



**HAL**  
open science

## Detection of Deepfake Environmental Audio

Hafsa Ouajdi, Oussama Hadder, Modan Tailleur, Mathieu Lagrange, Laurie  
M Heller

► **To cite this version:**

Hafsa Ouajdi, Oussama Hadder, Modan Tailleur, Mathieu Lagrange, Laurie M Heller. Detection of Deepfake Environmental Audio. EUSIPCO, 2024, Lyon, France. hal-04603448v2

**HAL Id: hal-04603448**

**<https://hal.science/hal-04603448v2>**

Submitted on 13 Jun 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Detection of Deepfake Environmental Audio

Hafsa Ouajdi  
École Centrale Nantes  
Nantes, France  
hafsaouajdimp@gmail.com

Oussama Hadder  
École Centrale Nantes  
Nantes, France  
natsuhadder001@gmail.com

Modan Tailleur, Mathieu Lagrange  
Nantes Université,  
École Centrale Nantes,  
CNRS, LS2N, UMR 6004,  
F-44000 Nantes, France  
first.last@ls2n.fr

Laurie M. Heller  
Dept. of Psychology,  
Carnegie Mellon University,  
Pittsburgh, PA, U.S.  
hellerl@andrew.cmu.edu

**Abstract**—With the ever-rising quality of deep generative models, it is increasingly important to be able to discern whether the audio data at hand have been recorded or synthesized. Although the detection of fake speech signals has been studied extensively, this is not the case for the detection of fake environmental audio. We propose a simple and efficient pipeline for detecting fake environmental sounds based on the CLAP audio embedding. We evaluate this detector using audio data from the 2023 DCASE challenge task on Foley sound synthesis.

Our experiments show that fake sounds generated by 44 state-of-the-art synthesizers can be detected on average with 98% accuracy. We show that using an audio embedding trained specifically on environmental audio is beneficial over a standard VGGish one as it provides a 10% increase in detection performance. The sounds misclassified by the detector were tested in an experiment on human listeners who showed modest accuracy with nonfake sounds, suggesting there may be unexploited audible features.

**Index Terms**—Fake detection, Environmental sound, Deep learning, Classification, Deepfake audio

## I. INTRODUCTION

The rapid evolution of generative models, such as those based on Diffusion models, ushers in an era where the boundaries between reality and synthetic content tend to blur more and more. In audio synthesis, there exists a sizable literature dedicated to the detection of deep fakes in speech, aiming to detect adversarial attacks ranging from misinformation dissemination to identity theft [1].

As generative models become more sophisticated, particularly those rooted in deep learning architectures, their capacity to produce eerily realistic audio forgeries has grown exponentially. Innovations such as Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) have empowered malicious actors to craft audio content that is virtually indistinguishable from genuine recordings. Furthermore, deepfake audio is often used in conjunction with deepfake video to create more realistic and convincing fake movies. In light of these trends, the design of effective fake audio detection systems is important.

Deepfake audio refers to audio that has been generated or augmented using deep learning techniques [2]. There are different types of Deepfake Audio: Text-to-Speech [3], Voice conversion [4], emotion fakes [5], scene fakes [6], and partial fakes [7].

While deepfake detection for speech is well studied, there appears to be little research on the detection of fake content for environmental sounds. This paper aims to fill this gap with respect to Foley sound synthesis, *i.e.*, sound categories such as *hand clap*, *rain*, etc.

We propose a simple and effective fake detection pipeline based on CLAP embeddings [8]. Experiments are based on a publicly available audio dataset developed during Task 7 of the IEEE AASP DCASE 2023 Challenge [9]. This is a dataset of more than 6 hours of recorded audio and 28 hours of generated audio.

The paper is organized as follows. Section II gives a brief review of the relevant state of the art on deepfake detection in audio. Section III introduces the proposed deepfake detector which is benchmarked using an experimental protocol described in Section IV. Performance is discussed in Section V and Section VI discusses the outcome of listening tests on detection mistakes of the proposed detector and directions for improvement. Code, supplementary material, and audio examples are available on the companion page.<sup>1</sup>

## II. RELATED WORK

This section discusses machine-learning techniques for Deepfake Audio detection. In [10], the authors use linear regression to detect fake voices. The features are extracted using a signal’s entropy calculated based on Shannon’s equation for each second of the signal and for the whole signal. The model reaches 98% accuracy, with all fake audio being correctly detected. In [11], the authors adopt a combination of Bispectral analysis and MFCC (Mel Frequency Cepstral Coefficients) for speech detection and feature extractions. The former helps to identify components in generated audio that are not present in human speech and the latter is used to detect the features in human speech tied to the vocal tract that are absent in the AI-generated speech. The features are then fed into seven different ML algorithms. The results show that the Quadratic SVM model gives the highest accuracy of 96.1%. In [12], the feature vector is extracted by filtering the signal under analysis and extracting different statistics with short-term and long-term prediction order. The features are then fed to a Linear SVM and Random Forest model. The results show that the

<sup>1</sup>Companion page: <https://mathieulagrange.github.io/audioFoleyDeepFake>

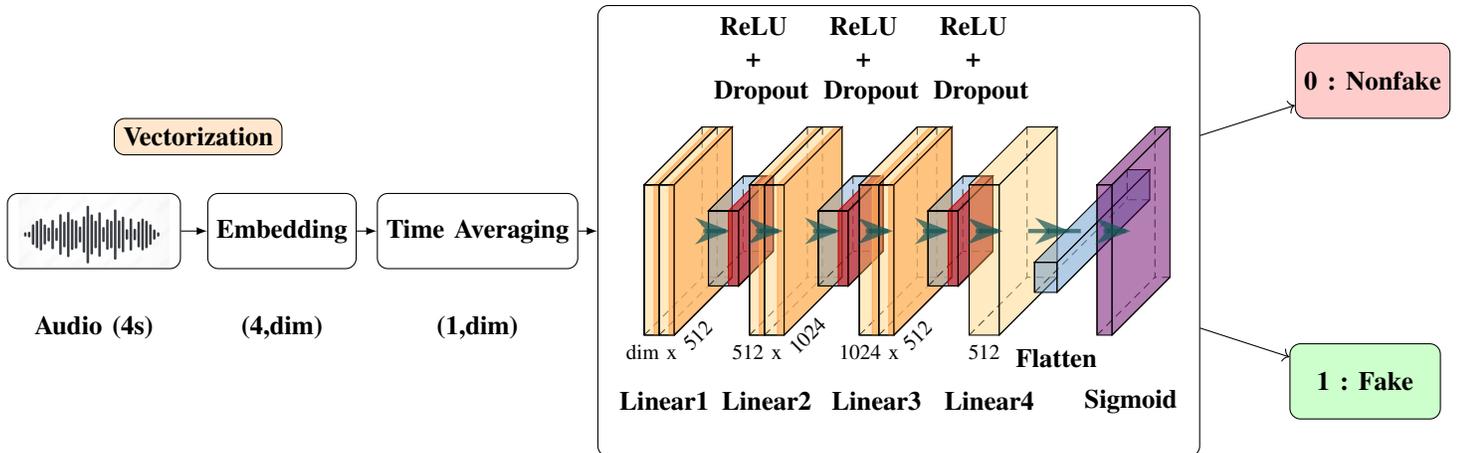


Fig. 1: Overview of the pipeline used in the experiments for the Deepfake detection, with a representation of the MLP’s network architecture. The value of *dim* depends on the embedding method used.

SVM outperforms the RF in different conditions.

Traditional machine-learning models require manual feature extraction. Consequently, research was directed towards more sophisticated feature extraction algorithms based on deep learning models.

In [13], they employed both CNNs (Convolutional Neural Net), using local dependencies, and LSTMs (Long Short-Term Memory), using both local and sequential dependencies, to classify audio based on MFCCs. The former model helps achieve an accuracy of 80%. A CNN has been used with a similar strategy in [14], which compares different approaches (MFCC, STFT, FFT of the signal) to deduce that MFCC gives the best results and also that CNNs are beneficial in such tasks because they can learn to detect important features. In [15], a shallow CNN architecture, Deep2Net, was used. The authors transformed the audio classification task into a computer vision problem by using a histogram of the audio. The model reached a global accuracy of 0.985.

### III. PROPOSED APPROACH

The detection of fake sounds is treated as a binary classification task. As can be seen in Figure 1, the proposed architecture for solving this task leverages the power of pre-trained audio embeddings. This allows us to simplify the learning process by requiring less training data and using less power as the learnable part of the detector consists of three dense layers, as shown in Section 1. Each layer is followed by 1) ReLU activation functions to reduce the backpropagation errors and accelerate the learning process, and 2) dropout layers for regularization to avoid overfitting. The final dense layer is followed by a sigmoid function as the output activation function.

To provide input to the decision module, we tested 3 embeddings produced by deep learning architectures, namely: VGG, CLAP, and PANN.

The VGG embedding is produced by the VGGish network,

a convolutional neural network pre-trained for audio classification tasks, and is adapted from the VGG [16] image classification architecture. This method adapts the VGG image classification network to the audio domain by converting audio signals into log-mel spectrograms, which are visual representations of the frequency and amplitude of the sound. The spectrograms are then fed into the network, which was pre-trained on a large-scale YouTube dataset with diverse audio categories, such as music, speech, or animal sounds. The CLAP embedding is produced by the Contrastive Language-Audio Pretraining (Clap) model [17]. By leveraging a contrastive learning approach, CLAP enables models to learn representations that encode the semantic content of language and the acoustic characteristics of audio simultaneously. During pre-training, these models are trained using contrastive learning objectives to project both language and audio inputs into a shared embedding space. The model learns to encode similar language-audio pairs closely together while pushing dissimilar pairs apart. This process enables the extraction of rich embeddings that capture semantic and acoustic similarities between inputs.

The PANN embedding is produced by Pretrained Audio Neural Networks (PANNs) [18], which are yet another class of models. These networks are trained on spectrogram representations of audio signals that capture both the frequency and temporal characteristics of sound. The pre-trained models are usually initialized with weights from models pre-trained on large-scale audio datasets, such as AudioSet or UrbanSound, which contain diverse audio categories ranging from musical genres to environmental sounds.

## IV. EXPERIMENTS

### A. DATASET

The **DCASE2023 Challenge** [19] aimed to produce fake environmental sounds by training on nonfake sounds. It consisted of two tracks: Track A, with 10 generation systems

from participating teams using varied Foley sound synthesis methods and permitting limited use of external resources; and Track B, with 28 generation systems based on provided code, prohibiting external resources. Both tracks accepted rule-based and ensemble systems, provided they utilized sounds only from the development set.

The resulting dataset consists of both nonfake and fake sounds across seven distinct sound classes (*dog\_bark*, *foot-step*, *gunshot*, *keyboard*, *moving\_motor\_vehicle*, *rain*, and *sneeze\_cough*). Specifically, the dataset comprises 5,550 non-fake sounds alongside 25,200 fake sounds, the latter of which were generated by challenge participants. These sounds are analyzed through the lens of four different embeddings, detailed in section II.B, each with its specific dimensionality: VGGish embeddings at (4,128), Microsoft CLAP (MS-Clap) embeddings at (4,1024), and two PANNs embeddings at (4,2048), where '4' refers to the four-second duration of the audio clips. For training and evaluation purposes, we separate the dataset into a training set (70%), a validation set (10%), and an evaluation set (20%)<sup>2</sup>.

## B. TRAINING PROCEDURE

Figure 1 shows the pipeline used in the experiment protocol. As we have four different input embeddings, the procedure yielded four models to use and compare: VGGish, MS-Clap, PANN Wavegram, and PANNcnn14-32k. Note that we take the time average of each embedding to avoid the temporal sequence aspect (four-second duration) affecting the deepfake detection.

These models were trained using the Adam optimizer, configured with a learning rate of  $7e-4$ . The training protocol was standardized across models, with a batch size of 128 samples, a decision threshold of 0.5 on both training and validation predictions, and training over 100 epochs. Throughout this process, the Binary Cross-Entropy (BCE) loss function given in Equation 1 is used to optimize the weights parameters in the backpropagation process:

$$J(y, \hat{y}) = -(y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y})) \quad (1)$$

Where  $y$  is the true label (0 or 1) and  $\hat{y}$  is the predicted probability.

During training, we implement a checkpointing mechanism that saves the state of the model every 10 epochs of training. Following the best-accuracy selection criterion [20], the final model is chosen based on the checkpoint demonstrating the highest validation accuracy. For each implementation of the detector, 10 training runs are performed and statistics are reported. We carry out all the experiments on a machine with 4 physical cores and 8 threads, 16GB of RAM, and an NVIDIA GeForce RTX 3060 graphic card with 6GB of RAM. To address the imbalance between fake and nonfake samples, we employ a balancing technique on the training set.

<sup>2</sup>Dataset of recorded and generated audio: <https://zenodo.org/records/8091972>

## V. RESULTS

### A. INFERENCE TIME

To evaluate the computational efficiency of each model, we assessed the time required for the embedding process and the model's inference for each embedding on one random audio sample across 100 runs. The inference time (in seconds) is then computed by averaging over the 100 inference times. In Table II, the averaged inference time is expressed as a percentage of the real-time duration of the audio sample. This metric provides a clear understanding of the model's speed relative to the actual audio duration.

### B. OVERALL ACCURACY

We calculate both the relative overall validation and evaluation accuracy, along with the time per epoch in seconds (s), and generate a Confusion Matrix for the evaluation dataset. As presented in Table II, statistics such as Overall Evaluation Accuracy and Training Time per Epoch are detailed for each model. Among the experiments conducted under the same conditions, the MS-Clap model outperformed others with the highest Evaluation Accuracy of 98.02%, surpassing both PANN models by 5% and the VGGish model by 10%.

Among the 40 identical runs, we pick the best-performing run for each model (*i.e.*, the highest evaluation accuracy). The confusion matrices resulting from the four chosen models demonstrate a varied performance across the different architectures. The VGGish model shows difficulty with correctly classifying nonfake sounds, as it produces 9% false positive judgements. The MS-Clap model shows the best proficiency in identifying the nonfake category of sounds, with the lowest false prediction rate (2% of the dataset). On the other hand, the PANN-Wavegram and the PANN\_32k models behave approximately the same by having a high evaluation accuracy and also a low false prediction rate, yet they do not perform as well overall as the Clap model.

### C. STATISTICAL ANALYSIS

We test the four models in the same configurations as shown in Table II. We train the models under each configuration for 10 iterations. We then perform the Mann-Whitney U-test to assess the similarity of accuracy distribution. The U-test is adopted instead of the t-test since it doesn't impose the assumption of normal distribution. In our case, the Null hypothesis assumes that run A is similar to run B. If the p-value  $< 0.05$ , we can confirm that the alternative hypothesis is true (run A is similar to run B) with a confidence interval of 95%.

The accuracy of MS-Clap is higher than that of Pann-Wavegram with 5.02%, and that of PANN 32k with 5.11%, and that of VGGish with 10.23%. In all cases, the U-test confirms that those differences are significant, as for all pairs the Null hypothesis is rejected.

### D. CLASS-WISE ACCURACY

As the MS-Clap model is the best-performing model among the four trained models, we chose to carry out our next analysis

Predicted	VGGish		MS-Clap		PANN_Wavegram_logmel		PANN_32k	
	Nonfake (%)	Fake (%)	Nonfake (%)	Fake (%)	Nonfake (%)	Fake (%)	Nonfake (%)	Fake (%)
Nonfake	15	3	17	1	14	3	14	5
Fake	9	73	1	81	4	79	3	77

TABLE I: Confusion Matrix Evaluation for the three embeddings

TABLE II: Statistics on Overall Accuracy and the Inference Time expressed as a real-time percentage.

Model x Embeddings	Accuracy (%)	Inference Time (%)
	Mean	Averaged percentage
VGGish	88.11 $\pm$ 0.73	0.423
MS-Clap	<b>98.02</b> $\pm$ 0.18	1.82
PANN-Wavegram	93.15 $\pm$ 0.34	0.318
PANNcnn14_32k	93.04 $\pm$ 0.32	0.234

with this model. To have a better understanding of the behavior of the model’s performance, we look closely at the model’s accuracy for each class (Table III). This closer look helps us understand how the model performed with different types of data. We find that the model was consistently good at identifying all classes (98% approximately for each class), with *dog\_bark* being the class where the model performs the best, and it performs slightly worse than the others when it has to predict on *keyboard* and *sneeze\_cough* classes.

### E. ACCURACY VS. GENERATOR QUALITY

We now compare our classifier’s performance against the outcomes of the DCASE2023 challenge task 7. For this purpose, we conduct a comparison and analysis between the scores of our predicted likelihoods by using the BCE as a score function (the higher, the better), and the Fréchet Audio Distance (FAD) scores [19] used for the official rankings of the challenge (the lower, the better).

We observe a significant difference in correlation across the two tracks. While Track A demonstrated a strong negative correlation of -0.86, indicating a general agreement between the two scoring methods, Track B showed a notably lower correlation of -0.27. This discrepancy may be due to two major differences between the two tracks: Track B imposed restrictions on external resources, whereas Track A had none, and Track B had more systems than Track A (28 versus 10). Furthermore, the low correlation in Track B could be due to the diversity, complexity, and originality of the features used in the generating systems. For instance, feature extraction methods such as log-mel spectrogram and log-magnitude spectrogram were used by Track B’s systems, while in Track A the majority used Gaussian latent variables and a spectrogram for feature extraction; additionally, some data augmentation strategies such as time masking, tanh distortion, and sound wrapping were used in Track B while being unused in Track A.

## VI. LISTENING TEST

Human listening suggests opportunities for improvement in the detection of distortion, realistic echoes and temporal structure, and patterns of noise and repetition which may be noticeable

to humans but not to a frame-based detector such as the one considered in this study.

### A. DATA

#### *Incorrect Positives*

Erroneously classifying a nonfake sound as a fake is an "Incorrect Positive" whereas classifying a fake sound as a fake is a "Correct Positive". We examined 38 Incorrect Positive sounds, spanning all 7 categories, based on the erroneous high likelihood of the MS-Clap model. These sounds are from the DCASE development and evaluation dataset that were presumed to be real recordings; however, although these sounds were aurally screened by the organizers, the datasets may have nonetheless contained a few Foley or processed sounds. To detect this possibility and also to detect other reasons the system incorrectly classified a sound as fake, all authors carefully listened to these sounds. We noted two recordings that had very strong echoes. We suspected that two recordings were Foley-simulated footsteps rather than recordings of real footsteps. Some recordings included non-target sounds (e.g. footsteps during rain, or a snore after a sneeze). Many sounds had significant background noise. We verbally characterized five sound features that may have triggered an incorrect fake detection and used them to create a menu of "reasons" for a person suspecting that an audio is fake.

#### *Incorrect Negatives*

The 43 fake sounds that were erroneously classified as nonfake by the MS-Clap detector are referred to as "Incorrect Negatives". There were sounds from all 7 categories. We noted a couple of recordings that contained very little sound (perhaps a brief thud) and so there was not much available acoustic information. Many sounds had significant background noise, and often the noise sounded more like static than the typical noise heard in real recordings, such as environmental noise or distortion from overloading a microphone (clipping/saturation). Sometimes the onset of the target sound contained a burst of noise. One sound had audible repetition; evidence of splicing/editing was confirmed via a spectrogram.

### B. METHODS

Twenty CMU undergraduates participated for academic credit (Mean age=19.8; 9 female, 10 male, 1 nonbinary) who reported normal hearing and passed a binaural screening survey to ensure careful listening over headphones. They used an online platform (gorilla.sc) and gave online consent approved by CMU’s IRB. Each sound was accompanied by its category name on the screen (e.g. "dog bark"), and each sound was played once before a judgement was made. All 7 sound

Sound Class	dog_bark	footstep	gunshot	keyboard	moving_motor_vehicle	rain	sneeze_cough
Accuracy (%)	99	98.1	98.3	97.7	98.1	98.5	97.7

TABLE III: Per Class Accuracy of the MS-Clap deepfake predictor.

categories from the System’s test set were represented in the set of 43 fake and 38 real sounds on which the ML system made errors. Participants were instructed to respond “real” if the sound seemed like a real recording of an event, and to respond “fake” if it seemed like it was generated by a computer program. They were told that the existence of background noise was not an indicator of fakeness and that cues could be present in either the sound from the category or the background noises.

### C. OUTCOMES

When a fake (model-generated) sound was missed by the ML system (a system incorrect negative), the percent of sounds that listeners judged as fake (listener correct positives,  $M=49\%$ ) was not significantly greater than chance,  $z=-0.07$ ,  $p=0.6$ . When a nonfake (recorded) sound was judged as fake by the ML system (system incorrect positive), the percent of sounds that listeners judged as real (listener correct negatives,  $M=71\%$ ) was greater than chance,  $z=1.34$ ,  $p < 0.0001$ . This human sensitivity advantage implies that some acoustic features which indicated fakeness to the ML system did not seem artificial to human listeners. When a sound was judged as fake by a participant, they indicated one of six possible reasons in the following proportions: The audio of the target sound is distorted (underwater, wobbly): 34.2% The sound does not sound like an example of the target category. 32.2% I hear suspicious artifacts in the background (whining, etc.): 12.9%. The sound is too brief or quiet to be able to give a reason. 10.0% The sound is too repetitive. 6.8% Other: 4%.

### VII. CONCLUSION

We documented the first study of deep fake detection for environmental audio generation. The experiments reported considered a public dataset of sounds generated by 10 state of the art systems and another public dataset of environmental audio recordings.

The excellent results of the proposed deep fake detector demonstrate that, to this day, generated sounds can be detected rather easily. The listening test provide some interesting cues that can suggest research directions for the improvement of the generators.

### VIII. ACKNOWLEDGMENTS

The authors thank Justine Sullivan and Anjelica Ferguson for helping with the blind listening test described in Section VI.

### REFERENCES

- [1] H. Tak, M. Todisco, X. Wang, J.-w. Jung, J. Yamagishi, and N. Evans, “Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation,” in *ODYSSEY 2022, The Speaker Language Recognition Workshop, June 28th-July 1st, 2022, Beijing, China*, ISCA, Ed., Beijing, 2022.
- [2] T. Zhang, “Deepfake generation and detection, a survey,” *Multimedia Tools and Applications*, vol. 81, no. 5, pp. 6259–6276, 2022.
- [3] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, “Spoofing and countermeasures for speaker verification,” *Speech Communication*, vol. 66, no. C, pp. 130–153, 2015.
- [4] S. Das, S. Seferbekov, A. Datta, M. S. Islam, and M. R. Amin, “Towards solving the deepfake problem: An analysis on improving deepfake detection using dynamic face augmentation,” in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. IEEE, 2021, pp. 3769–3778.
- [5] Y. Zhao, J. Yi, J. Tao, C. Wang, C. Zhang, T. Wang, and Y. Dong, “Emofake: An initial dataset for emotion fake audio detection,” 11 2022.
- [6] J. Yi, C. Wang, J. Tao, Z. Tian, C. Fan, H. Ma, and R. Fu, “Scenefake: An initial dataset and benchmarks for scene fake audio detection,” *CoRR*, vol. abs/2211.06073, 2022.
- [7] J. Yi, Y. Bai, J. Tao, H. Ma, Z. Tian, C. Wang, T. Wang, and R. Fu, “Half-Truth: A Partially Fake Audio Detection Dataset,” in *Proc. Interspeech 2021*, 2021, pp. 1654–1658.
- [8] B. Elizalde, S. Deshmukh, M. Al Ismail, and H. Wang, “Clap learning audio concepts from natural language supervision,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [9] K. Choi, J. Im, L. Heller, B. McFee, K. Imoto, Y. Okamoto, M. Lagrange, and S. Takamichi, “Foley sound synthesis at the dcase 2023 challenge,” 04 2023.
- [10] H. Florez and S. Misra, *Applied Informatics: Third International Conference, ICAI 2020, Ota, Nigeria, October 29–31, 2020, Proceedings*. Springer Nature, 2020, vol. 1277.
- [11] A. K. Singh and P. Singh, “Detection of ai-synthesized speech using cepstral & bispectral statistics,” in *2021 IEEE 4th International Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 2021, pp. 412–417.
- [12] C. Borrelli, P. Bestagini, F. Antonacci, A. Sarti, and S. Tubaro, “Synthetic speech detection through short-term and long-term prediction traces,” *EURASIP Journal on Information Security*, vol. 2021, no. 1, pp. 1–14, 2021.
- [13] I. Altalahin, S. AlZu’bi, A. Alqudah, and A. Mughaid, “Unmasking the truth: A deep learning approach to detecting deepfake audio through mfcc features,” in *2023 International Conference on Information Technology (ICIT)*. IEEE, 2023, pp. 511–518.
- [14] A. Qais, A. Rastogi, A. Saxena, A. Rana, and D. Sinha, “Deepfake audio detection with neural networks using audio features,” in *2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP)*. IEEE, 2022, pp. 1–6.
- [15] D. M. Ballesteros, Y. Rodriguez-Ortega, D. Renza, and G. Arce, “Deep4snet: deep learning for fake speech classification,” *Expert Systems with Applications*, vol. 184, p. 115465, 2021.
- [16] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *3rd International Conference on Learning Representations (ICLR 2015)*. Computational and Biological Learning Society, pp. 1–14.
- [17] B. Elizalde, S. Deshmukh, M. A. Ismail, and H. Wang, “CLAP learning audio concepts from natural language supervision,” in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 1–5.
- [18] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, “Panns: Large-scale pretrained audio neural networks for audio pattern recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.
- [19] Y. Yuan, H. Liu, X. Liu, X. Kang, M. D. Plumbley, and W. Wang, “Latent diffusion model based foley sound generation system for DCASE challenge 2023 task 7,” *CoRR*, vol. abs/2305.15905, 2023.
- [20] H. V. Pham, S. Qian, J. Wang, T. Lutellier, J. Rosenthal, L. Tan, Y. Yu, and N. Nagappan, “Problems and opportunities in training deep learning software systems: An analysis of variance,” *2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 771–783, 2020.