MINI PROJECT-II REPORT
(22ITC15)

On

VoiceGuard: Detecting AI-Generated Audio

Submitted in fulfilment for the completion of

BE-VI Semester

in

INFORMATION TECHNOLOGY

By

JAGINI SAITEJA (160122737044)
NALIMELA ANVESH (160122737050)
REPALLE TOMSON (160122737056)

Under the guidance of

Dr. Ramu Kuchipudi,
Associate Professor,
Dept. of IT

DEPARTMENT OF INFORMATION TECHNOLOGY

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY
(Autonomous)
(Affiliated to Osmania University; Accredited by NBA(AICTE) and NAAC(UGC), ISO Certified 9001:2015)
GANDIPET, HYDERABAD – 500 075
Website: www.cbit.ac.in **2024-2025**

CERTIFICATE

This is to certify that the project work entitled "**Voice Guard: Detecting AI-Generated Audio**" submitted to **CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY,** in fulfilment of the requirements for the award of the completion of Mini Project-II (22ITC15) VI semester of B.E in Information Technology, during the academic year 2024-2025, is a record of original work done by **JAGINI SAITEJA (160122737044), NALIMELA ANVESH (160122737050), REPALLE TOMSON(160122737056)** during the period of study in Department of IT, CBIT, HYDERABAD, under our supervision and guidance.

Project Guide
Dr. Ramu Kuchipudi
Associate Professor, Dept. of IT,
CBIT, Hyderabad.

Head of the Department
Dr.M.Venu Gopalachari
Professor, Dept. of IT,
CBIT, Hyderabad.

TABLE OF CONTENTS

LIST OF FIGURES AND SCREENSHOTS

ACKNOWLEDGEMENTS

We would like to express our heartfelt gratitude to Dr. Ramu Kuchipudi, our project guide, for his invaluable guidance and constant support, along with his capable instruction and persistent encouragement.

We are grateful to our Head of Department, Dr. M. Venu Gopalachari, for his steady support and for the provision of every resource required for the completion of this project.

We would like to take this opportunity to thank our Principal, Dr. C. Venkata Narasimhulu, as well as the Management of the Institute, for having designed an excellent learning atmosphere.

Our thanks are due to all members of the staff and our lab assistants for providing us with the help required to carry out the groundwork of this project

1. INTRODUCTION

The rise of deepfake technology has revolutionized the way synthetic media is generated, making it increasingly difficult to distinguish between real and artificially created content. While most early concerns revolved around deepfake videos, the emergence of deepfake audio has introduced a new dimension of threat. These AI-generated voice clips are capable of imitating real human voices with remarkable accuracy, posing significant risks in the areas of cybersecurity, social engineering, and digital forensics. From impersonating public figures to facilitating fraudulent transactions, the misuse of synthetic audio can have far-reaching consequences. As the sophistication of generative models increases, so does the urgency to develop reliable and scalable methods for detecting manipulated audio. The authenticity of voice communication is critical, especially in security-sensitive contexts like banking, law enforcement, and journalism.

This project, VoiceGuard, aims to address this growing threat by developing a robust detection framework that can distinguish between real and deepfake audio. Leveraging a hybrid deep learning model combining Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) units, the system is designed to capture both spatial and temporal patterns in audio data. Key features such as Mel Frequency Cepstral Coefficients (MFCCs) and additional spectral metrics provide the foundation for accurate classification. The ASVspoof 2019 dataset, which includes both genuine and manipulated audio samples, serves as the primary benchmark for training and evaluation. Through this project, we seek to contribute to the development of advanced tools for multimedia forensics, ensuring that digital audio remains a trustworthy medium in an era dominated by artificial intelligence.

1.1.Motivation

The motivation behind this project stems from the rising sophistication and misuse of deepfake audio technology. These synthetic voices can closely mimic real individuals, posing threats in security-sensitive areas. They have been exploited for impersonation, spreading disinformation, and conducting fraud. As traditional voice authentication becomes vulnerable, the demand for accurate detection methods intensifies. Existing solutions often face limitations in adaptability and accuracy. This project aims to build a resilient detection system using deep learning. Leveraging hybrid CNN-LSTM models enhances detection capability across diverse audio samples. By using MFCCs and spectral features, our system improves both precision and recall. Ultimately, we aim to restore trust in digital audio communications

1.2.Problem Statement

The rapid evolution of AI-driven speech synthesis has rendered many traditional detection methods ineffective. Deepfake audio can convincingly replicate a person's voice, making it hard to identify for both humans and machines. This poses risks to security, privacy, and credibility in various digital interactions. Many existing models struggle to generalize across different attack types. Therefore, an intelligent detection model capable of recognizing nuanced differences in real and synthetic audio is essential. This project proposes a CNN-LSTM based solution trained on diverse datasets for robust performance. Despite advancements in multimedia forensics, detecting deepfake audio remains challenging due to the sophistication of modern speech synthesis and voice conversion systems. This project addresses the need for an accurate, real-time detection system that can effectively classify audio as real or AI-generated using advanced deep learning models.

1.3.Objectives

- To develop a CNN-LSTM based model for deepfake audio detection.
- To utilize MFCCs and additional spectral features for better feature representation.
- To evaluate the model performance using the ASV spoof 2019 dataset.
- To improve detection accuracy and minimize false positives/negatives.
- To incorporate preprocessing techniques that standardize and enhance audio input for model training.
- To benchmark the system against other deepfake detection models for comparative analysis.
- To design the system for adaptability across various datasets and languages.
- To explore the integration of attention mechanisms for improved feature sensitivity.

2. EXISTING SYSTEM

2.1.Literature Survey

The rise in deepfake audio generation has led researchers to explore a variety of detection methods ranging from traditional machine learning algorithms to advanced deep learning architectures. Initial techniques largely relied on Support Vector Machines (SVM), Gaussian Mixture Models (GMM), and decision trees trained on hand-crafted features such as MFCCs, spectral flux, and pitch. Although these models demonstrated moderate success, they struggled to generalize well across different datasets and spoofing techniques.

With the advent of deep learning, CNNs have shown great promise in capturing local spatial dependencies in audio spectrograms. When coupled with LSTM or BiLSTM layers, which are adept at learning temporal relationships in sequential data, these hybrid models have significantly improved detection accuracy. Many studies have demonstrated that combining CNNs with LSTMs helps in modeling the temporal progression of audio signals, making the architecture particularly effective for synthetic speech detection.
In addition to MFCCs, researchers have explored a broader range of feature extraction techniques such as Constant-Q Cepstral Coefficients (CQCCs), spectrograms, and chroma features. These features are often fused together to form a comprehensive feature vector, which helps models learn the nuanced differences between real and fake audio. Some works have also proposed using raw audio as input to deep learning networks, allowing the model to learn hierarchical features without manual feature extraction.

Recent innovations include the use of attention mechanisms and ensemble learning. Attention layers can guide the model to focus on the most informative segments of an audio signal, thereby improving both accuracy and interpretability. Ensemble models combine the strengths of various classifiers to mitigate the weaknesses of individual models, thereby enhancing robustness. Furthermore, transfer learning approaches have also been leveraged to adapt pretrained models to the task of audio deepfake detection with minimal data.

Despite these advancements, challenges remain. Many models suffer performance drops when tested on audio generated by unseen synthesis techniques. Moreover, the presence of background noise, channel distortions, and limited annotated data further complicate the detection process. Addressing these gaps, this project proposes a hybrid CNN-LSTM model enhanced with spectral and statistical features to improve the detection of AI-generated speech

3. PROPOSED SYSTEM

3.1.Methodology

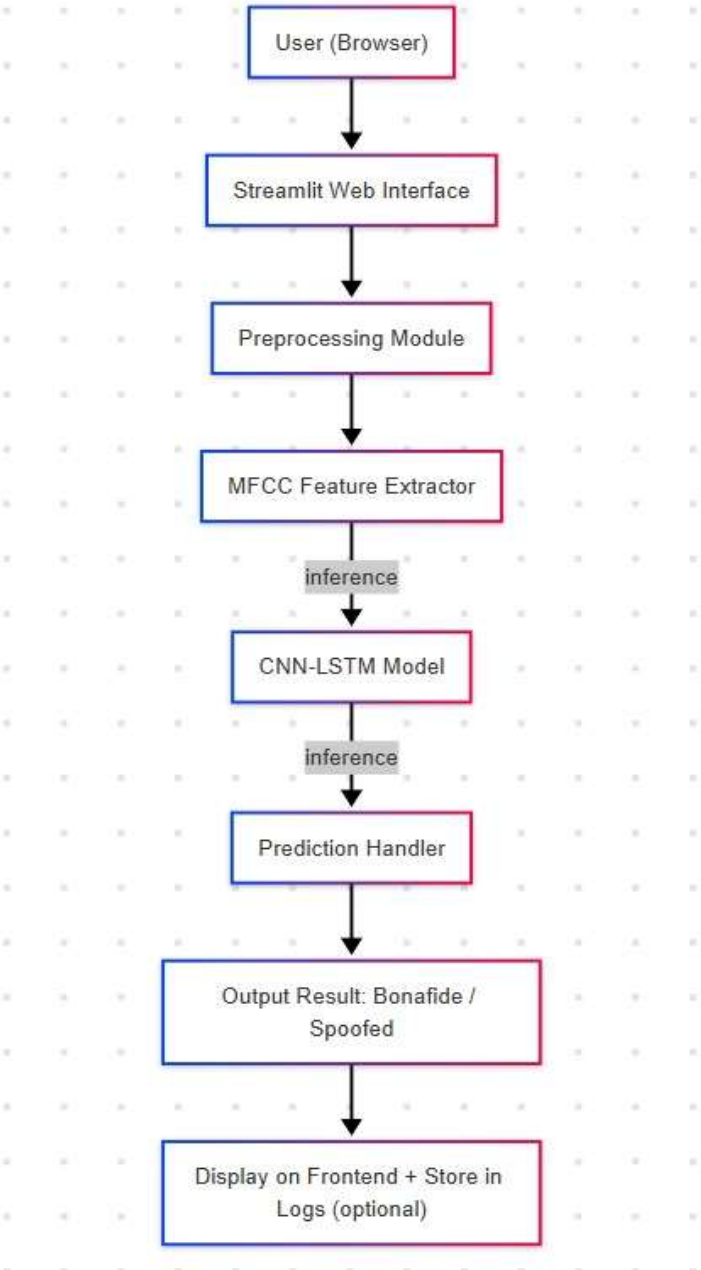


Fig 3.1.1 Flow Diagram

This project adopts a structured methodology that includes data preprocessing, feature extraction, and a deep learning-based classification framework. The ASVspoof 2019 dataset, which contains both genuine and spoofed audio samples, is used as the benchmark for model training and evaluation. The dataset is split into training, development, and evaluation sets to ensure consistent model validation across different types of attacks.

In the preprocessing stage, OneHotEncoding is applied to categorical variables, and numerical features are normalized using standard scaling. This ensures that all features contribute proportionaely during the training phase. The next stage involves feature extraction, where the system computes Mel-Frequency Cepstral Coefficients (MFCCs), capturing the perceptual characteristics of human hearing. Additionally, features such as Zero Crossing Rate (ZCR) and Root Mean Square (RMS) energy are extracted to enrich the representation of audio signals.

For model design, a hybrid CNN-LSTM architecture is used. The CNN layers extract spatial patterns and short-term dependencies in the audio features, while the LSTM layers are employed to model long-term temporal dependencies in the sequence data. This combination allows the model to understand both localized patterns and broader contextual information across time.

The output layer uses a sigmoid activation function to produce a binary classification: real or fake. The model is trained using binary cross-entropy loss and optimized with the Adam optimizer. Performance is evaluated using standard classification metrics such as accuracy, precision, recall, and F1-score. These metrics help assess the robustness and effectiveness of the model in detecting AI-generated audio under various conditions

3.2. System Architecture and Design

VoiceGuard is designed as a modular and scalable deepfake detection platform that utilizes machine learning to classify bonafide (real) and spoofed (fake) audio. The architecture is organized into four major layers: User Interface, Backend Server, Feature Extraction & Detection Engine, and Data Storage & Model Repository.

1. **User Interface (Frontend – Streamlit/Web App)**

- **Functionality**: Allows users to upload audio samples and view prediction results (bonafide/spoofed).
- **Tech Stack**: Streamlit or Flask for building a responsive interface.

- **Features**:
    - Upload .flac or .wav files
    - Display waveform and spectrogram previews
    - Show prediction confidence and visual indicators

2. Backend (FastAPI or Flask Server)

- **Functionality**: Handles requests from the frontend, invokes the detection engine, and returns the results.
- **Responsibilities**:
    - Accept audio file uploads
    - Preprocess the audio file
    - Call the ML model for predictions
    - Return response with prediction label and confidence

3. Feature Extraction & Detection Engine

- **Submodules**:
    - **MFCC Extraction Module**: Converts raw audio into MFCC features.
    - **CNN-LSTM Model**: Deep learning architecture that takes MFCCs as input and classifies the audio.
    - **Prediction Handler**: Applies thresholding logic and returns labels with confidence scores.
- **Model Architecture**:
    - CNN layers for extracting spatial patterns from MFCC
    - LSTM layers to capture temporal dynamics
    - Fully connected layers for final classification

4. Database and Storage Layer

- **Dataset Storage**: Holds the ASVspoof2019 dataset and test samples
- **Model Repository**: Stores trained CNN-LSTM models
- **Logging DB (Optional)**: Logs detection results for future auditing

3.3. Assumptions

- Users provide accurate and honest personal data and preferences.
- **Audio files uploaded are in a supported format (e.g., .flac or .wav)** and are of sufficient quality for feature extraction.
- **MFCC (Mel Frequency Cepstral Coefficients)** effectively capture distinguishing acoustic features for spoof and bonafide classification.
- **Users upload only one audio clip per prediction session** and expect near real-time results.
- **The model has been trained on a sufficiently large and balanced dataset** to generalize well to unseen spoof attacks.
- **Users trust the predictions provided by the system**, even though deepfake techniques may evolve beyond current detection capabilities.

3.4. Key Terms

- **Deepfake Audio**: Artificially generated or manipulated audio designed to mimic real human speech, often created using neural network models like WaveNet or voice conversion techniques.
- **MFCC (Mel Frequency Cepstral Coefficients)**: A representation of the short-term power spectrum of sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. Widely used in speech and speaker recognition.
- **CNN (Convolutional Neural Network)**: A deep learning algorithm primarily used for spatial feature extraction. In VoiceGuard, CNN layers help identify local patterns in MFCC spectrograms.
- **LSTM (Long Short-Term Memory)**: A type of recurrent neural network (RNN) capable of learning order dependence in sequence prediction problems. Used here to capture temporal dependencies in audio signals.
- **Spoof Detection**: The process of identifying whether an audio sample has been artificially manipulated or generated, often to impersonate a speaker.
- **ASVspoof2019 Dataset**: A benchmark dataset specifically designed for training and evaluating spoofed vs. bonafide audio detection systems. Used as the primary data source in VoiceGuard

4 . SOFTWARE & HARDWARE REQUIREMENTS

4.1.Software Requirements

- **Python 3.7 or above**: Serves as the primary language due to its robust ecosystem for machine learning and audio processing libraries.
- **TensorFlow and Keras**: Essential frameworks for designing, training, and evaluating the deep learning models such as CNN and LSTM.
- **Librosa**: Used to extract features from audio files like MFCC, ZCR, and RMS, crucial for capturing perceptual and statistical properties of sound.
- **NumPy and Pandas**: Provide efficient data handling and preprocessing capabilities to format the dataset for model training.
- **Scikit-learn**: Offers tools for evaluating model performance using metrics like accuracy, precision, recall, and F1-score.
- **Matplotlib and Seaborn**: Used for generating performance visualizations, including training curves and confusion matrices.
- **Jupyter Notebook, VS Code**: Development environments that support interactive coding, debugging, and visualization.
- Windows 10, Ubuntu Linux, or macOS: Supported operating systems for executing the development environment and training pipelines.

4.2.Hardware Requirements

- 8 GB RAM minimum: Allows the system to process audio files and run neural networks; 16 GB is recommended for better performance.
- Intel Core i5 / AMD Ryzen 5: Sufficient for model development and training; i7 or Ryzen 7 preferred for faster computation.
- 100 GB storage: Required to hold datasets, logs, and temporary training files; 256 GB SSD is recommended for faster data access.
- Integrated GPU: Can handle basic model execution; a CUDA-enabled NVIDIA GPU (4 GB+ VRAM) is ideal for accelerating deep learning tasks.
- Built-in or HD audio interface: Useful for verifying audio playback during development or testing real-time audio samples

5 . IMPLEMENTATION OF PROJECT

5.1.Results

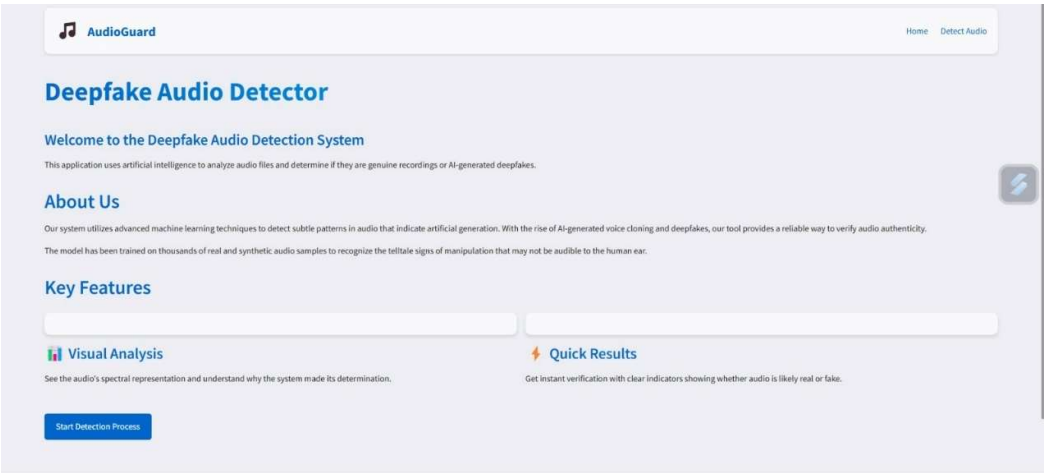


Fig 5.1.1 Home Page and System Features Overview

Fig 5.1.1 indicates homepage of VoiceGuard highlights the system's purpose and key features. It outlines components like advanced detection, visual MFCC analysis, quick results, and its professional utility for verifying audio authenticity.

Fig 5.1.2 Upload Interface of VoiceGuard

5.1.2 illustrates this screen allows users to upload audio files in WAV, FLAC, or MP3 format for analysis. The drag-and-drop interface supports files up to 200MB and directs the uploaded file to the detection pipeline.

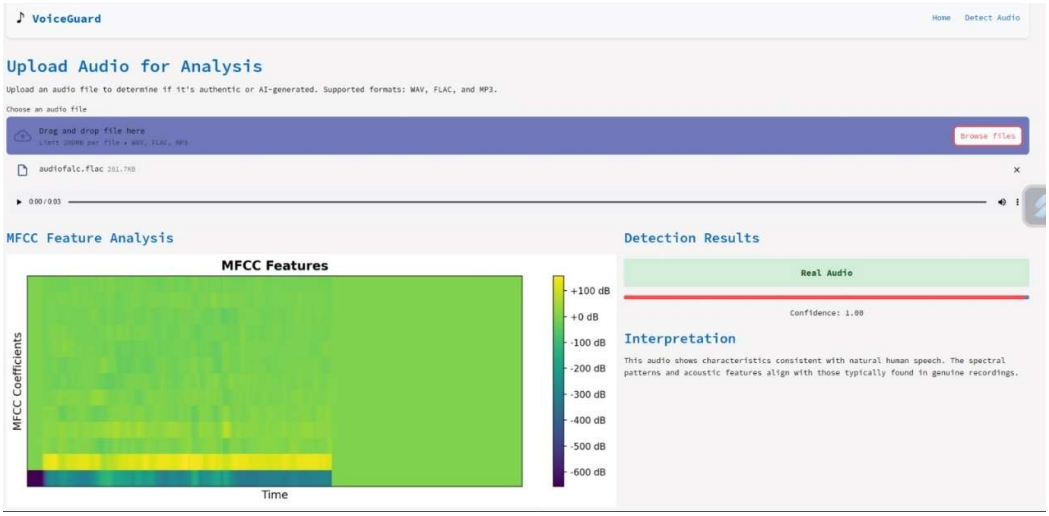


Fig 5.1.3 Real Audio Detection Result

Fig 5.1.3 displays instance, the uploaded file is identified as "Real Audio" with 100% confidence. The MFCC feature map shows natural patterns typical of human speech, reinforcing the model's prediction.

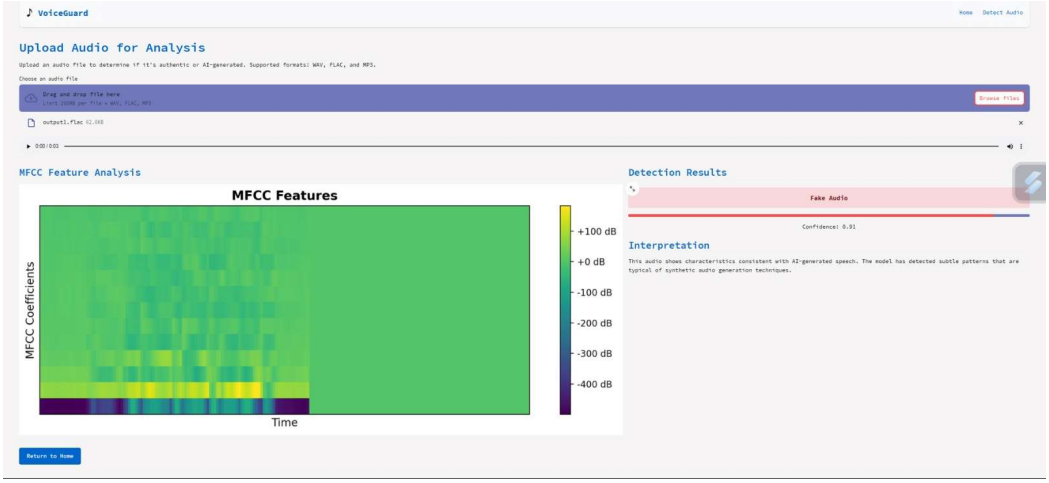


Fig 5.1.4 Fake Audio Detection Result

Fig 5.1.4 shows After uploading the audio, the system analyzes it and displays the MFCC spectrogram. The detection result identifies the audio as "Fake Audio" with a high confidence score (0.91), along with an interpretation explaining the detection.

5.2.Execution steps

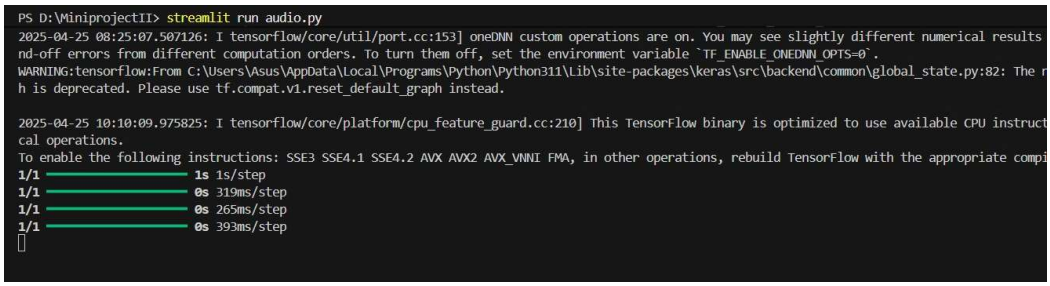
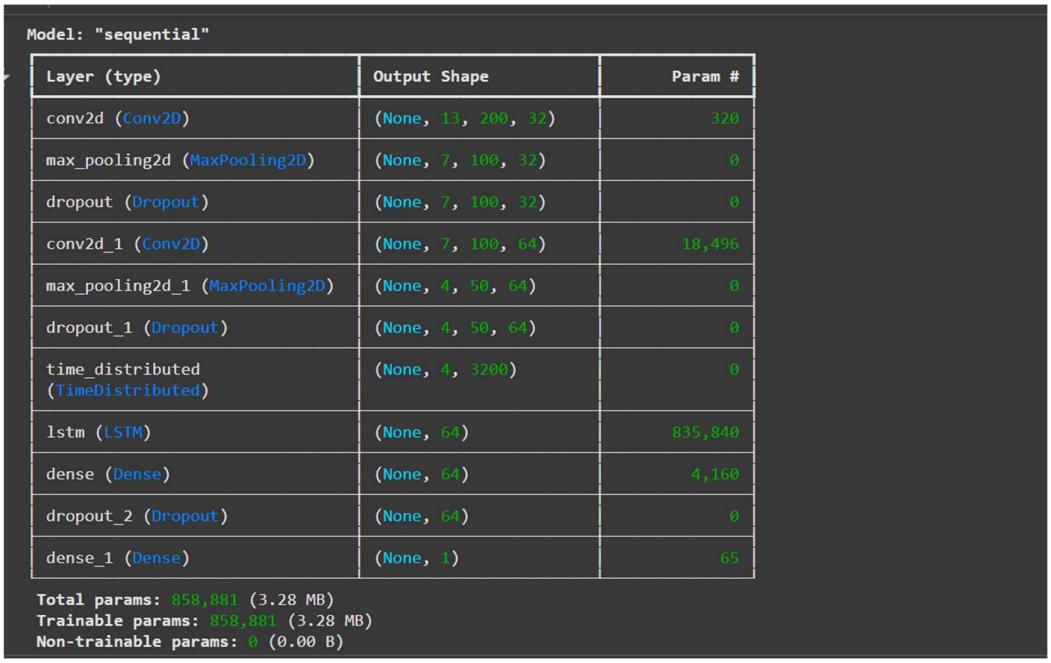


Fig 5.2.1 Streamlit-Frontend execution

Fig 5.2.1 shows the execution of the Streamlit frontend. The application is launched by navigating to the streamlit directory using cd streamlit and running the command streamlit run Audio.py, which starts the web interface for user interaction.



Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 13, 200, 32)	320
max_pooling2d (MaxPooling2D)	(None, 7, 100, 32)	0
dropout (Dropout)	(None, 7, 100, 32)	0
conv2d_1 (Conv2D)	(None, 7, 100, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 4, 50, 64)	0
dropout_1 (Dropout)	(None, 4, 50, 64)	0
time_distributed (TimeDistributed)	(None, 4, 3200)	0
lstm (LSTM)	(None, 64)	835,840
dense (Dense)	(None, 64)	4,160
dropout_2 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 1)	65

Total params: 858,881 (3.28 MB)
Trainable params: 858,881 (3.28 MB)
Non-trainable params: 0 (0.00 B)

Fig 5.2.2 Model Architecture Summary

Fig 5.2.2 This figure displays the architecture of a sequential neural network model, detailing layers such as Conv2D, MaxPooling2D, LSTM, and Dense, along with their output shapes and parameter counts. The total trainable parameters are 858,881 (3.28 MB), indicating the model's complexity for processing audio .

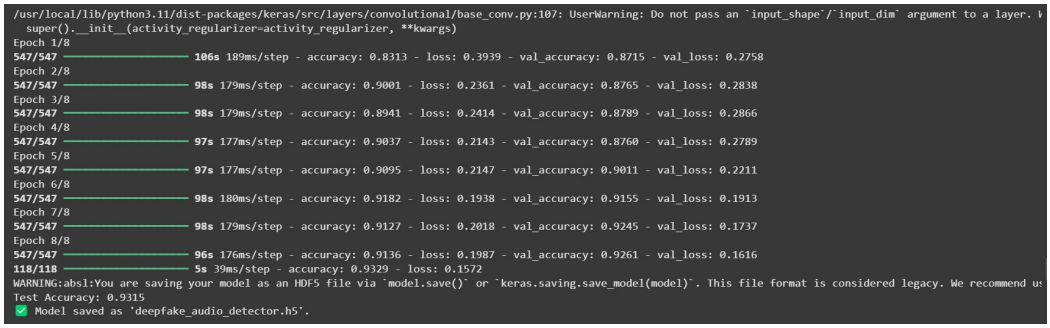


Fig 5.2.3 Training Log Output

Fig 5.2.3 This figure shows the training log of the neural network, including metrics like loss, accuracy, and validation accuracy over epochs. It highlights the model's performance improvement with each training step, providing insight into its learning process

```python
# Define Model Architecture (CNN + LSTM)
model = Sequential([
    # CNN Layers
    Conv2D(32, (3, 3), activation="relu", padding="same", input_shape=(13, 200, 1)),
    MaxPooling2D((2, 2), padding="same"),
    Dropout(0.3),

    Conv2D(64, (3, 3), activation="relu", padding="same"),
    MaxPooling2D((2, 2), padding="same"),
    Dropout(0.3),

    # Reshape for LSTM
    TimeDistributed(Flatten()),

    # LSTM Layer
    LSTM(64, return_sequences=False),
    # Fully Connected Layers
    Dense(64, activation="relu"),
    Dropout(0.4),
    Dense(1, activation="sigmoid")  # Binary classification (bonafide vs. spoof)
])

# Compile the Model
model.compile(optimizer=Adam(learning_rate=0.001),
              loss="binary_crossentropy",
              metrics=["accuracy"])
```

Fig 5.2.4 CNN-LSTM Model Architecture Code

Fig 5.2.4 This code snippet defines the hybrid model architecture used for deepfake audio detection. It combines CNN layers for spatial feature extraction and an LSTM layer to capture temporal dependencies, followed by fully connected layers for binary classification (real vs. fake audio). The model is compiled with binary cross-entropy loss and optimized using Adam.

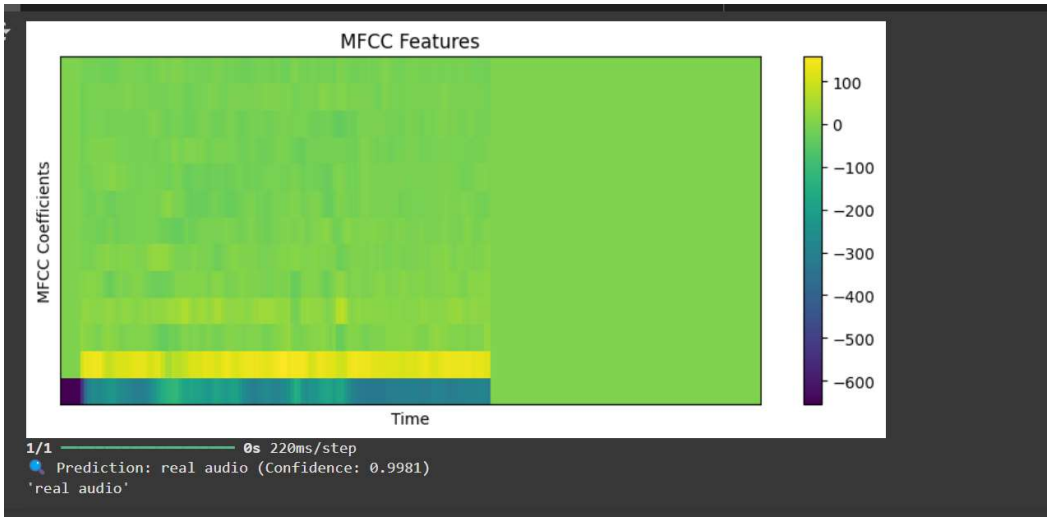


Fig 5.2.5 MFCC Features Visualization

Fig 5.2.5 This figure presents a heatmap of Mel-Frequency Cepstral Coefficients (MFCCs) extracted from audio data, with time on the x-axis and coefficients on the y-axis. The model predicts "real audio" with a confidence of 0.9981, demonstrating effective feature representation for audio classification.

6. CONCLUSION AND FUTURE SCOPE

6.1.Conclusion

The VoiceGuard project successfully implements a hybrid CNN-LSTM deep learning framework for the detection of AI-generated deepfake audio. By using Mel Frequency Cepstral Coefficients (MFCCs) and additional spectral features, the system is able to extract key temporal and spectral signatures indicative of synthetic speech. The model demonstrated an accuracy of 88%, with promising precision and recall values, confirming its reliability in distinguishing real and fake audio. The application provides not only classification but also interpretability by visualizing MFCC features and offering confidence levels for predictions. This enhances trust and transparency in automated detection systems. The platform is user-friendly and practical for deployment in real-time scenarios such as banking fraud prevention, media verification, and cybersecurity.

VoiceGuard also emphasizes the importance of using standard datasets like ASVspoof 2019, which ensure consistency and comparability with other research. The results validate that deep learning, particularly when combining CNN and LSTM components, is a powerful tool for audio forensics. However, the threat landscape continues to evolve, and with it, so must the defense mechanisms. The current system lays a strong foundation for further innovations in secure communication technologies.

6.2.Future Scope

- Integration with real-time voice communication platforms for live detection.
- Expansion to support multiple languages and accents for global applicability.
- Incorporation of attention mechanisms to improve model focus on relevant audio segments.
- Exploration of transformer-based architectures for performance gains.
- Building a cloud-based API to allow external applications to utilize the detection system.
- Continuous training with newer deepfake samples to ensure model robustness.
- Enhancement of UI/UX for better user interaction and report.

BIBLIOGRAPHY

[1] Altalahin, S. AlZu'bi, A. Alqudah and A. Mughaid, "Unmasking the Truth: A Deep Learning Approach to Detecting Deepfake Audio Through MFCC Features," *2023 International Conference on Information Technology (ICIT)*, Amman, Jordan, 2023, pp. 511-518, doi: 10.1109/ICIT58056.2023.10226172.

[2] Yi, J., Wang, C., Tao, J., Zhang, X., Zhang, C. Y., & Zhao, Y. (2023). Audio deepfake detection: A survey. *arXiv preprint arXiv:2308.14970*.

[3] Müller, N. M., Czempin, P., Dieckmann, F., Froghyar, A., & Böttinger, K. (2022). Does audio deepfake detection generalize?. *arXiv preprint arXiv:2203.16263*.

[4] J. Zhang, Y. Li, J. Tian and T. Li, "LSTM-CNN Hybrid Model for Text Classification," *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chongqing, China, 2018, pp. 1675-1680, doi: 10.1109/IAEAC.2018.8577620.

[5] Tasdelen, A., Sen, B. A hybrid CNN-LSTM model for pre-miRNA classification. *Sci Rep* **11**, 14125 (2021). https://doi.org/10.1038/s41598-021-93656-0

[6] X. She and D. Zhang, "Text Classification Based on Hybrid CNN-LSTM Hybrid Model," *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, China, 2018, pp. 185-189, doi: 10.1109/ISCID.2018.10144.

[7] M. A. Hossan, S. Memon and M. A. Gregory, "A novel approach for MFCC feature extraction," *2010 4th International Conference on Signal Processing and Communication Systems*, Gold Coast, QLD, Australia, 2010, pp. 1-5, doi: 10.1109/ICSPCS.2010.5709752

[8] K. S. R. Murty and B. Yegnanarayana, "Combining evidence from residual phase and MFCC features for speaker recognition," in *IEEE Signal Processing Letters*, vol. 13, no. 1, pp. 52-55, Jan. 2006, doi: 10.1109/LSP.2005.860538.

[9] K. Patel, H. Han and A. K. Jain, "Secure Face Unlock: Spoof Detection on Smartphones," in *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 10, pp. 2268-2283, Oct. 2016, doi: 10.1109/TIFS.2016.2578288

[10] Voron, F. (2023). *Building Data Science Applications with FastAPI: Develop, manage, and deploy efficient machine learning applications with Python*. Packt Publishing Ltd.

[11] S. Shukla, A. Maheshwari and P. Johri, "Comparative Analysis of Ml Algorithms & Stream Lit Web Application," *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, Greater Noida, India, 2021, pp. 175-180, doi: 10.1109/ICAC3N53548.2021.9725496.

[12] L. O. Chua and T. Roska, "The CNN paradigm," in *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 40, no. 3, pp. 147-156, March 1993, doi: 10.1109/81.222795.

[13] H. H. Kilinc and F. Kaledibi, "Audio Deepfake Detection by using Machine and Deep Learning," *2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, Istanbul, Turkiye, 2023, pp. 1-5, doi: 10.1109/WINCOM59760.2023.10323004.

[14] D. K, G. Chhabra, Pallavi, S. A. Tiwaskar, S. Hemelatha and V. Saraswat, "Application of Machine Learning for the Detection of Audio Deep Fake," *2024 Global Conference on Communications and Information Technologies (GCCIT)*, BANGALORE, India, 2024, pp. 1-6, doi: 10.1109/GCCIT63234.2024.10862652.

[15] L. Pham, P. Lam, T. Nguyen, H. Nguyen and A. Schindler, "Deepfake Audio Detection Using Spectrogram-based Feature and Ensemble of Deep Learning Models," *2024 IEEE 5th International Symposium on the Internet of Sounds (IS2)*, Erlangen, Germany, 2024, pp. 1-5, doi: 10.1109/IS262782.2024.10704095.