Voice-based fraud detection with neural networks

Igor Matheus S. Moreira [*1] **João Victor da Silva D. Canavarro** [*1]

Abstract

Widespread and decentralized information dissemination has put the trustworthiness of the received information and its sources in question. Deep-learning-based fake media, also known as Deepfakes, have emerged as a substantial liability, with multiple techniques being able to disguise discourse or engagement in activities as being done by someone else by means of audio, imagery and video. Audio Deepfakes are of particular concern, even more so as models acquire the capability of real-time faking, opening margin for real-time misrepresentation in phone calls. Motivated by this conjecture and aiming at thwarting fraudulent speech, this Deep Learning project proposes implementing a neural network for Deepfake speech detection.

1. Introduction

As advancements in deep-learning-based models are disclosed and productionalized, their implications are made increasingly present in aspects as varied as social influence, psychological ownership, and the privacy calculus (Martin & Zimmermann, 2024). Imagining such capabilities is not new: history is filled with imaginations of virtual humans or artificial intelligence agents embodied in characters such as Pinocchio, in the homonymous story, and Samantha, in the *Her* movie. Also in the movie realm, the implications of mechanisms such as leveraging audio visual cloning as an insurance policy have been speculated as early as 2001 (*vide*, e.g., Beard's analysis on the matter).

Although heralded as a technological feat, Deepfakes also threaten social security given its potential for ill-intended applications, thus motivating an equal attention towards synthesized data detection techniques generated from deep-learning-based methods (Abbas & Taeihagh, 2024). Detecting fake speech in particular is paramount: as put by Bird & Lotfi (2023), humans "use voice as a method of recognizing others in social situations and often go unquestioned"; hence, Deepfakes open margin for unwarranted impersonation, misrepresentation, identity theft, and fraud. The task of detecting synthetically-generated audio is termed Deepfake Speech Detection, or DSD (Pham et al., 2024).

Motivated by this conjecture, this Deep Learning (DL) project proposes the implementation of a neural network architecture geared towards fraud detection by ascertaining whether a given audio sample was synthetically generated. The objective is to leverage lessons accrued throughout the discipline while being mindful of state-of-the-art techniques disclosed in the DSD literature to propose an implementation capable of competently performing this task.

The remainder of this work is divided as follows: section 2 goes through relevant work in the literature of the problem at hand; section 3 further details the architecture of the neural network to be implemented herein; section 4 exposes the experimental setup; section 5 explores attained results and how they stand against previously-proposed architectures; and sections 6 and 7 respectively summarize the findings of this work and enumerate further work.

2. Related work

Deepfake audio holds the potential to propagate misinformation (e.g., credibility defamation of prominent figures) leading to political insecurity and manipulation of public opinion (Chintha et al., 2020; Zhang et al., 2020). Moreover, the rapid growth of Deepfake audio synthesis algorithms also puts voice-enabled devices at risk since the synthesized voices can maliciously take over the control of a device.

The main techniques in speech synthesis are Text-to-Speech (TTS) and Voice Conversion (VC). TTS models take text as input and employ vocoders to produce natural-sounding speech that reflects the linguistic features of the text. Leading TTS algorithms are often based on autoregressive models, such as WaveNet (Van Den Oord et al., 2016) and Tacotron (Wang et al., 2017), or use GAN-based architectures, like HiFi-GAN (Kons et al., 2019).

On the other hand, VC attacks alter original speech to mimic

[*] Equal contribution [1] *Departamento de Ciência da Computação, Instituto de Ciências Exatas, Universidade Federal de Minas Gerais*, Belo Horizonte, MG, Brazil. Correspondence to: Igor Mathueus S. Moreira <igormoreira@ufmg.br>, João Victor da Silva D. Canavarro <jvcanavarro@ufmg.br>.

Required work for the Deep Learning course offered during 2024/2.

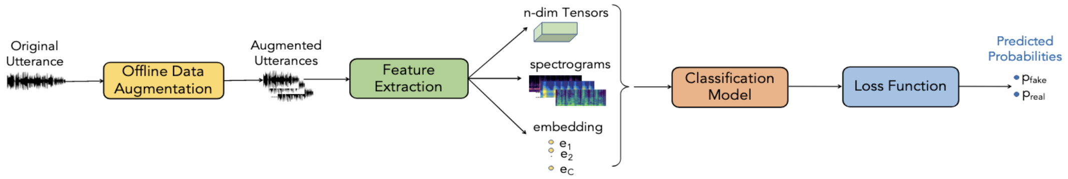

Figure 1. The high-level architecture of a Deepfake Speech Detection system.

the invoice of a specified target speaker while preserving linguistic information. As Deepfake audio threats continue to emerge, a variety of Deepfake audio detection challenges have been developed and played a pivotal role in fostering the development of advanced algorithms to combat such threats (Liu et al., 2023; Yi et al., 2023).

In addition to conventional Machine Learning classifiers, cutting-edge anti-Deepfake audio algorithms predominantly utilize DL architectures, such as Convolutional Neural Networks (CNN) and Residual Networks (ResNet), as classifiers. Certain models integrate multiple DL architectures to enhance performance; for example, RawNet2 (Tak et al., 2021) incorporates a gated recurrent unit (GRU) layer subsequent to its ResNet blocks. Below, we classify these models according to their primary structural configurations.

One of the foundational approaches to Deepfake audio detection involves the extraction of handcrafted features, such as Mel-frequency cepstral coefficients (MFCCs), in combination with traditional classifiers like Support Vector Machines (SVM) and Gaussian Mixture Models (GMM). These early methods established a baseline for detecting audio generated by speech synthesis algorithms; however, their effectiveness diminished as newer, more advanced synthesis models emerged (Gibiansky et al., 2017). Notably, GMM-based classifiers have served as a fundamental baseline technique in the field of Deepfake audio detection.

Regarding the DL field, CNN architectures are also well-known for its effectiveness in capturing local and hierarchical features (Lavrentyeva et al., 2019). In addition, ResNets are one of the significant CNN architecture variants, addressing the vanishing gradient problem in deep networks by incorporating skip connections. ResNets are also widely used in audio classification tasks and achieve relevant outcomes (Tak et al., 2021).

Encoder transformers are often integrated with other DL architectures, such as ResNet or CNNs (Li et al., 2023). The compact convolutional Transformer (Bartusiak & Delp, 2021) incorporates two 2-dimensional convolutional layers before encoders to enhance generalization ability. This strategy aims at extracting high-level embeddings from the input spectrogram rather than directly dividing the spectrogram into patches and feeding them to the Transformer network. Recent advancements have been made to adapt the Transformer to equally treat the temporal and frequency dimensions. Zhang et al. (2023) leverage both the feature matrix and its transposed version to facilitate self-attention mechanism across both temporal and frequency domains. Moreover, attention mechanisms have been integrated in recent research into classifier architectures such as LCNN and ResNet (Ma et al., 2023; Hansen & WANG, 2022).

The use of ensemble methods can be beneficial in overcoming architecture-specific limitations. For example, while CNN-based detectors are commonly employed due to their effectiveness in extracting local patterns, they may encounter difficulties in capturing long-term dependencies. In such scenarios, combining CNNs with Transformer-based models can enhance the system's ability to capture global temporal features. Other architectural combinations comprising a CNN can be observed in the literature: one such example is the work of Wani et al. (2024), which employs it as feature-enriching step prior to ingestion by a Bidirectional Long Short-Term Memory (BiLSTM) network, a variant of the regular LSTM architecture that leverages both past and future input in enhancing its processing of sequential data.

Dataset diversity also plays a crucial role in enhancing model robustness. Large datasets such as ASVspoof (Yamagishi et al., 2019; Liu et al., 2023), specifically designed for Deepfake and synthetic speech detection, have enabled the development of models that generalize better across various synthesis techniques and speaker variations. Transfer learning has also proven effective in the aforementioned scenario, where models pre-trained on large corpora of human speech are fine-tuned on Deepfake audio datasets to capture synthetic characteristics across different generative models and languages. This notwithstanding, the predominant research still lies on the ASVspoof series of datasets, and despite the emergence of newer datasets like ITW (Müller et al., 2022) and WaveFake (Frank & Schönherr, 2021), there remains a substantial gap between these experimental datasets and conditions encountered in daily life.

In terms of other current challenges, not much research has tackled real-time scenarios such as fake phone calls, IoT edge devices, or other low-latency conditions. They require detection models to be computationally efficient. Techniques like model pruning, distributed computing, and real-time incremental learning can be integrated into them.

Research in Deepfake speech detection is expected to keep focusing on improving robustness and scalability of detection methods to keep up with evolving synthesis techniques. This includes addressing challenges such as multilingual Deepfakes, detection of mixed synthetic and real audio, and enhancing real-time detection capabilities to safeguard digital communications against synthetic voice impersonation.

Figure 1 shows the envisioned high-level architecture of the Deepfake speech detector to be implemented herein.

3. Proposed methodology

Based on the audio Deepfake detection pipeline comprised by a CNN and a BiLSTM presented by Wani et al. (2024), we propose a simplified pipeline aiming at achieving similarly competitive performance in audio Deepfake speech detection by only employing a BiLSTM architecture, thus discarding the preceding CNN and its so-touted feature-enriching capabilities. The motivation lies in the belief that the BiLSTM data-processing capabilities suffice to capture relevant speech idiosyncrasies and distinguish fake samples from real ones. Figure 2 depicts the proposed methodology, which are broken down into audio pre-processing, feature normalization, and model training.

3.1. Audio pre-processing

Two pre-processing methods were adopted herein:

- **Normalization:** this is meant to standardize samples of discrepant audio magnitudes into a consistent range between -1 and 1. Normalization mitigates volume discrepancies in favor of an audio content assessment mainly focused on its other traits.

- **Segmentation:** this aims at tackling samples of differing lengths. Each audio file was segmented into three-second-long chunks by either truncating longer samples or appending shorter ones with zeros, thus ensuring a uniform temporal dimension in hopes of promoting a more reliable feature extraction and cross-sample comparisons.

3.2. Feature extraction

Feature extraction ensues by extracting four feature sets from the pre-processed samples: Mel Feature Cepstral Coefficients (MFCC), Mel Spectrograms, Constant Q Transform (CQT), and Constant Q Cepstral Coefficients (CQCC). They are succinctly defined based on Wani et al. (2024). Figure 3 presents a heatmap visualization of two enriched feature sets, MFCCS and Mel Spectrograms.

- **Mel Feature Cepstral Coefficients (MFCC):** this set of coefficients represent the short-term power spectrum of a sound, thus capturing timbral particularities. The i^{th} coefficient comes from the application of a Mel-scale filter bank to the log magnitude spectrum of the signal followed by a direct cosine transform of the log filter bank energies:

$$C_i = \sum_{j=1}^{N} \log(S_j) \cdot \cos\left(i \cdot (j - 0.5) \cdot \frac{\pi}{N}\right),$$

  where S_j is the log energy in the j^{th} Mel-frequency bin and N is the desired number of Mel-frequency bins.

- **Mel Spectrograms:** these are representations in the Mel scale, which is perceptual and meant to portray human response to pitch, thus accentuating idiosyncrasies of human speech. This is obtained in two steps. First, a short-term Fourier transform is performed:

$$\text{STFT}(x(t)) = \int_{-\infty}^{\infty} x(t)w(t-\tau)e^{-j\omega t}dt,$$

  where $x(t)$ is the audio signal, $w(t-\tau)$ is a window function, and $e^{-j\omega\tau}$ facilitates the conversion to the frequency domain. After attaining the power spectrum from the STFT, it is subsequently mapped onto the Mel scale:

$$M(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right).$$

- **Constant Q Transform (CQT):** different from customary Fourier transforms, this set of time-based coefficients have its frequency bins geometrically spaced while the ratio of the center frequency to the bandwidth (i.e., the Q-factor) remains constant. These representations are regarded as more closely resembling human perception of pitch. The k^{th} frequency at the n^{th} time frame can be attained as follows:

$$X_k(n) = \sum_{n=0}^{N-1} x(n)w_k(n)e^{-\frac{j2\pi kn}{Q}}.$$

  $x(n)$ is the audio signal, $w_k(n)$ is a window function, and Q is the quality factor, which determines the resolution and spacing of the frequency bins. $e^{-\frac{j2\pi kn}{Q}}$ has a similar purpose to the exponential component of Fourier transforms, but herein it is adapted to the logarithmic space of CQT.

- **Constant Q Cepstral Coefficients (CQCC):** this coefficient set builds upon CQT by applying two subsequent transformations to robustly represent spectral

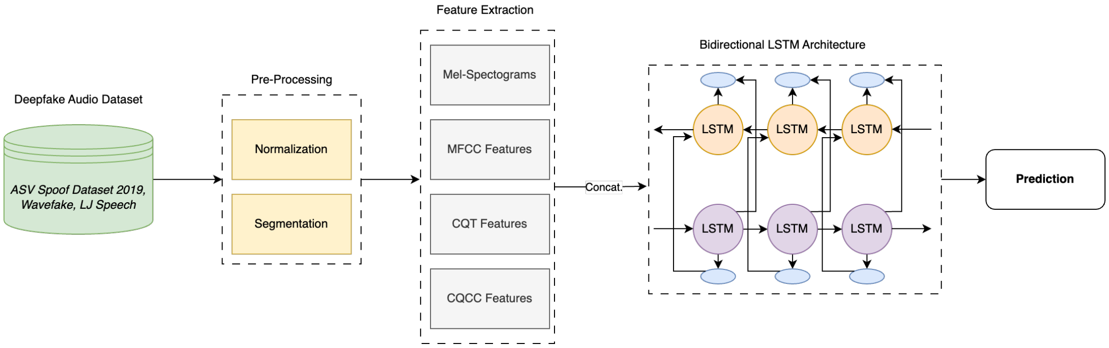

Figure 2. Main components of the proposed DSD system.

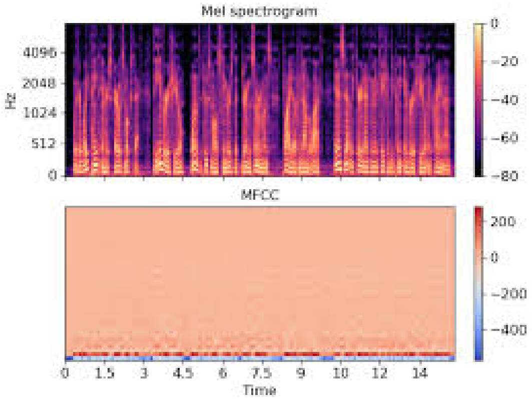

Figure 3. Voice's sample MFCCs and Mel spectrograms visualization (McFee et al., 2015).

traits of the audio signal. The first transformation is the logarithm of the power spectrum:

$$L_k(n) = \log(|X_k(n)|^2),$$

where $\log(|X_k(n)|^2)$ represents the power spectrum of the CQT. Then, a direct cosine transforms it into the cepstral domain:

$$C_m = \sum_{k=1}^{K} L_k(n) \cdot \cos\left(\frac{\pi m}{K}(k - 0.5)\right),$$

where C_m denotes the m^{th} cepstral coefficient, K is the number of bins adopted in the CQT, and m ranges from 1 to an arbitrary upper limit.

3.3. Model training

The architecture proposed herein is a simplified take at the work of Wani et al. (2024). This simplification removes the CNN responsible for feature enriching, as well as the subsequent Principal Component Analysis employed for dimensionality reduction. This means that extracted features are directly provided to the BiLSTM network under the assumption that its modeling capabilities competently fit idiosyncrasies of Deepfake and real speech, thus being able to distinguish between samples without further feature extraction steps other than those outlined above.

The BiLSTM architecture remains the same: three layers of 128 hidden units each, followed by a classification layer whose output predicts whether the provided sample is fake.

4. Experimental set-up

This section focuses on describing employed datasets and the experiments *per se*, as well as hyper-parameters and evaluation criteria. Each dimension will be further elaborated upon below.

4.1. Datasets

Three datasets are used herein:

- **LJ Speech (Ito & Johnson, 2017):** a public domain dataset comprising more than thirteen thousand short audio clips of a real female speaker reading passages from seven non-fiction books published between 1884 and 1964. Clips range from one to ten seconds, roughly totaling 24 hours, or roughly 3 GB in size.

- **ASVspoof 2019 (Yamagishi et al., 2019):** a dataset used as part of a challenge. It contains *bona fide* utterances from a database termed VCTK from 107 speakers (46 male and 61 female). Fake utterances were made from these *bona fide* ones by a variety of methods, totaling around 4 GB in size.

- **WaveFake (Frank & Schönherr, 2021):** a data set for audio Deepfake detection, composed of over a hundred thousand fake audio clips generated via multiple Deepfake methods totaling 175 hours, or roughly 29 GB in size.

4.2. Experiments

The proposed pipeline was put to test in two dataset configurations. In the first, only the ASVspoof 2019 dataset was employed; in the second, LJ Speech and WaveFake were jointly used. This results in two datasets containing both real and spoofed samples. A train-validation-test split was made in a stratified fashion following a 70-15-15 proportion for the respective partitions, thus resulting in data sets with balanced classes.

Due to the imbalance between real and fake audio classes in the ASVspoof dataset, the dataset was subject to resampling prior to training, resulting in a final dataset comprising approximately 26,000 entries. This is a relevant difference from the original proposal, as it did not balance the data prior to training and evaluation. Similarly, the LJ Speech and WaveFake datasets were also combined, resulting in nearly 25,000 examples in the final dataset.

Each of the previously described feature sets – MFCCs, Mel spectrograms, CQTs, and CQCCs – yielded 15 features each, resulting in a total of 60 enriched features extracted from a single audio signal.

Finally, all experiments were conducted in an environment equipped with four NVIDIA A100-SXM4-40GB GPUs, leveraging TensorFlow's multi-GPU capabilities to optimize training time. Initial attempts were made to employ 16-bit mixed-precision floating point operations *in-traning*; however, due to technical difficulties, this plan was not fully implemented.

4.3. Hyper-parameters and evaluation criteria

Relevant hyper-parameters were reused from the work of Wani et al. (2024). Specifically, the BiLSTM model was trained with a batch size of 32 and training was carried out over 50 epochs. The Adam optimizer was employed and cross-entropy was set as the loss function. Likewise, the learning rate was set to 0.001. Likewise, the accuracy and equal error rate (EER) used as performance evaluation criteria therein will also be used herein.

Early stopping techniques were also evaluated; however, the results achieved with this approach were surpassed by those obtained through full-length training in both scenarios.

5. Results and discussion

This section outlines the primary results achieved by the models trained for the Deepfake Audio Detection task. It also compares these performance metrics to those of relevant studies in the literature, demonstrating that the proposed work can attain competitive results despite employing a less complex and parameter-efficient architecture.

A summary of the findings can be found in table 1, which compares our results against those of relevant counterparts as reported in the literature.

The first three lines represent the previously defined benchmarks, while the final two lines present the metrics achieved by this study. It is evident that employing a simpler model, without incorporating an ensemble approach, did not result in substantial performance degradation. The model trained on the ASVspoof dataset achieved an accuracy of 92.89% and an Equal Error Rate (ERR) of 1.6% on the test set, indicating consistent performance despite utilizing a more compact architecture with fewer parameters.

Conversely, the model trained on the LJ Speech and WaveFake datasets yielded highly satisfactory outcomes, with an accuracy of 97.22% and an ERR of 0.30%. It is important to highlight that this dataset may be less representative of real-world scenarios, given that the recordings are of high quality and contain minimal noise.

Following subsections why dive into the individual performance of both models throughout training.

5.1. ASVspoof 2019

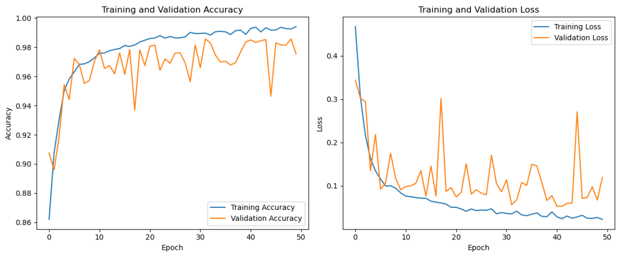

Figure 4. BiLSTM accuracy and loss on both training and validation sets.

Initially, the proposed model was trained using the primary evaluation dataset, ASVspoof 2019. Figure 4 illustrates the performance in terms of accuracy and loss across the training epochs for the training and validation datasets. Convergence is observable within the initial epochs, likely attributable to the relatively high learning rate suggested by the original work. This hypothesis is further supported by the validation metrics, which display reduced stability and loss peaks throughout the epochs.

As a disclaimer, we were unable to train the network using TensorFlow's parallelization capabilities, unlike the training conducted for the second model.

5.2. LJ Speech + WaveFake

Training the architecture with the second dataset resulted in more stable learning. Figure 4 shows the accuracy and loss curves across the training epochs, illustrating that the model

Table 1. Performance comparison against relevant counterparts. Missing values are due to lack of their report in the corresponding paper.

Study	Dataset	Features	Classifier	Accuracy	EER%
Wang et al. (2020)	ASVspoof 2019	CQCC, LFCC, Spec	Densely connected CNN	-	1.40%
Xue et al. (2023)	ASVspoof 2019	CQCC, MFCC, LFCC, Face	DenseNet	-	2.82%
Wani et al. (2024)	ASVspoof 2019	MFCC, Mel spectrogram, CQT, CQCC	CNN + BiLSTM	96.63%	0.074%
Proposed Approach	ASVspoof 2019	MFCC, Mel spectrogram, CQT, CQCC	BiLSTM	92.89%	1.60%
Proposed Approach	LJ + WaveFake	MFCC, Mel spectrogram, CQT, CQCC	BiLSTM	97.22%	0.30%

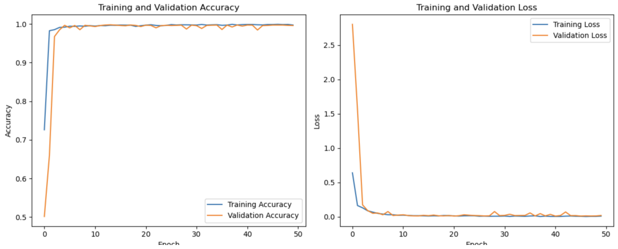

Figure 5. BiLSTM accuracy and loss on both training and validation sets.

converged more effectively throughout the entire training period, outperforming the first model.

This behavior can be correlated to several factors, including cleaner and higher quality datasets, balanced data, and alignment with the architecture being used. Further investigations can be conducted to analyze the enriched features of used feature sets such as MFCCs.

6. Conclusion

With the enhanced generative capabilities of contemporary AI models, there has been significant growth in using them to develop novel methods for scams and fake news dissemination. Among these methods, utilizing deepfake voice clips to manipulate targets has become particularly prominent. In response to this challenge, this work introduces an optimized model architecture specifically designed for the task of detecting deepfake audio.

The trained models demonstrated competitive performance compared to other works in the literature, achieving accuracy metrics very close to those of more complex ensemble-based architectures. Additionally, the model was successfully trained with two other datasets, maintaining consistent results and showcasing its ability to adapt to inputs with varying levels of quality. This adaptability highlights the model's robustness and versatility in handling diverse audio datasets.

This work was developed as the final project deliverable for the Deep Learning course taught at the Federal University of Minas Gerais (UFMG). By leveraging the theoretical knowledge and practical expertise gained during the semester, the project aimed at creating an effective and optimized model architecture capable of addressing challenges posed by advanced AI-generated audio manipulation.

Finally, the source code for this project, along with the trained model weights, is available on GitHub[1].

7. Further work

As promising avenues to further this work, the following aspects are highlighted:

- New feature sets and experimentation with existing ones could be performed towards promoting an optimal representation of the leveraged audio samples.
- Other state-of-the-art architectures (e.g., Transformers or architectures with attention mechanisms, combinations such as CNNs + Transformers) could be compared in search of other competitive model architectures that tackle this problem context;
- Gains derived from more training and/or training with more data could be investigated;
- Fine-tuning a foundational model could be explored to leverage larger architectures for deepfake audio detection. By adapting such a model, this approach can utilize its generalized knowledge and transfer learning benefits to enhance performance and accuracy in this specific context;
- Experiments of performance on edge devices can be explored for assessing the model's feasibility with limited computational resources; and

[1] https://github.com/jvcanavarro/deepfake-audio-detection-DL-UFMG

- Evaluation with more recent datasets, such as ASVspoof 2024, could provide valuable insights into the model's robustness and ability to cope with the latest advancements in deepfake technology.

References

Abbas, F. and Taeihagh, A. Unmasking deepfakes: A systematic review of deepfake detection and generation techniques using artificial intelligence. *Expert Systems with Applications*, 252:124260, 2024. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2024.124260. URL https://www.sciencedirect.com/science/article/pii/S0957417424011266.

Bartusiak, E. R. and Delp, E. J. Synthesized speech detection using convolutional transformer-based spectrogram analysis. In *2021 55th Asilomar Conference on Signals, Systems, and Computers*, pp. 1426–1430. IEEE, 2021.

Beard, J. J. Clones, bones and twilight zones: protecting the digital persona of the quick, the dead and the imaginary. *J. Copyright Soc'y USA*, 49:441, 2001.

Bird, J. J. and Lotfi, A. Real-time detection of ai-generated speech for deepfake voice conversion. *arXiv preprint arXiv:2308.12734*, 2023.

Chintha, A., Thai, B., Sohrawardi, S. J., Bhatt, K., Hickerson, A., Wright, M., and Ptucha, R. Recurrent convolutional structures for audio spoof and video deepfake detection. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):1024–1037, 2020.

Frank, J. and Schönherr, L. Wavefake: A data set to facilitate audio deepfake detection. *arXiv preprint arXiv:2111.02813*, 2021.

Gibiansky, A., Arik, S., Diamos, G., Miller, J., Peng, K., Ping, W., Raiman, J., and Zhou, Y. Deep voice 2: Multi-speaker neural text-to-speech. *Advances in neural information processing systems*, 30, 2017.

Hansen, J. H. and WANG, Z. Audio anti-spoofing using simple attention module and joint optimization based on additive angular margin loss and meta-learning. *Interspeech 2022*, Sep 2022. doi: 10.21437/interspeech.2022-904.

Ito, K. and Johnson, L. The lj speech dataset. https://keithito.com/LJ-Speech-Dataset/, 2017.

Kons, Z., Shechtman, S., Sorin, A., Rabinovitz, C., and Hoory, R. High Quality, Lightweight and Adaptable TTS Using LPCNet. In *Proc. Interspeech 2019*, pp. 176–180, 2019. doi: 10.21437/Interspeech.2019-1705.

Lavrentyeva, G., Novoselov, S., Tseren, A., Volkova, M., Gorlanov, A., and Kozlov, A. Stc antispoofing systems for the asvspoof2019 challenge. *Interspeech 2019*, Sep 2019. doi: 10.21437/interspeech.2019-1768.

Li, M., Ahmadiadli, Y., and Zhang, X.-P. Robust deepfake audio detection via bi-level optimization. In *2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6. IEEE, 2023.

Liu, X., Wang, X., Sahidullah, M., Patino, J., Delgado, H., Kinnunen, T., Todisco, M., Yamagishi, J., Evans, N., Nautsch, A., et al. Asvspoof 2021: Towards spoofed and deepfake speech detection in the wild. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

Ma, K., Feng, Y., Chen, B., and Zhao, G. End-to-end dual-branch network towards synthetic speech detection. *IEEE Signal Processing Letters*, 30:359–363, 2023.

Martin, K. D. and Zimmermann, J. Artificial intelligence and its implications for data privacy. *Current Opinion in Psychology*, 58:101829, 2024. ISSN 2352-250X. doi: https://doi.org/10.1016/j.copsyc.2024.101829. URL https://www.sciencedirect.com/science/article/pii/S2352250X24000423.

McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., and Nieto, O. librosa: Audio and music signal analysis in python. In *SciPy*, pp. 18–24, 2015.

Müller, N. M., Czempin, P., Dieckmann, F., Froghyar, A., and Böttinger, K. Does audio deepfake detection generalize? *arXiv preprint arXiv:2203.16263*, 2022.

Pham, L., Lam, P., Nguyen, T., Tang, H., Nguyen, H., Schindler, A., and Vu, C. A comprehensive survey with critical analysis for deepfake speech detection. *arXiv preprint arXiv:2409.15180*, 2024.

Tak, H., Patino, J., Todisco, M., Nautsch, A., Evans, N., and Larcher, A. End-to-end anti-spoofing with rawnet2. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6369–6373. IEEE, 2021.

Van Den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K., et al. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 12, 2016.

Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q., Agiomyrgiannakis, Y., Clark, R., and Saurous, R. A. Tacotron: Towards End-to-End Speech Synthesis. In *Proc. Interspeech 2017*, pp. 4006–4010, 2017. doi: 10.21437/Interspeech.2017-1452.

Wang, Z., Cui, S., Kang, X., Sun, W., and Li, Z. Densely connected convolutional network for audio spoofing detection. In *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1352–1360. IEEE, 2020.

Wani, T. M., Qadri, S. A. A., Comminiello, D., and Amerini, I. Detecting audio deepfakes: Integrating cnn and bilstm with multi-feature concatenation. In *Proceedings of the 2024 ACM Workshop on Information Hiding and Multimedia Security*, pp. 271–276, 2024.

Xue, J., Zhou, H., Song, H., Wu, B., and Shi, L. Cross-modal information fusion for voice spoofing detection. *Speech Communication*, 147:41–50, 2023.

Yamagishi, J., Todisco, M., Sahidullah, M., Delgado, H., Wang, X., Evans, N., Kinnunen, T., Lee, K. A., Vestman, V., and Nautsch, A. Asvspoof 2019: The 3rd automatic speaker verification spoofing and countermeasures challenge database. 2019.

Yi, J., Tao, J., Fu, R., Yan, X., Wang, C., Wang, T., Zhang, C. Y., Zhang, X., Zhao, Y., Ren, Y., et al. Add 2023: the second audio deepfake detection challenge. *arXiv preprint arXiv:2305.13774*, 2023.

Zhang, J., Tu, G., Liu, S., and Cai, Z. Audio anti-spoofing based on audio feature fusion. *Algorithms*, 16(7):317, 2023.

Zhang, T., Deng, L., Zhang, L., and Dang, X. Deep learning in face synthesis: A survey on deepfakes. In *2020 IEEE 3rd International Conference on Computer and Communication Engineering Technology (CCET)*, pp. 67–70. IEEE, 2020.