Ability to Recognize Synthetic Speech in Ordinary Conversation. In *2023 International Conference of the Biometrics Special Interest Group (BIOSIG)*. 1–5. https://doi.org/10.1109/BIOSIG58226.2023.10346006

[29] Lawrence R. Rabiner and Ronald W. Schafer. 2007. Introduction to Digital Speech Processing. *Found. Trends Signal Process.* 1, 1 (jan 2007), 1–194. https://doi.org/10.1561/2000000001

[30] Ricardo Reimao. 2019. *Synthetic Speech Detection Using Deep Neural Networks*. Master's thesis. York University, Toronto, Ontario.

[31] Ricardo Reimao and Vassilios Tzerpos. 2019. FoR: A Dataset for Synthetic Speech Detection. In *2019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*. 1–10. https://doi.org/10.1109/SPED.2019.8906599

[32] Ricardo Reimao and Vassilios Tzerpos. 2021. Synthetic Speech Detection Using Neural Networks. In *2021 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*. 97–102. https://doi.org/10.1109/SpeD53181.2021.9587406

[33] Philippe Remy. 2020. Temporal Convolutional Networks for Keras. https://github.com/philipperemy/keras-tcn.

[34] Ryosuke Sonobe, Shinnosuke Takamichi, and Hiroshi Saruwatari. 2017. JSUT corpus: free large-scale Japanese speech corpus for end-to-end speech synthesis. arXiv:1711.00354 [cs.CL]

[35] Massimiliano Todisco, Héctor Delgado, and Nicholas Evans. 2016. A New Feature for Automatic Speaker Verification Anti-Spoofing: Constant Q Cepstral Coefficients. In *Proc. The Speaker and Language Recognition Workshop (Odyssey 2016)*. 283–290. https://doi.org/10.21437/Odyssey.2016-41

[36] Massimiliano Todisco, Xin Wang, Ville Vestman, Md. Sahidullah, Héctor Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi H. Kinnunen, and Kong Aik Lee. 2019. ASVspoof 2019: Future Horizons in Spoofed and Fake Audio Detection. In *Proc. Interspeech 2019*. 1008–1012. https://doi.org/10.21437/Interspeech.2019-2249

[37] Xin Wang, Junichi Yamagishi, Massimiliano Todisco, Héctor Delgado, Andreas Nautsch, Nicholas Evans, Md Sahidullah, Ville Vestman, Tomi Kinnunen, Kong Aik Lee, Lauri Juvela, Paavo Alku, Yu-Huai Peng, Hsin-Te Hwang, Yu Tsao, Hsin-Min Wang, Sébastien Le Maguer, Markus Becker, Fergus Henderson, Rob Clark, Yu Zhang, Quan Wang, Ye Jia, Kai Onuma, Koji Mushika, Takashi Kaneda, Yuan Jiang, Li-Juan Liu, Yi-Chiao Wu, Wen-Chin Huang, Tomoki Toda, Kou Tanaka, Hirokazu Kameoka, Ingmar Steiner, Driss Matrouf, Jean-François Bonastre, Avashna Govender, Srikanth Ronanki, Jing-Xuan Zhang, and Zhen-Hua Ling. 2020. ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech. *Computer Speech & Language* 64 (2020), 101114. https://doi.org/10.1016/j.csl.2020.101114

[38] Junichi Yamagishi, Massimiliano Todisco, Md Sahidullah, Héctor Delgado, Xin Wang, Nicolas Evans, Tomi Kinnunen, Kong Aik Lee, Ville Vestman, and Andreas. Nautsch. 2019. ASVspoof 2019: The 3rd Automatic Speaker Verification Spoofing and Countermeasures Challenge database. https://doi.org/10.7488/ds/2555

[39] Junichi Yamagishi, Xin Wang, Massimiliano Todisco, Md Sahidullah, Jose Patino, Andreas Nautsch, Xuechen Liu, Kong Aik Lee, Tomi Kinnunen, Nicholas Evans, and Héctor Delgado. 2021. ASVspoof 2021: accelerating progress in spoofed and deepfake speech detection. arXiv:2109.00537 [eess.AS]

[40] Jining Yan, Lin Mu, Lizhe Wang, Rajiv Ranjan, and Albert Y. Zomaya. 2020. Temporal Convolutional Networks for the Advance Prediction of ENSO. *Scientific Reports* 10, 1 (15 May 2020), 8055. https://doi.org/10.1038/s41598-020-65070-5

[41] Frank Zalkow and Meinard Müller. 2015. Log-Frequency Spectrogram and Chromagram. Retrieved March 1, 2022 from https://www.audiolabs-erlangen.de/resources/MIR/FMP/C3/C3S1_SpecLogFreq-Chromagram.html

[42] Heiga Zen, Yannis Agiomyrgiannakis, Niels Egberts, Fergus Henderson, and Przemysław Szczepaniak. 2016. Fast, Compact, and High Quality LSTM-RNN Based Statistical Parametric Speech Synthesizers for Mobile Devices. In *Proc. Interspeech 2016*. 2273–2277. https://doi.org/10.21437/Interspeech.2016-522

[43] Heiga Zen, Andrew Senior, and Mike Schuster. 2013. Statistical parametric speech synthesis using deep neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 7962–7966. https://doi.org/10.1109/ICASSP.2013.6639215

[44] Zhenyu Zhang, Yewei Gu, Xiaowei Yi, and Xianfeng Zhao. 2020. SynSpeechDDB: a new synthetic speech detection database. https://doi.org/10.21227/ta8z-mx73

[45] Zhenyu Zhang, Yewei Gu, Xiaowei Yi, and Xianfeng Zhao. 2022. FMFCC-A: A Challenging Mandarin Dataset for Synthetic Speech Detection. In *Digital Forensics and Watermarking*, Xianfeng Zhao, Alessandro Piva, and Pedro Comesaña-Alfaro (Eds.). Springer International Publishing, Cham, 117–131.