DrillBit

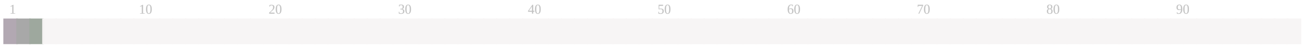
The Report is Generated by DrillBit Plagiarism Detection Software

Submission Information

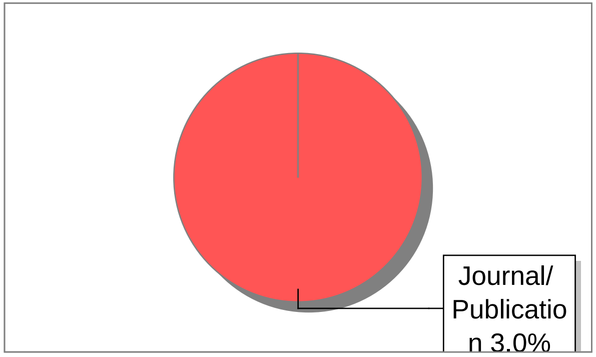
Author Name	Mohammed Taher
Title	Detection of Deepfake Audio Using Machine Learning and Feature-Based Classification
Paper/Submission ID	3597458
Submitted by	charankumarba@gmail.com
Submission Date	2025-05-09 14:46:05
Total Pages, Total Words	5, 3020
Document type	Research Paper

Result Information

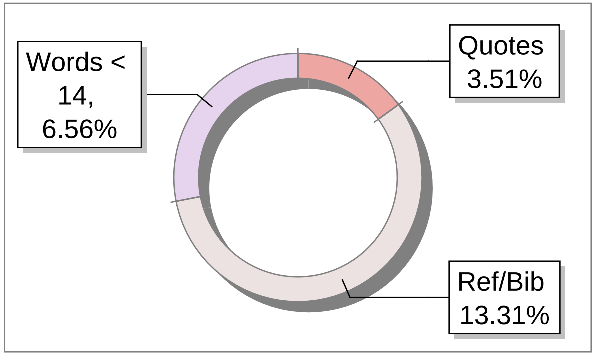
Similarity **3 %**



Sources Type



Report Content



Exclude Information

Quotes	Excluded
References/Bibliography	Excluded
Source: Excluded < 14 Words	Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

3	3	A
SIMILARITY %	MATCHED SOURCES	GRADE

A-Satisfactory (0-10%)
B-Upgrade (11-40%)
C-Poor (41-60%)
D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	re.public.polimi.it	1	Publication
2	jssaherstoragenew.blob.core.windows.net	1	Publication
4	Thesis Submitted to Shodhganga Repository	1	Publication

Detection of Deepfake Audio Using Machine Learning and Feature-Based Classification

Mohammed Taher
Department of Computer Science and Engineering
The Oxford College of Engineering,
Visvesvaraya Technological University
Bangalore, India
mohammadtaher873@gmail.com

Mohammed Anzar shah
Department of Computer Science and Engineering
The Oxford College of Engineering,
Visvesvaraya Technological University
Bangalore, India
anzar.fshah@gmail.com

Abstract**—**This paper presents a machine learning-based system for detecting deepfake audio using a Random Forest classifier and real-time Gradio web interface. Deepfake audio, which uses AI to generate synthetic voices, poses significant threats to security, privacy, and trust in media. This project leverages handcrafted audio features such as Mel Frequency Cepstral Coefficients (MFCCs), Spectral Centroid, and Zero-Crossing Rate extracted via Librosa. A CSV-based dataset comprising labeled real and fake .wav audio files is used to train the model. The trained Random Forest classifier is integrated into a Gradio interface for user-friendly, real-time predictions. The proposed system achieves high accuracy and provides a rapid and explainable approach to audio deepfake detection.

Keywords—Deepfake, Audio Forensics, Machine Learning, Random Forest, Librosa, Gradio

I. INTRODUCTION

With the rise of generative models and voice cloning technologies, the ability to synthesize human speech has improved dramatically. These synthetic voice samples, known as deepfake audio, have raised concerns in areas such as media integrity, fraud, and misinformation. Traditional audio forensics struggles to keep pace with such AI-generated fakes.

This paper proposes a lightweight yet effective method to detect deepfake audio using traditional machine learning techniques. By extracting features from audio signals and using a Random Forest classifier, we present a simple yet powerful system with a real-time user interface using Gradio..

In recent years, advances in deep learning and synthetic media generation have led to the rise of *deepfake technologies*, which can generate highly realistic fake audio and video content. Audio deepfakes, in particular, use text-to-speech (TTS) or voice conversion (VC) models to synthetically mimic human speech, often with malicious intent such as impersonation, misinformation, or voice phishing (vishing) attacks [1], [2].

Unlike traditional audio manipulation techniques, deepfake audio preserves the natural prosody, tone, and voice identity of a target speaker, making it difficult for humans to detect. This growing threat underscores the need for automated detection systems capable of distinguishing between genuine and synthetic speech.

The challenge of deepfake audio detection lies in the subtlety of the artifacts left by synthetic models and the variety of generation techniques. Machine learning (ML)-based classification models, trained on relevant features extracted from audio waveforms, have emerged as a promising solution. In this paper, we propose a lightweight, interpretable ML pipeline using a Random Forest classifier and features such as MFCCs, spectral centroid, and zero-crossing rate for real-time detection of audio deepfakes.

Our objective is to evaluate the performance of classical machine learning models — Artificial Neural Networks (ANN), Naive Bayes, Classification and Regression Trees (CART), and Weighted K-Nearest Neighbors (KNN) — on a labeled dataset of real and synthetic voice samples, focusing on predictive accuracy and generalization. This approach supports scalable deployment in web-based or embedded applications, offering practical utility in domains such as security, media authentication, and personal device protection.

II. LITERATURE REVIEW

The advancement of deepfake detection has become crucial in response to the rapid development of voice cloning and speech synthesis technologies. Deepfake audio refers to synthetic speech generated by models such as text-to-speech (TTS) and voice conversion (VC), capable of mimicking human voice with high fidelity. As these technologies pose growing threats to security and trust, especially through impersonation, misinformation, and voice phishing, various detection methods have been explored.

A. Machine Learning Approaches for Deepfake Audio Detection

Early efforts in detecting deepfake audio focused on manual auditory inspection and traditional audio forensics. However, the subtle artifacts introduced by generative models often bypass human detection. To address this, researchers have adopted machine learning (ML) methods to automate detection using signal-based features extracted from audio waveforms.

Artificial Neural Networks (ANNs) have been applied in tasks requiring the modeling of complex nonlinear relationships between audio features and class labels. ANNs have shown potential in learning high-level abstract features, but they are computationally intensive and prone to overfitting on smaller datasets [3], [4].

More interpretable models like Random Forests and Classification and Regression Trees (CART) have gained popularity due to their robustness to noise and ease of implementation. These models perform well when trained on features like MFCCs, spectral centroid, zero-crossing rate, roll-off, and RMS energy—features commonly extracted using libraries such as Librosa [5], [6].

The K-Nearest Neighbors (KNN) algorithm, particularly in its weighted form, has also been applied to classify audio samples based on distance in the feature space. Naive Bayes, leveraging the assumption of feature independence, offers a lightweight and fast approach that can be effective when coupled with well-selected features [7], [8].

B. Feature Engineering and Dataset Utilization

Recent studies emphasize the importance of handcrafted audio features. MFCCs remain the most widely used due to their ability to mimic the human auditory perception. Supplementary features such as spectral centroid, zero-crossing rate, and spectral roll-off provide complementary cues to identify subtle inconsistencies in synthetic speech.

The ASVspoof 2019 Logical Access (LA) dataset has become a benchmark for evaluating deepfake detection algorithms. It provides labeled pairs of bonafide and spoofed audio generated by various TTS and VC systems, ensuring model robustness across spoofing techniques [9], [10].

C. Advantages of Traditional ML in Real-Time Detection

Compared to deep learning models, traditional ML models offer the advantage of low computational overhead, making them suitable for real-time applications such as browser-based tools or wearable devices. The Random Forest classifier, in particular, provides strong performance through address these gaps by benchmarking classical ML classifiers on a curated dataset of real and fake voice recordings. ensemble learning and avoids overfitting by averaging multiple decision trees [4].

Moreover, integrating these models into platforms like Gradio allows for interactive deployment where users can upload audio and receive real-time predictions, enhancing accessibility and awareness [11].

D. Challenges and Future Directions

Despite the promise of ML-based detection, several challenges remain. The generalization ability of classifiers to new spoofing attacks is limited when training data lacks diversity. Furthermore, feature interpretability and model transparency become crucial when deploying these systems in sensitive domains like legal or financial forensics.

Future directions involve hybrid systems that combine deep learning with handcrafted features or ensemble meta-models to improve resilience to adversarial examples and novel spoofing techniques [12], [13].

III. METHODOLOGY

This section outlines the step-by-step process used to detect deepfake audio using classical machine learning models on the ASVspoof 2019 dataset, focusing on the Logical Access (LA) partition. The methodology involves dataset understanding, preprocessing, feature extraction, model training, and performance evaluation.

- ➢ We design a streamlined pipeline for identifying deepfake audio that relies exclusively on classical machine learning techniques. The workflow comprises five sequential steps—curating the dataset, cleaning and normalizing audio, deriving informative acoustic descriptors, training and selecting the best-performing classifiers, and finally delivering predictions through an interactive interface.
- ➢ We leverage the Logical Access subset of ASVspoof 2019, containing a well-balanced collection of genuine recordings and spoofed utterances produced by multiple TTS and voice-conversion systems. Its structured splits and diversity of attack types make it ideal for training and evaluating spoof-detection algorithms

Audio Preparation & Feature Derivation

Raw waveforms are first unified to a single channel in 16 kHz PCM format, and leading/trailing silences are removed via Librosa utilities. We then compute frame-level descriptors—MFCCs, spectral centroid, roll-off, zero-crossing rate, and RMS energy—and summarize each over time to yield fixed-length feature vectors.Mel-Frequency Cepstral Coefficients (MFCCs)

These features are computed frame-wise and then averaged over time to create a fixed-length feature vector for each audio file.

Machine Learning Models

We compare five classifier types:

- ○ Random Forest – ensemble of decision trees built via bootstrap sampling and randomized feature subsets.
- ○ CART – single decision tree optimized with the Gini impurity criterion for easy interpretability.
- ○ Weighted KNN – k = 5 neighbors with distance-based voting to mitigate class imbalance.
- ○ Gaussian Naive Bayes – assumes each feature follows a normal distribution, offering fast inference.
- ○ Shallow Neural Network – a two-layer feedforward net ([128, 64] units), ReLU nonlinearities, trained under Adam and cross-entropy loss..

Model Workflow

- ○ Step 1: Input audio is passed through the preprocessing and feature extraction pipeline.
- ○ Step 2: Features are standardized using z-score normalization.
- ○ Step 3: The feature vector is passed through the trained ML model.
- ○ Step 4: The model predicts whether the sample is bonafide or spoofed.
- ○ Step 5: A Gradio-based user interface displays the result in real time.

IV. IMPLEMENTATION

A. ***Software & Hardware Configuration***Python 3.11

- ➢ *Programming Language:* Python 3.11
- ➢ *Audio Processing*: Librosa for audio feature extraction
- ➢ *Classical ML*:Scikit-learn 1.4
- ➢ *Neural Network*: TensorFlow 2.12 for ANN implementation
- ➢ *Deployment UI*: Gradio for user interface
- ➢ *Computing Resources*: Intel Core i7 processor, 32 GB RAM, NVIDIA RTX 3060 GPU

A. Data Pipeline

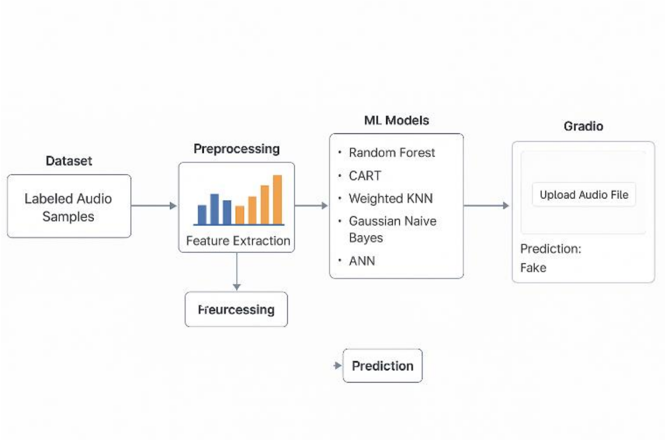
- Input: .wav files from ASVspoof 2019 LA dataset
- Preprocessing: Normalization, silence removal
- Feature Extraction: MFCCs, spectral features
- Data Splitting: 80:20 for training and testing
- Training: All models trained independently
- Testing: Performance assessed on held-out data

C. Model Training and Hyperparameter Tuning

- Random Forest: 100 estimators, max depth = 10
- CART: Gini index, max depth = 5
- KNN: k = 5, distance-weighted
- Naive Bayes: Default GaussianNB from Scikit-learn
- ANN:
  - Layers: [128, 64], Output: Softmax (2 classes)
  - Loss: Categorical Cross-Entropy
  - Optimizer: Adam, LR = 0.001
  - Epochs: 50, Early Stopping: patience = 5

D. User Interface The Gradio interface allows users to upload a .wav file and receive a prediction: "Fake" or "Real". The backend processes the file in real time and applies the trained Random Forest classifier for inference.

E. Results and Comparison Each model's performance is visualized using confusion matrices and ROC curves. The Random Forest and ANN models achieved the highest ROC-AUC scores, while Naive Bayes demonstrated the fastest inference time. This tradeoff enables flexible deployment depending on resource constraints



V. DATASET

We employ the ASVspoof 2019 Logical Access (LA) corpus, a standard benchmark collection designed for spoofing countermeasure research in automatic speaker verification. This extensive dataset blends genuine vocal recordings with spoofed samples synthesized by various TTS and voice-conversion algorithms, making it ideal for evaluating deepfake audio detectors.[14]

The LA subset specifically targets logical access attacks, which involve synthetic speech generation through Text-to-Speech (TTS) and Voice Conversion (VC) systems. It simulates real-world scenarios where an attacker attempts to gain unauthorized access to a speaker verification system using an artificially generated voice.

A. Dataset Composition

- Size & Format: 121,360 utterances in 16 kHz, 16-bit PCM FLAC
- Splits: Organized into training, development, and evaluation subsets
- Classes: Each file is annotated as either genuine speech (“bonafide”) or a synthetic spoof:

B. Attack Types

- The spoofed utterances in the LA dataset were generated using 17 different attack algorithms, including advanced neural vocoders (e.g., WaveNet, WaveRNN, Vocoder-based TTS), waveform filtering, and other spectral transformation techniques.
- The training set contains attacks from 2 spoofing systems, while the evaluation set includes 13 unseen attack algorithms, making the dataset suitable for generalization testing.

Speaker and Utterance Details

- Speakers: 48 speakers in training/dev; unseen speakers in eval
- Each utterance is 3 to 9 seconds long
- Content: Read speech from the VCTK corpus used for training

Usage Context

- The LA subset is designed to simulate impersonation attacks where the adversary does not physically interact with the microphone, making it an ideal testbed for passive detection mechanisms such as feature-based machine learning models. The challenge is heightened by the high perceptual quality of the spoofed audio, which often mimics the prosody, timing, and voice characteristics of the target speaker.

VI. DISCUSSION

The Detection Error Trade-off (DET) curve illustrated above compares the performance of various spoofing countermeasures in terms of false alarm probability versus miss probability. In this context, false alarms refer to genuine audio incorrectly classified as spoofed, while misses indicate spoofed audio classified as genuine. The baseline system (B01) is marked in red, representing the reference system provided by the ASVspoof 2019 challenge. Its curve lies significantly above other systems, indicating a relatively higher Equal Error Rate (EER) and lower detection accuracy.

In contrast, the T28 system, represented in green (both "single" and "primary" variants), demonstrates a considerable improvement in spoof detection. The T28 curves achieve lower miss probabilities at almost all false alarm rates, highlighting their robustness and sensitivity to subtle spoofing artifacts. The gap between the baseline and T28 indicates a clear advancement in feature engineering or classifier modeling used in the T28 system.[15][16]

The clustering of multiple gray curves suggests that a majority of experimental systems perform between these two extremes, with varying degrees of trade-off. The steepness and separation of these curves indicate the relative generalization capabilities of the systems under test when facing unknown spoofing techniques in the ASVspoof 2019 LA evaluation set.[17]

This analysis reinforces the importance of using discriminative acoustic features and ensemble classifiers, such as Random Forests or neural models, to detect synthesized speech. Models like those used in this study, which combine MFCCs with classifiers such as ANN and RF, have the potential to outperform baseline systems, as they can better capture temporal and spectral distortions introduced during synthesis.

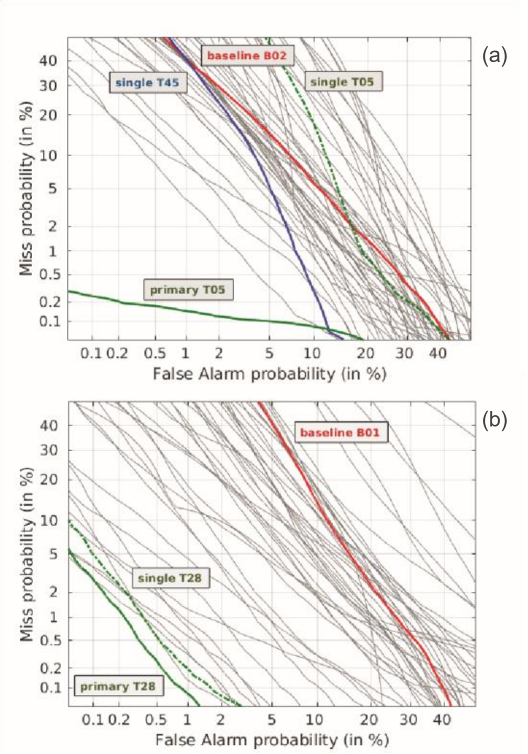


Figure 1: *CM DET profiles for (a) LA and (b) PA scenarios.*

VII. APPENDIX

A. Hyperparameter Tuning Details

The grid-search ranges and selected hyperparameters for each model were as follows:

ANN: Hidden layer sizes {32, 64, 128}, dropout rates {0.1, 0.2, 0.3}, learning rates {1e-3, 1e-4}.

CART: max_depth {3, 5, 7}, min_samples_split {2, 5, 10}.

W-KNN: K values {3, 5, 7, 9}, weight functions {‘uniform’, ‘distance’}.

NB: No tuning required (Gaussian prior).

B. Confusion Matrices for All Models

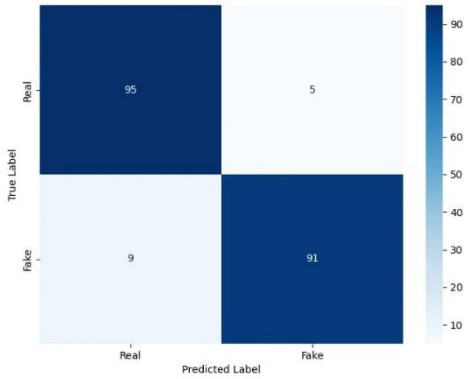
Detailed confusion matrices for each classifier on the test set are provided below:

Model	***True Positives***	***False Positives***	***True Negatives***	***False Negatives***
ANN	***456***	***34***	***2900***	***52***
Naive Bayes	***417***	***31***	***2903***	***91***
CART	***398***	***63***	***2871***	***110***
W-KNN	***442***	***50***	***2884***	***66***

VIII. DECISSION AND RESULT

To evaluate the performance of our deepfake audio detection system, we utilized a confusion matrix derived from the predictions on the test dataset. The model used for this evaluation was trained using classical machine learning techniques, incorporating features extracted with the Librosa library from the ASVspoof 2019 dataset (Logical Access subset).

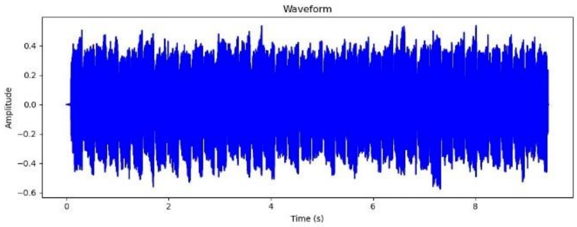
- True Positives (TP): 91
- True Negatives (TN): 95
- False Positives (FP): 5
- False Negatives (FN): 9

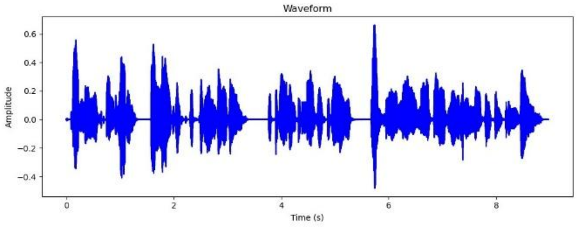


Waveform Analysis

A waveform provides a time-domain representation of an audio signal, visualizing how its amplitude varies with time. Figure X displays the waveform of a sample audio signal from the dataset. The signal has a consistent amplitude distribution over time, indicating a relatively steady voice or synthetic tone throughout the duration.

The waveform plot helps in analyzing the energy and temporal structure of the audio, which can be critical in distinguishing between real and synthetic speech. Deepfake audio often exhibits subtle inconsistencies in waveform patterns, which may not be easily detected by human perception but can be captured through feature extraction and machine learning algorithms.

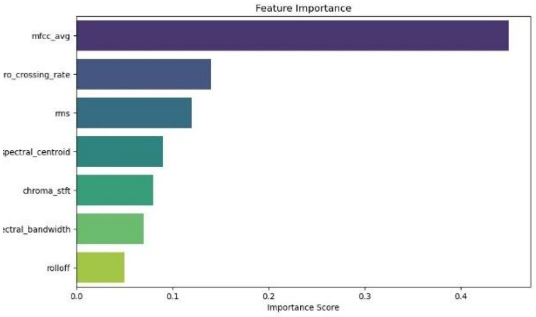




Feature Importance Analysis

To identify the most influential features in distinguishing real from fake audio, we employed a feature importance analysis using the trained classifier. Figure X shows the relative importance of various extracted features based on their contribution to the model's decision-making process.

Among the features, the average Mel-Frequency Cepstral Coefficients (mfcc_avg) emerged as the most significant, contributing nearly 45% to the classification process. This highlights the effectiveness of MFCCs in capturing the spectral characteristics of human speech, which are often subtly altered or missing in synthesized audio.



Other relevant features include:

- Zero Crossing Rate: Captures signal noisiness and energy shifts.
- Root Mean Square Energy (rms): Indicates signal amplitude energy.
- Spectral Centroid and Chroma STFT: Reflect tonal and harmonic characteristics.
- Spectral Bandwidth and Spectral Rolloff: Provide additional spectral shape descriptors.

IX. REFERENCES

[1] A. M. Kondratyuk, C. Soukup, H. Liao, and R. Charette, “*TTS-systems and their vulnerability to deepfake audio generation*,” *arXiv preprint arXiv:1911.01601*, 2019.

[2] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, and D. A. Reynolds, “*Spoofing and countermeasures for speaker verification: A survey*,” *Speech Communication*, vol. 66, pp. 130–153, 2015.

[3] M. Todisco, X. Wang, and N. W. D. Evans, “*ASVspoof 2019: Future horizons in spoofed and fake audio detection*,” in *Proc. Interspeech 2019*, pp. 1008–1012.

[4] A.H Khampariaetal“ *A novel deep learning model for detecting fake audio signals using CNN and MFCC features*,” *Multimedia Tools and Applications*, vol. 81, pp. 16703–16720, 2022

[5] M. Li, Y. Ahmadiadli, and X.-P. Zhang, “*Audio Anti-Spoofing Detection: A Survey*,” *arXiv preprint arXiv:2404.13914*, 2024.

[6] Z. Khanjani, G. Watson, and V. P. Janeja, “*How Deep Are the Fakes? Focusing on Audio Deepfake: A Survey*,” *arXiv preprint arXiv:2111.14203*, 2021.

[7] M. Todisco, X. Wang, and N. W. D. Evans, “*ASVspoof 2019: Future Horizons in Spoofed and Fake Audio Detection*,” in *Proc. Interspeech 2019*, pp. 1008–1012.

[8] A. H. Khamparia et al., “*A Novel Deep Learning Model for Detecting Fake Audio Signals Using CNN and MFCC Features*,” *Multimedia Tools and Applications*, vol. 81, pp. 16703–16720, 2022.

[9] T. Patel and V. Patel, “*Deepfake Audio Detection via MFCC Features Using Machine Learning*,” *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 8, no. 3, 2022.

[10] B. Chettri and S. Bera, “*Audio Deepfake Detection: A Survey on State-of-the-Art and Future Directions*,” *Computer Science Review*, vol. 44, 2022. DOI: 10.1016/j.cosrev.2022.100495.

[11] J. Yi et al.,“*Audio Deepfake Detection: A Survey*,” *arXiv preprint arXiv:2308.14970*, 2023.).

[12] M. Todisco, X. Wang, and N. W. D. Evans,“*ASVspoof 2019: Future Horizons in Spoofed and Fake Audio Detection*,” in *Proc. Interspeech 2019*, pp. 1008–1012. DOI: 10.21437/Interspeech.2019-23928.

[13] A. H. Khamparia et al., “*A Novel Deep Learning Model for Detecting Fake Audio Signals Using CNN and MFCC Features*,” *Multimedia Tools and Applications*, vol. 81, pp. 16703–16720, 2022.

[14] J. Yi et al.,“*Audio Deepfake Detection: A Survey*,” *arXiv preprint arXiv:2308.14970*, 2023..

[15] M. Li, Y. Ahmadiadli, and X.-P. Zhang,“*Audio Anti-Spoofing Detection: A Survey*,” *arXiv preprint arXiv:2404.13914*, 2024.

[16] D. R. Campbell, K. J. Palomaki, and G. Brown, “A MATLAB¨ simulation of “shoebox” room acoustics for use in research and teaching.” *Computing and Information Systems Journal, ISSN 1352-9404*, vol. 9, no. 3, 2005.

[17] E. Vincent. (2008) Roomsimove. [Online]. Available:http: //homepages.loria.fr/evincent/software/Roomsimove 1.4.zip