Audio Deepfake Detection using Temporal – Spectral Hybrid Features

Chakrapani Anisetti[1], Dr. P. Sanyasi Naidu[2]
[1]Student, Andhra University College of Engineering (A), Visakhapatnam, Andhra Pradesh
[2]Professor, Andhra University College of Engineering (A), Visakhapatnam, Andhra Pradesh

Abstract— **Deepfake audio detection is becoming more important because of the growing misuse of synthetic speech technology in harmful ways. This study suggests a new set of features that combine Linear Frequency Cepstral Coefficients (LFCC), their first and second derivatives, and high-energy features to effectively detect deepfake audio. Using the Fake-or-Real (FoR) dataset, we validated our feature-integration strategy to guarantee consistent spoof-detection performance across various acoustic settings. With accuracies ranging from 70% to 99% on the four FoR subsets, the method demonstrated strong cross-condition generalization. This shows its strength and effectiveness for real-world deepfake audio detection tasks.**

Index Terms— **Audio deepfake detection, Handcrafted features, Temporal-spectral hybrid features (TSHF), Fake-or-Real dataset.**

I. INTRODUCTION

Advancements in generative artificial intelligence have made it possible to create highly realistic human speech, known as deepfake audio. While having many benefits across multiple fields they also come with significant risks, they can be exploited by people to make malicious actions like impersonation, fraudulent transactions which calls need for digital security. As deepfake generation tools become more accessible, we need reliable detection methods to protect voice-based systems. Early research on deepfake audio detection mostly employed conventional classifiers like support vector machines (SVMs) in conjunction with spectral descriptors like Mel-frequency cepstral coefficients (MFCCs). Recent methods leverage deep learning models — including convolutional, recurrent, and temporal-convolutional networks — that automatically extract hierarchical spectral–temporal features from either raw audio or time–frequency representations (e.g., spectrograms). These models have demonstrated high accuracy on datasets such as ASVspoof and FoR when recordings are well controlled. Even though existing detection frameworks are effective, they have several limitations— Computational Overhead, Limited Interpretability, Environmental Vulnerability. These challenges highlight the need for a robust, understandable, and resource-efficient solution that can maintain performance in various acoustic settings. This research introduces a set of features called Temporal-Spectral Hybrid Features (TSHF), specifically designed to improve the detection of synthetic speech. TSHF is a novel integration of LFCC, First and second-order temporal derivatives (Δ and ΔΔ), Frame-level statistical descriptors (mean, standard deviation, minimum, maximum), High-frequency energy metrics targeting the 3-8 kHz range. The objective is to develop a lightweight, interpretable detection framework that maintains high accuracy across the FoR subsets (FoR-Original, FoR-Norm, FoR-2seconds, FoR-Rerec). Using conventional machine-learning models, the proposed system shows strong generalization and is suitable for deployment in resource-constrained and forensic settings.

II. RELATED WORK

The rise of audio deepfakes in the last years inspired researchers to develop deepfake detection algorithms to ensure security, avoid fraud, and maintain integrity in speaker verification systems. Since initially, approaches have evolved from simple handcrafted feature-based models to deep learning models, hybrid systems, and novel time-frequency representations. The first countermeasures consist of using handcrafted acoustic features as input to traditional classifiers. Such systems extract cepstral features such as MFCC, LFCC, or GTCC as input to a classifier such as Support Vector Machines (SVM), Random Forests

(RF), or k-Nearest Neighbors (KNN). Deepfake Audio A. Hamza et al [1] achieved good results on the Fake-or-Real (FoR) dataset by concatenating MFCCs with classifiers such as SVM and Gradient Boosting. A N. Chakravarty and M. Dua[2] showed that Mel-spectrogram-based features can be compressed using Linear Discriminant Analysis (LDA) before classification. This system achieved high accuracy with low computational cost. P. Chiddarwar [3] also falls into this category, focusing on low latency by connecting MFCC features to a lightweight convolutional network. The next stage consists of end-to-end deep learning models where spectrograms or even raw waveforms are fed into neural networks. L. P. Valente, M. M. S. de Souza, and A. M. D. Rocha [4] applied VGG-style CNNs on Mel-spectrograms. L. Pham, P. Lam, T. Nguyen, H. Nguyen, and A. Schindler [5] applied different types of spectrograms as input to CNN, RNN, and Transformer backbones. This system achieved very low Equal Error Rates (EERs) on the ASVspoof 2019 dataset. Efficient V. Sunkari and A. Srinagesh [6] applied CNNs on both spatial (MFCC) and temporal (LSTM) levels to achieve a good accuracy–computing cost trade-off. Hybrid and multi-view feature fusion systems try to combine the best of both handcrafted and learned representations. Y. Yang et al [7] applied LFCC, MFCC, and self-supervised embeddings (wav2vec2, HuBERT) to improve generalization in unseen environments. A. J. Lakshmi, V. Sindhuja, B. Meghana, and G. S. Gupta [8] applied handcrafted MFCC features as input to deep architectures (Spectrogram-based Feature). Another line of work applied different time-frequency representations. U. Avaiya, N. Badlani, R. Baadkar, and K. Talele [9] applied a Deep Scattering Network that uses mathematically stable wavelet filters to create features that are interpretable and robust to common distortions. I. Altalahin, S. AlZu'bi, A. Alqudah, and A. Mughaid [10] applied deep neural networks on cepstral data (MFCC), improving robustness over traditional machine learning standards. The proposed Temporal-Spectral Hybrid Features (TSHF) are an extension of the clarity and efficiency of previous hand-crafted methods[1], [2], [3] while lifting their weakness in terms of feature diversity. TSHF obtain their spectral and temporal information from synthetic speech by fusing LFCCs and first- and second-order temporal derivatives (Δ and ΔΔ), frame-level statistical pooling, and selective high-frequency energy metrics (3–8 kHz). While being less complex than any multi-model or deep feature fusion system[5], [7], [8] TSHF obtain high accuracy on all FoR subsets using traditional machine learning codes, rendering them a candidate for real-time, resource-constrained, and forensic applications.

III. METHODOLOGY

The proposed system is a feature-based approach for synthetic speech detection. It is interpretable, lightweight, and works well in various acoustic environments. While deep learning methods require large amounts of training data and powerful computation, this method uses specific audio features and traditional machine learning algorithms, and requires little overhead. We propose a new type of features, Temporal-Spectral Hybrid Features (TSHF), that capture both spectral qualities and time variations of the speech signal. The spectral part is derived from Linear Frequency Cepstral Coefficients (LFCCs), which create a linear frequency scale that retains more high-frequency information - information that often corresponds to synthetic audio. To understand how the speech changes over time, we compute first and second derivatives (Δ and ΔΔ) of LFCCs, which show unnatural transitions and modulations of deepfake audio. In addition to analyzing each frame individually, we also compute overall statistical descriptors (mean, standard deviation, minimum, and maximum) of each LFCC dimension over the entire audio sample, which describes how the signal behaves and makes the system more robust to noise and variation. Finally, we gather high-frequency energy metrics from the 3 – 8 kHz range of the signal, which is often missing in synthetic speech (due to vocoder limitations), and concatenate everything together to form our final feature vector.

IV. SYSTEM ARCHITECTURE

We implement the detection system as a modular pipeline with four primary stages: data preprocessing, feature extraction, classifier training, and performance evaluation. Each module is engineered for reproducibility and scalability, and tuned to address the practical challenges of detecting synthetic speech across diverse recording conditions. The structure uses a mix of open-source libraries and parallel processing

techniques to improve performance with large datasets.

DATA SET AND PREPROCESSING

Download the data and split it into train, test and validation sets. Set labels (binary: fake/real). Save preprocessed audio and extracted features in output directories. Most of the hyperparameters: target_sampling_rate (16kHZ) - minimum duration (1 sec/16k samples) to make input clips of uniform size, cleaning (This pipeline will automatically remove duplicate files using MD5 hashing, zero-bit/silent files and very short clips), padding short clips (Zero-padding short clips to the 1 second), normalization (Amplitude normalization using StandardScaler to standardize signal strength), All audio will be stored as WAV files (to have uniform inputs), MP3 files (in case of FoR-Original dataset) will be converted to WAV using pydub, Finally, preprocessing is done in batch mode (This enables us to use joblib for parallelization of the process and tqdm to keep track of the progress).

DATASET BALANCING AND INTEGRITY CHECKING.

In order to prevent class imbalance, we randomly remove some samples from majority class. Integrity checks: make sure that the files are in the right WAV format, labels are valid (fake = 1, real = 0), and there is no corrupt files in the dataset. At the end, save the balanced dataset on Google Drive. Also delete the extra raw files to save the space.

FEATURE EXTRACTION PIPELINE

The feature extraction module takes raw audio signals and transforms them into a structured array of numbers that represent certain characteristics of the speech, with an emphasis on subtle artifacts introduced by the generation of synthetic audio, which are useful for deepfake detection. To compute Linear Frequency Cepstral Coefficients (LFCCs), we construct a filterbank create_filterbank(sr, num_filters=40, n_fft=512). The filterbank is linearly spaced up to 1 kHz to preserve fine details, and then logarithmically spaced above 1 kHz to the Nyquist frequency to increase coverage of higher-frequency regions where synthesis artifacts are likely to occur. We then compute FFT bin center frequencies with librosa.fft_frequencies. Filter edges line up to FFT bins using np.searchsorted. Each filter is applied as a normalized rectangular (boxcar) window, so that the energy is approximately preserved across bands. We store the complete filterbank as a reusable 2D NumPy array (num_filters × freq_bins), which will be used to extract features from audio. Each audio file is then processed with extract_features_for_file(args). We load input waveforms in mono at 16kHz with librosa.load. We then perform a Short-Time Fourier Transform (STFT) to obtain a time-frequency representation of the waveform, from which we derive the magnitude spectrum. Applying the filterbank yields band-limited spectral coefficients, which we convert into LFCCs. This results in small feature vectors that summarize information about both the fine structure and high-frequency energy patterns of speech, such that important cues for deepfake detection are preserved. To compute LFCCs, the following steps are followed:

- The STFT matrix is downsampled to match the resolution of the filterbank (usually 257 bins for n_fft=512).
- The filterbank is applied through matrix multiplication, projecting the spectrum onto predefined frequency bands.
- Logarithmic compression is used to reduce dynamic range and improve robustness against variations in amplitude.
- A Discrete Cosine Transform (DCT-II) is applied to reduce correlation between the filter energies and retain the first 20 coefficients per frame.
- To capture temporal variations:
- First-order (delta) and second-order (delta-delta) derivatives are calculated using librosa.feature.delta.
- For each of the three coefficient sets (static LFCCs, delta, delta-delta), statistical measures such as mean, standard deviation, minimum, and maximum are computed across time frames.

These statistical values are combined into a single vector of shape (20 coefficients × 4 stats × 3 types) = 240 dimensions. In addition to the cepstral features, the system calculates high-frequency energy metrics to detect unnatural patterns in the spectral content. Frequency bins within the 3–8 kHz range are isolated

using a binary mask. The magnitude spectrum in this region is analyzed to compute:

- Mean and standard deviation of high-frequency energy.
- Ratio of high-frequency energy to total energy.
- Standard deviation of per-frame energy ratios, indicating temporal consistency.

These metrics are combined into a 4-dimensional vector. The LFCC-based statistics and high-frequency energy features are merged into a single feature vector of length 244. This vector serves as the input to the classification module. If an audio file cannot be loaded or processed, the system returns None for the feature vector but keeps the associated label for record-keeping. Table -1 describes the Audio Feature vector composition

Table 1: Audio Feature vector composition

Feature Type	Base Count	Final Dimensions	Description
LFCC Statistics	20	80	Mean, Standard Deviation, Min and Max for each LFCC sample
Δ LFCC	20	80	Temporal derivatives of LFCC with same statistical summaries
ΔΔ LFCC	20	80	Acceleration coefficients with statistical summaries
High-Frequency Energy Statistics	—	4	Handcrafted Energy features from the 3-8 kHz band

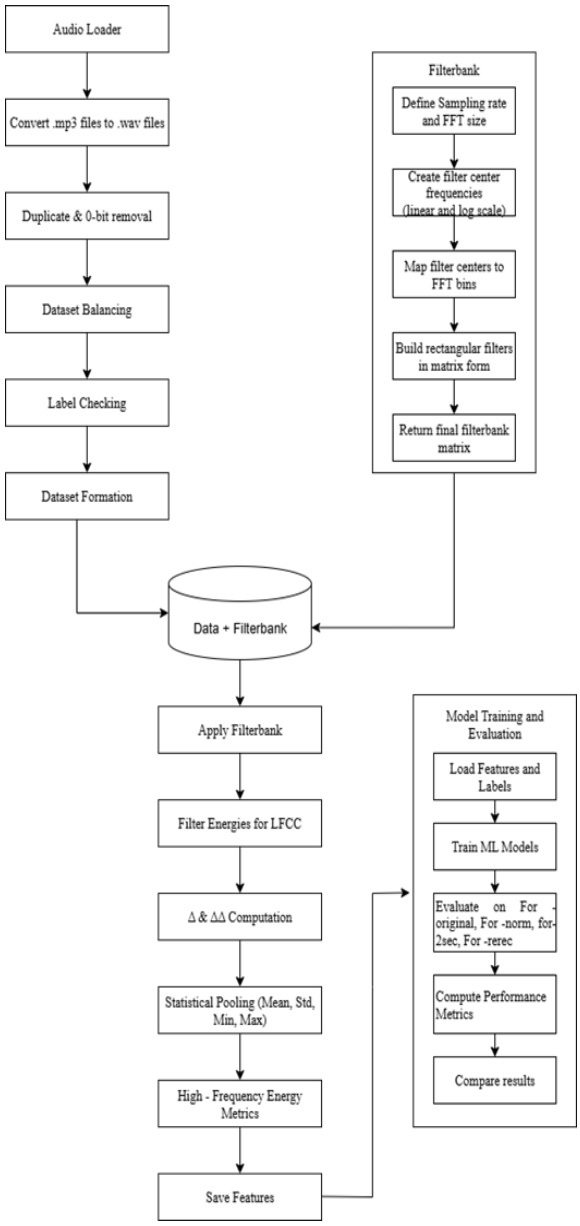

Figure 1: *System Architecture*

V. RESULTS

The proposed system was evaluated on the FoR dataset, which contains four subsets: FoR-Original, FoR-Norm, FoR-2seconds, and FoR-Rerec. These subsets represent clean recordings, normalized versions, temporally truncated audio, and re-recorded samples through speakers and microphones, respectively—offering a wide spectrum of challenges for synthetic speech detection. Multiple classifiers—SVM, KNN, Logistic Regression, and Random Forest—were trained using the handcrafted TSHF. The Random Forest classifier consistently

outperformed alternative models, achieving 99.80% accuracy on FoR-Original and 84.93% accuracy on FoR-Rerec, demonstrating both high performance in controlled conditions and resilience to re-recording artifacts. The high-frequency energy components (3–8 kHz), which are often suppressed in synthetic speech generated by vocoders, proved especially discriminative. Temporal derivatives (Δ and ΔΔ) further enhanced model performance by capturing unnatural transitions and modulation patterns. The inclusion of statistical descriptors added sensitivity to distributional anomalies. Precision, recall, and F1-score were consistently high across all FoR subsets, supporting the system's generalizability. The framework achieves a favorable trade-off between computational efficiency and detection accuracy, making it a practical alternative where deep-learning detectors are not viable.

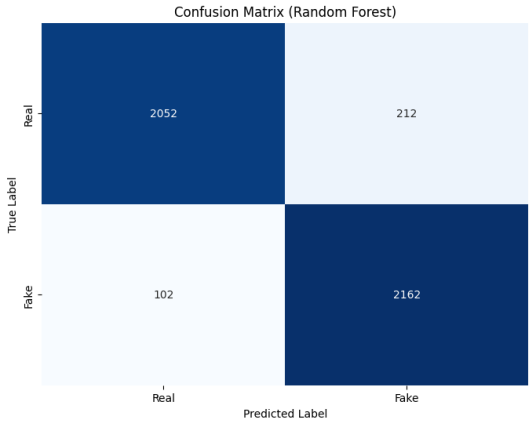

Figure 2: *Confusion matrix for the Random Forest classifier on the FoR-Original subset*

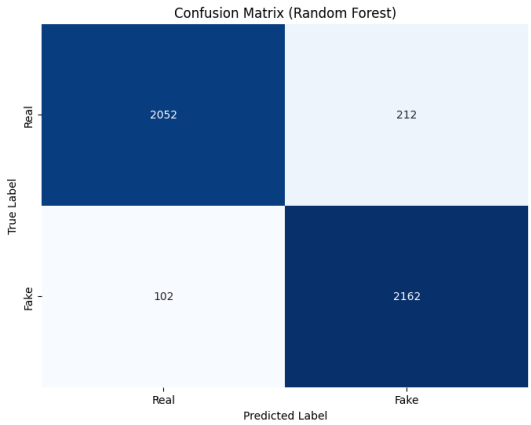

Figure 3 *Confusion matrix for the Random Forest classifier on the FoR-Norm subset*

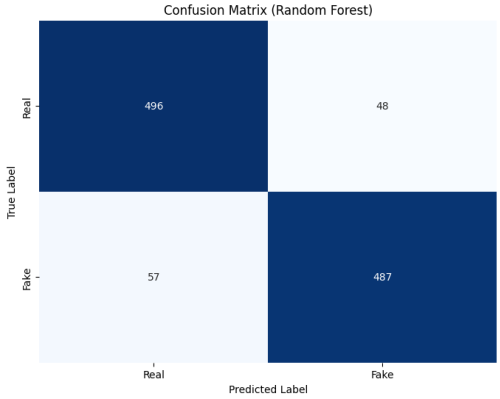

Figure 4 *Confusion matrix for the Random Forest classifier on the FoR-2Sec subset*

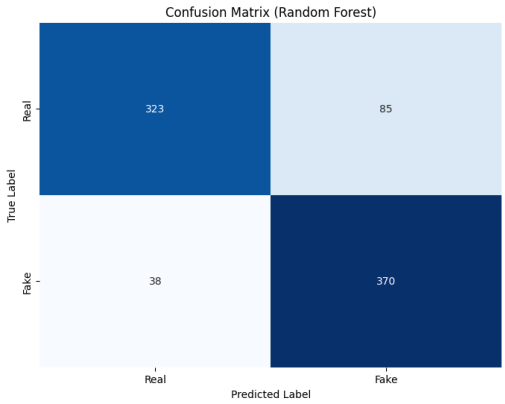

Figure 5 *Confusion matrix for the Random Forest classifier on the FoR-Rerec subset*

Table 2: *Accuracies for different models used.*

Models used	For - Original	For - Norm	For - 2Sec	For - Rerec
LinearSVC	97.11	88.78	73.44	77.94
RBF - SVM	99.14	91.10	83.64	83.33
Polynomial – SVM	98.98	93.31	87.04	82.97
Sigmoid – SVM	78.47	67.40	70.96	73.53
Logistic Regression	97.04	88.80	73.35	77.21
Random Forest	99.80	93.07	90.35	84.93
Naïve Bayes	97.55	75.95	66.73	76.96
MLP	98.83	88.78	81.89	77.94
KNN	96.25	88.03	79.60	73.53
XGBoost	99.78	89.44	81.89	80.64
AdaBoost	99.49	80.06	75.37	69.49
LDA	97.44	88.05	74.91	76.10
QDA	90.08	83.48	81.16	84.19

VI. CONCLUSION

This study introduces a Temporal–Spectral Hybrid Feature (TSHF)–based detection system that captures fine-grained temporal and spectral cues characteristic of synthetic speech. By avoiding heavy neural architectures, the approach emphasizes interpretability and low computational cost, making it suitable for real-time and forensic use. The system showed excellent performance on the Fake-or-Real dataset, with particularly strong results on clean recordings and respectable outcomes under noisy, re-recorded conditions. The handcrafted features, including LFCCs, high-frequency metrics, and temporal derivatives, captured both spectral and transitional irregularities common to deepfake audio.

VII. FUTURE SCOPE

Possible extensions include integrating Temporal–Spectral Hybrid Features (TSHF) with compact neural models, adapting the system for multilingual and multi-accent datasets, and developing defenses against adversarial manipulations. Pursuing these avenues will clarify the scalability and resilience of feature-driven detectors as synthetic audio becomes more sophisticated.

REFERENCE

[1] A. Hamza *et al.*, "Deepfake Audio Detection via MFCC Features Using Machine Learning," *IEEE Access*, vol. 10, pp. 134018–134028, 2022, doi: 10.1109/ACCESS.2022.3231480.

[2] N. Chakravarty and M. Dua, "A lightweight feature extraction technique for deepfake audio detection," *Multimed. Tools Appl.*, vol. 83, no. 26, pp. 67443–67467, Aug. 2024, doi: 10.1007/s11042-024-18217-9.

[3] P. Chiddarwar, "Real-Time Detection of AI-Generated Deepfake Audio: A Novel Approach," in *2024 IEEE 4th International Conference on ICT in Business Industry & Government (ICTBIG)*, Dec. 2024, pp. 1–5. doi: 10.1109/ICTBIG64922.2024.10911062.

[4] L. P. Valente, M. M. S. de Souza, and A. M. D. Rocha, "Speech Audio Deepfake Detection via Convolutional Neural Networks," in *2024 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, May 2024, pp. 1–6. doi:10.1109/ EAIS58494.2024.10569111.

[5] L. Pham, P. Lam, T. Nguyen, H. Nguyen, and A. Schindler, "Deepfake Audio Detection Using Spectrogram-based Feature and Ensemble of Deep Learning Models," July 01, 2024, *arXiv*: arXiv:2407.01777. doi: 10.48550/arXiv.2407.01777.

[6] V. Sunkari and A. Srinagesh, "Efficient Deepfake Audio Detection Using Spectro-Temporal Analysis and Deep Learning," *J. Electr. Syst.*, vol. 20, no. 5s, pp. 10–18, Apr. 2024, doi: 10.52783/jes.1829.

[7] Y. Yang *et al.*, "A robust audio deepfake detection system via multi-view feature," Mar. 04, 2024, *arXiv*: arXiv:2403.01960. doi: 10.48550/arXiv.2403.01960.

[8] A. J. Lakshmi, V. Sindhuja, B. Meghana, and G. S. Gupta, "Detection of Deepfake Audio Using Deep Learning," in *2024 9th International Conference on Communication and Electronics Systems (ICCES)*, Dec. 2024, pp. 1878–1882. doi: 10.1109/ICCES63552.2024.10859752.

[9] U. Avaiya, N. Badlani, R. Baadkar, and K. Talele, "Audio Deepfake Detection using Deep Scattering Network," in *2024 9th International Conference on Communication and Electronics Systems (ICCES)*, Dec. 2024, pp. 2063–2069. doi: 10.1109/ICCES63552.2024.10859322.

[10] 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)*, Aug. 2023, pp. 511–518. doi: 10.1109/ICIT58056.2023.10226172.