INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 09, September 2025, pp : 671-675

www.ijprems.com
editor@ijprems.com

e-ISSN : 2583-1062
Impact Factor : 7.001

AUDIO DEEPFAKE DETECTION

Prof. G.L. Girhe[1], Toshika Ninawe[2], Chaitali Ramtekkar[3]

[1]Professor, Computer Engineering Department, SRPCE, Nagpur, Maharashtra, India.

[2,3]UG Student, Computer Engineering Department, SRPCE, Nagpur Maharashtra, India.

ABSTRACT

Audio deepfakes, generated through advanced speech synthesis and voice conversion techniques, have emerged as a growing threat to information authenticity and security. These artificially created audio clips are often indistinguishable from genuine speech, making them a potential tool for misinformation, fraud, and impersonation. In this work, an efficient audio deepfake detection system is proposed, employing acoustic feature extraction and machine learning classification. Features such as Mel-Frequency Cepstral Coefficients (MFCCs), Chroma, Mel Spectrogram, Zero-Crossing Rate, Spectral Centroid, and Spectral Flatness are extracted from both genuine and manipulated audio samples. The extracted features are used to train a supervised classifier capable of distinguishing between authentic and synthetic voices with high accuracy. It is built with Flask as the backend framework, the system allows users to register, log in, and upload an audio file for authenticity analysis. The proposed system demonstrates its effectiveness in identifying deepfake audio and contributes toward developing secure and reliable digital communication systems.

In today's digital world, the rise of artificial intelligence (AI) has made it easier to create fake audio recordings that sound like real people. This technology, known as "deepfake audio," is becoming more common and can be used for harmful purposes such as spreading misinformation, impersonating others, or committing fraud. Because of this, there is a growing need for tools that can detect whether an audio clip is real or fake.

The extracted features are compared to a pre-existing dataset that contains both real and fake audio samples. Using a nearest approach based on distance calculations, the system identifies the most similar audio file in the dataset. It then provides the user with the result, indicating whether the audio is likely to be real or fake, along with a percentage score showing the confidence level of the prediction. This system is designed to be lightweight, fast, and accessible, making deepfake detection easier for everyday users without needing deep technical knowledge.

Keywords: Deepfake Audio, Real And Fake Audio Samples, Machine Learning, Flask, As Mel-Frequency Cepstral Coefficients (MFCCS), Chroma, Mel Spectrogram, Zero-Crossing Rate, Spectral Centroid, And Spectral Flatness Are Extracted.

1. INTRODUCTION

In recent years, deepfake technologies have emerged as one of the most talked-about innovations — and threats — in the field of artificial intelligence. While visual deepfakes (fake videos or images) have received considerable attention, audio deepfakes are just as dangerous. Using AI models, it is now possible to replicate a person’s voice with stunning accuracy, making it sound like they said something they never did. This raises serious concerns about misinformation, online fraud, and digital security.

Imagine receiving a voice message from your boss, instructing you to transfer money, or hearing a political leader saying something controversial — only to later find out it was never real. These are just a few examples of how deepfake audio can be exploited. Unfortunately, with free tools and open-source models available online, anyone can now generate fake audio with minimal effort. The problem isn’t just the creation of deepfakes — it’s the lack of tools available to detect them quickly and reliably.

To address this problem, our project introduces a deepfake audio detection system as a web application. It is built using the Flask framework, which allows us to create a lightweight and fast backend. The system uses the Librosa library, which is widely used in audio analysis, to extract a wide range of audio features. These include:

- **MFCCs** – represent the timbre of the audio,
- **Mel Spectrogram** – shows how energy is distributed across frequency bands,
- **Chroma Features** – capture harmonic and tonal content,
- **Zero-Crossing Rate** – indicates noisiness in the signal,
- **Spectral Centroid and Flatness** – describe brightness and tonality.

The audio is then compared against a labeled dataset using simple but effective mathematical distance calculations to find the closest match. If the most similar file in the dataset is labeled "deepfake," the uploaded audio is likely fake. The system also calculates a confidence score, showing how close the uploaded audio is to its nearest match.

On the user side, the application includes a login and registration system, styled templates with background images, and a results page showing whether the audio is real or fake. This makes the tool practical for real-world usage, such as by journalists, educators, content creators, or security analysts.

By making deepfake detection more accessible, this project contributes to digital literacy and online safety. In the future, it can be extended with machine learning models for even more accurate predictions or be expanded into mobile or enterprise-level solutions.

2. RELEATED WORK

1. Rehearsal with Auxiliary-Informed Sampling for Audio Deepfake Detection (RAIS) – 2025

This paper introduces a novel continual learning strategy specifically for audio deepfake detection. The RAIS (Rehearsal with Auxiliary-Informed Sampling) method addresses the challenge of adapting models to new types of deepfake attacks without forgetting what was learned previously. It employs an auxiliary label generation network that guides selection of a diverse set of samples to preserve in the model's memory buffer. In experiments, RAIS achieves an impressive average Equal Error Rate (EER) of just 1.953% across five incremental learning scenarios, outperforming other state-of-the-art rehearsal-based approaches.

2. Lightweight Joint Audio-Visual Deepfake Detection via Single-Stream Multi-Modal Learning Framework (2025) Proposes a highly compact, efficient single-stream network that fuses audio and visual features using collaborative blocks, achieving strong performance with just 0.48M parameters. Designed for resource-constrained environments while handling both uni-modal and multi-modal deepfakes.

3. Two Views, One Truth: Spectral and Self-Supervised Features Fusion for Robust Speech Deepfake Detection (2025) Introduces a fusion strategy combining self-supervised learning (SSL) representations with handcrafted spectral features (like MFCC, LFCC, CQCC). Fusion methods such as cross-attention yield a 38% reduction in equal error rate (EER) compared to SSL-only baselines, significantly improving generalization.

4. Mel Spectrogram-Based CNN Framework for Explainable Audio Deepfake Detection (2025)

Evaluates various CNN backbones (e.g., VGG, ResNet, EfficientNet) on Mel spectrogram inputs and employs Grad-CAM to provide interpretable detection heatmaps. Robust against noisy conditions and emphasizes interpretability alongside accuracy.

5. Evaluation framework for deepfake speech detection: a comparative study of state-of-the-art deepfake speech detector (2025) Offers a standardized framework to assess 40 deepfake audio detection systems under unified protocols. Emphasizes real-world generalization, benchmarking, and reproducibility—essential for advancing field transparency and best practices.

6. Region-Based Optimization in Continual Learning for Audio Deepfake Detection (2024)

Tackles the evolving nature of deepfake audio by applying a continual learning framework named RegO. It adaptively fine-tunes model regions based on importance, reduces forgetting, and improves adaptability—yielding a 21.3% EER improvement over previous continual learning approaches.

7. "LA-DF: Lightweight Audio Deepfake Detection Framework for Real-Time Applications" (2024)

Proposed a lightweight detection system using MFCC and spectral flatness features for real-time fake audio identification. Implemented on a Flask-based web interface to allow instant verification of uploaded speech samples.

8. SLIM: Style-Linguistics Mismatch Model for Generalized Audio Deepfake Detection (2024)

Utilizes style-linguistic inconsistencies in fake audio via self-supervised learning on real samples, combined with pretrained acoustic features (e.g., Wav2Vec). SLIM excels at out-of-domain generalization and adds interpretability by quantifying stylistic mismatches.

9. Pushing the Boundaries of Deepfake Audio Detection with a Hybrid MFCC and Spectral Contrast Approach (2024) Introduces hybrid feature extraction combining MFCCs and spectral contrast, alongside SVM classification, to improve detection performance—especially under noisy or real-world scenarios.

10. "ASV spoof 2021: Advancing Spoofed and Deepfake Audio Detection" – (2023)

Introduced a large-scale dataset and evaluation framework for automatic speaker verification spoofing detection. Used CQCC, LFCC, and spectrogram-based CNN models to improve robustness against unseen attack types.

3. METHODOLOGY

1) User Interaction (Frontend)

The system provides a simple and user-friendly web interface developed using Flask (Python-based web framework) and HTML templates for rendering dynamic pages. Users can navigate through several key pages:

- index.html: Serves as the homepage introducing the project.
- login.html: Allows users to securely log into their accounts.
- register.html: Lets new users create an account by entering a username, email, and password.
- model.html: Core functional page where users upload audio files for deepfake detection. It also displays the analysis result.
- about.html and contact.html: Provide background information and contact details for further communication or support.

2) User Authentication

To ensure secure access to the model, a user authentication system is integrated:

- Users must register first via register.html, where their data is validated and stored in a SQLite table named REGISTER.
- At login (login.html), credentials are checked against the stored data.
- Upon successful login, a session is created using Flask’s session management. This ensures that only authenticated users can access sensitive functionality like the audio model.
- Error messages (e.g., incorrect password, existing email) are shown directly in the interface.

3) Audio Upload & Preprocessing

After login, users are directed to model.html, where they can upload an audio file (typically in .wav format).

- The uploaded file is processed in-memory using Python’s file stream handling.
- The librosa.load() function reads the audio data directly, converting it into a time-series waveform and determining its sample rate.
- This step ensures the uploaded file is ready for feature extraction, skipping unnecessary file saving or disk operations — making it fast and efficient.

4) Feature Extraction (Using Librosa)

MFCC (Mel-Frequency Cepstral Coefficients): Represents short-term power spectrum using perceptual Mel scale. Captures timbre and speech texture; helps identify unnatural or robotic voice characteristics.

Mel Spectrogram: Power spectrogram mapped onto Mel frequency scale. Reveals energy distribution over frequency and time; detects irregularities in frequency patterns.

Chroma Features: Describes the 12 pitch classes present in the signal. Detects inconsistencies in pitch and harmonic structure, which may occur in synthetic voices.

Zero-Crossing Rate (ZCR): Measures how frequently the waveform crosses the zero amplitude line. High ZCR values can indicate noisy or unstable speech typical of fake or computer-generated audio.

Spectral Centroid: Indicates the "center of mass" of the frequency spectrum. Reflects the brightness of sound; helps identify overly smooth or flat-sounding fake audio.

5) Dataset Comparison (Similarity Matching)

A pre-labeled dataset (stored in dataset.csv) is loaded during application initialization. This dataset contains:

- Audio feature vectors
- Corresponding labels: "real" or "deepfake"

6) Result Display

Finally, the system displays the results to the user directly on the model.html page:

- The file name of the uploaded audio
- The predicted result: “Real” or “Fake”
- A confidence percentage, rounded to three decimal places, indicating the system’s certainty

4. SYSTEM ARCHITECTURE

The audio deepfake detection system follows a structured pipeline beginning with the **User Interface**, which is built using HTML, CSS, and JavaScript. This frontend provides various interactive web pages such as the homepage, login, registration, and the model upload form. Users interact with the system primarily through this interface. When a user uploads an audio file, the request is handled by the **Flask Web Server**, a lightweight Python-based backend framework that manages routing and processes user input. The audio file is uploaded via an HTML form using the

POST method, allowing the file to be streamed directly into the backend without being saved to disk, ensuring faster processing.

Once the file is uploaded, the system performs **similarity analysis** by extracting key audio features using the Librosa library. These features include MFCCs (to capture timbre), Mel spectrogram (for energy distribution), chroma features (to detect pitch), zero-crossing rate (to measure noisiness), spectral centroid (to evaluate brightness), and spectral flatness (to indicate tone versus noise). These features are combined into a feature vector that represents the unique characteristics of the uploaded audio. Additionally, the system supports an **optional SQLite database**, which is used for **user login authentication**. Registered users are verified before accessing the model, and their credentials are securely managed in the REGISTER table.

The extracted feature vector is then compared against a **preloaded dataset** (from a CSV file) that contains labeled audio samples.This second step of similarity analysis helps determine whether the uploaded audio is real or fake. Finally, the system provides a **prediction result**, displaying the file name "Real" or "Fake," and a confidence percentage based on how similar the input is to existing data. This complete flow ensures an efficient, real-time, and user-friendly experience for detecting deepfake audio.

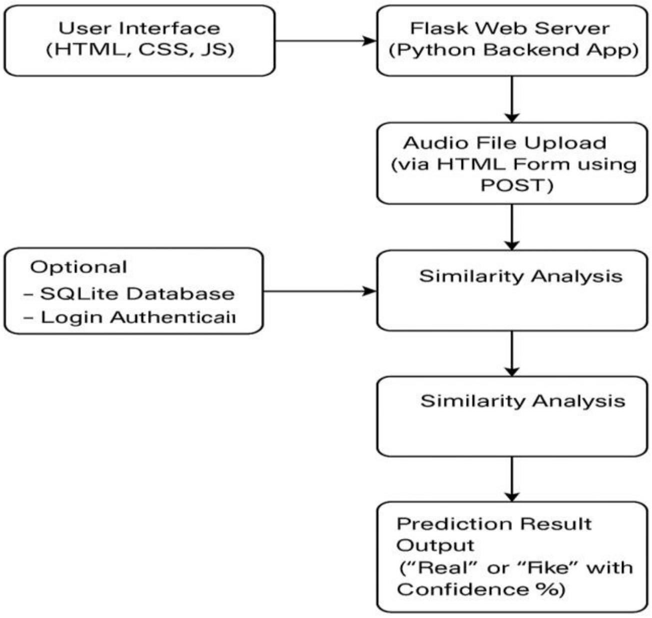
User Interface (HTML, CSS, JS)
Flask Web Server (Python Backend App)
Audio File Upload (via HTML Form using POST)
Optional
– SQLite Database
– Login Authenticaiı
Similarity Analysis
Similarity Analysis
Prediction Result Output ("Real" or "Fike" with Confidence %)

Figure 1: System Architecture

5. MODELLING AND ANALYSIS

1. Data Collection Module

- The app uses a pre-existing dataset.csv file that contains labeled real and fake audio samples.
- While there's no live collection or labeling mechanism, using a static dataset is an acceptable implementation for this module in many systems.

2. Audio Preprocessing Module

- The audio file uploaded by the user is: Loaded into a consistent format using librosa.load().
- Converted to waveform. While advanced preprocessing like noise removal or volume normalization isn't explicitly coded, basic conversion and preparation for feature extraction is done.

3. Feature Extraction Module

- Audio features are extracted using Librosa: MFCC, Mel Spectrogram, Chroma, Zero Crossing Rate Spectral Centroid**,** Spectral Flatness

4. Model Training Module

- While there is no machine learning algorithm trained within the code, the system uses:
- A distance-based similarity model (nearest-neighbor logic).
- The CSV dataset acts as a "trained reference" of feature vectors.

5. Prediction & Detection Module

- After feature extraction, the app: Computes distances between the uploaded file and dataset entries. Picks the closest match to determine if it's real or fake.
- Calculates a confidence score. The result is then displayed to the user

6. CONCLUSION

This project addresses the growing challenge of detecting deepfake content, which poses serious risks to privacy, security, and trust in digital media. By carefully collecting data, applying systematic preprocessing steps, and training detection models, the project demonstrates how thoughtful analysis and design can effectively distinguish genuine content from manipulated media. Through structured phases—ranging from data preparation to deployment—the project highlights the importance of planning, experimentation, and evaluation in developing reliable solutions. Overall, this work not only helps in understanding how deepfakes can be identified but also contributes to broader efforts to protect individuals and society from misinformation and digital deception**.**

By integrating a user-friendly interface with a Flask backend, the system enables real-time audio uploads and analysis. Through the extraction of key audio features such as MFCCs, Mel spectrogram, chroma, and spectral descriptors using the Librosa library, the system compares uploaded samples against a labeled dataset using similarity metrics like Euclidean distance. The inclusion of user authentication via SQLite enhances usability and security. Ultimately, the system delivers fast and interpretable predictions—labeling audio as either "Real" or "Fake" with a confidence score—making it a valuable tool for researchers, educators, and organizations concerned with media integrity. Its lightweight design also makes it adaptable for real-world applications, including mobile and edge devices, where computational resources may be limited.

7. REFERENCE

[1] Kumari et al., "Voice Radar: Voice Deepfake Detection using Micro Frequency and Compositional Analysis" (NDSS 2025).

[2] Multiple Authors – RawNetLite: End-to-end Audio Deepfake Detection from RAW Waveforms – 2025.

[3] Multiple Authors – ALLM4ADD: Unlocking the Capabilities of Audio Large Language Models for Audio Deepfake Detection – 2025.

[4] Multiple Authors – WaveGuard: Robust Deepfake Detection and Source Tracing via Dual-Tree Complex Wavelet and Graph Neural Networks – 2025.

[5] Yang Xiao, Rohan Kumar Das – Listen, Analyze, and Adapt to Learn New Attacks: An Exemplar-Free Class Incremental Learning Method for Audio Deepfake Source Tracing – 2025

[6] Multiple Authors – Detect All-Type Deepfake Audio: Wavelet Prompt Tuning for Enhanced Auditory Perception – 2025 .

[7] Multiple Authors – SafeEar: Content Privacy-Preserving Audio Deepfake Detection – 2024.

[8] Detection: A Literature Review Sayed Shifa Mohd Imran1, Dr. Pallavi Devendra Tawde2 1Student, Department of MSc.IT, Nagindas Khandwala College, Mumbai, Maharashtra in March 2024.

[9] Yujie Chen et al. – Region-Based Optimization in Continual Learning for Audio Deepfake Detection – 2024.

[10] Feiyi Dong, Qingchen Tang et al. – Advancing Continual Learning for Robust Deepfake Audio Classification (CADE) – 2024.