**“AUDIO DEEPFAKE DETECTION”**

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF TECHNOLOGY

IN

#### Electronics and Communication Engineering

Submitted by:

#### Mohd. Rizwan (2K21/EC/141)

#### Niresh Kumar (2K21/EC/150)

Under the supervision of

Prof. O.P Verma

A circular logo with text and words

Description automatically generated

#### DEPT. OF ELECTRONICS AND COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering) Bawana Road, Delhi – 110042

#### MAY 2025

**DEPT. OF ELECTRONICS AND COMMUNICATION ENGINEERING**

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering) Bawana Road, Delhi – 110042

#### CANDIDATE’S DECLARATION

We, Mohd.Rizwan (2K21/EC/141) and Niresh Kumar (2K21/EC/150) students of B.Tech (Electronics and Communication Engineering), hereby declare that the Project Dissertation titled — “AUDIO DEEPFAKE DETECTION” which is submitted by us to the Department of Electronics and Communication Engineering, DTU, Delhi in fulfillment of the requirement for awarding of the Bachelor of Technology degree, is not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma, Fellowship or other similar title or recognition.

|  |  |  |
| --- | --- | --- |
| Place : Delhi | **MOHD RIZWAN** | **NIRESH KUMAR** |
| Date : 26/05/2025 | **(2K21/EC/141)** | **(2K21/EC/150)** |

#### DEPT. OF ELECTRONICS AND COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering) Bawana Road, Delhi – 110042

#### CERTIFICATE

I hereby certify that the Project titled “AUDIO DEEPFAKE DETECTION” which is submitted by Mohd. Rizwan (2K21/EC/141) and Niresh Kumar (2K21/EC/150) for fulfillment of the requirements for awarding of the degree of Bachelor of Technology (B.Tech) is a record of the project work carried out by the students under my guidance & supervision. To the best of my knowledge, this work has not been submitted in any part or fulfillment for any Degree or Diploma to this University or elsewhere.

Place : New Delhi Prof. O P Verma

Date : 26/05/2025 **(SUPERVISOR)**

Professor

Department of Electronics and Communication

Delhi Technological University

**ABSTRACT**

This project tackles the growing threat of audio deepfakes—synthetic speech generated by deep learning models that closely mimic human voices. As these technologies evolve, their misuse in fraud, impersonation, and misinformation raises serious concerns, calling for effective detection methods.

We evaluate multiple machine learning models for deepfake detection, including GMM, RNN, Bi-LSTM, Shallow CNN, and TSSD. Using LJSpeech for real speech and WaveFake for synthetic audio, we test these models in both in-distribution and out-of-distribution scenarios. Various audio feature representations, such as MFCCs and spectrograms, are compared to assess detection performance.

Our experiments reveal that temporal and spectral patterns play a key role in differentiating real and synthetic audio. We also observe that certain models generalize better to unseen voices, highlighting the importance of architecture choice. The results demonstrate the feasibility of building lightweight yet accurate detectors suitable for real-world deployment. These findings contribute toward enhancing the reliability of voice-based authentication and media verification systems.

**ACKNOWLEDGEMENT**

The successful completion of any task is incomplete and meaningless without giving any due credit to the people who made it possible without which the project would not have been successful and would have existed in theory.

First and foremost, we are grateful to Prof. O P Verma, HOD, Department of Electronics and Communication, Delhi Technological University, and all other faculty members of our department for their constant guidance and support, constant motivation and sincere support and gratitude for this project work. We owe a lot of thanks to our supervisor, Dr. O.P Verma Professor, Department of Electronics and Communication, Delhi Technological University for igniting and constantly motivating us and guiding us in the idea of a creatively and amazingly performed Minor project in undertaking this endeavor and challenge and for being there whenever we needed her guidance or assistance.

We would also like to take this moment to show our thanks and gratitude to one and all, who indirectly or directly have given us their hand in this challenging task. We feel happy and joyful and content in expressing our vote of thanks to all those who have helped us and guided us in presenting this project work for our Minor project. Last, but never least, we thank our well-wishers and parents for always being with us, in every sense and constantly supporting us in every possible sense whenever possible.

**Niresh Kumar Mohd Rizwan**

**(2K21/EC/150) (2K21/EC/141)**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **Title** | **Page No.** |
| **CHAPTER 1** | **INTRODUCTION** | **7-11** |
| 1.1 | Background and Motivation | **7** |
| 1.2 | Problem Statement | **8** |
| 1.3 | Research Challenges | **9** |
| 1.4 | Research Contributions | **10** |
| 1.5 | Related Works | **11** |
| **CHAPTER 2** | **DATASET** | **12-12** |
| 2.1 | Dataset Description | **12** |
| **CHAPTER 3** | **METHODS** | **13-17** |
| 3.1 | Audio Data Preprocessing and Processing Pipeline | **13** |
| 3.2 | Feature Extraction Methodologies | **14** |
| 3.3 | Classification Approaches | **15** |
| **CHAPTER 4** | **EXPERIMENTAL SETUP** | **18-18** |
| 4.1 | Training and Testing Setup | **18** |
| 4.2 | Loss Function | **18** |
| 4.3 | Hyper-parameters | **18** |
| **CHAPTER 5** | **RESULTS AND CONCLUSION** | **19-20** |
| 5.1 | Results | **19** |
| 5.2 | Analysis | **20** |
| 5.3 | Conclusion | **20** |
| **BIBLOGRAPHY** |  | **21** |

# TABLE OF CONTENTS

# CHAPTER 1 INTRODUCTION

#### Background and Motivation

### 1.1.1 The Evolution of Synthetic Speech Technology

The field of synthetic speech generation has undergone remarkable transformation over the past decade. Traditional text-to-speech (TTS) systems, which relied on concatenative synthesis and parametric approaches, have been largely superseded by neural network-based methods. Early neural approaches like WaveNet introduced autoregressive models for high-quality audio synthesis, while subsequent developments in Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) have enabled the creation of increasingly realistic synthetic speech.

Modern voice synthesis technologies, including Tacotron, FastSpeech, and various GAN-based architectures, can now produce audio that is virtually indistinguishable from genuine human speech to the untrained ear. These systems can clone voices with minimal training data, sometimes requiring only a few minutes of target speaker audio to generate convincing synthetic speech in their voice.

**1.1.2 The Democratization of Deepfake Technology**

What once required specialized knowledge and significant computational resources is now accessible to the general public through user-friendly applications and online services. Platforms and tools for voice cloning have proliferated, making synthetic speech generation available to both legitimate users and malicious actors. This democratization has accelerated the proliferation of audio deepfakes across various platforms and use cases.

The accessibility of these technologies has created a dual-edged scenario: while they offer tremendous potential for creative applications, accessibility services, and content creation, they simultaneously lower the barrier for malicious use. The same technology that enables personalized digital assistants and helps individuals with speech impairments can also be weaponized for fraud, misinformation, and social manipulation.

**1.1.3 The Arms Race: Synthesis vs Detection**

As synthesis technologies advance, so too must detection methods. This has created an ongoing "arms race" between increasingly sophisticated generation techniques and the detection systems designed to identify them. Early detection methods that worked well against simpler synthetic speech often fail when confronted with state-of-the-art generation models, necessitating continuous research and development in detection methodologies.

The challenge is further compounded by the fact that new synthesis methods are constantly emerging, each potentially exploiting different aspects of speech production and perception. Detection systems must therefore be robust enough to generalize across multiple synthesis techniques while remaining sensitive enough to identify subtle artifacts that distinguish synthetic from genuine speech.

**1.1.4 Societal Impact and Trust**

The proliferation of convincing audio deepfakes has broader implications for society's relationship with digital media. As the technology becomes more prevalent, there is a growing erosion of trust in audio evidence and recorded speech. This "liar's dividend" phenomenon allows malicious actors to dismiss authentic recordings as potentially fake, while simultaneously making it easier to spread disinformation through synthetic media.

The impact extends beyond individual cases of fraud or harassment to affect democratic institutions, judicial systems, and social cohesion. The ability to generate convincing fake audio of public figures, witnesses, or private individuals poses unprecedented challenges for information verification and truth establishment in the digital age.

#### 1.2Problem Statement

The rapid advancement of Artificial Intelligence (AI), particularly in the domain of deep learning, has facilitated the creation of highly convincing manipulated audio known as deepfakes. Audio DeepFakes refer to artificially generated audio that mimics human speech. These are created using Deep Neural Networks (DNNs) that learn from real audio samples to produce fake audio clips that sound realistic and human-like. While deepfake technology has applications in entertainment and digital media production, its misuse poses significant challenges and risks in various domains.

**1.2.1 Potential Threats and Misuse**

Although these can be used for entertainment, they have a high risk of misuse. Audio Deepfakes can be exploited for online harassment, political propaganda, impersonation, and misinformation campaigns.

**1.2.2 Real-World Concerns**

Combining audio Deepfakes with video Deepfakes has already shown real-world consequences. In 2024, a deepfake video of Ukrainian President Volodymyr Zelenskyy appeared on social media, falsely announcing a surrender to Russia [1]. Although it was quickly debunked, it demonstrated how such technology can be weaponized for information warfare and psychological manipulation in active geopolitical conflicts.

**1.2.3 Need for Detection Systems**

Due to the increasing threat posed by such technologies, there's a significant need to design and implement robust and reliable audio DeepFake detection systems.

**1.2.4 Project Objective**

This project aims to distinguish DeepFake audio generated by state-of-the-art GAN-based models from real audio. We explore different sequential modeling architectures and audio features for this detection task.

* 1. **Research Challenges**

**1.3.1 Technical Challenges**

* **Generalization Across Synthesis Methods:** Different synthesis techniques leave different artifacts and signatures in the generated audio. A detection system trained on one type of synthetic speech may fail when encountering audio generated by a different method. This cross-method generalization remains a significant challenge in the field.
* **Computational Constraints:** While detection accuracy is paramount, practical deployment scenarios often require real-time or near-real-time processing capabilities. Balancing detection performance with computational efficiency presents ongoing challenges, particularly for mobile or edge computing applications.
* **Data Scarcity and Imbalance:** High-quality labeled datasets for training detection systems are often limited, particularly for the most recent synthesis methods. Additionally, the rapid evolution of synthesis techniques means that training data may quickly become outdated as new generation methods emerge.
* **Adversarial Robustness:** Malicious actors may deliberately attempt to fool detection systems through adversarial perturbations or by specifically targeting known vulnerabilities in detection algorithms. Building robust defenses against such adaptive attacks remains an active area of research.

**1.3.2 Evaluation Challenges**

* **Cross-Dataset Generalization:** Models that perform well on specific datasets may fail when applied to audio from different sources, recording conditions, or synthesis methods. Establishing standardized evaluation protocols that reflect real-world deployment scenarios is crucial for meaningful performance assessment.
* **Temporal Degradation**: Detection systems may experience performance degradation over time as new synthesis methods emerge. Long-term evaluation and adaptation strategies are necessary to maintain effectiveness in evolving threat landscapes.
* **Ethical Considerations:** The development and deployment of detection systems must balance security needs with privacy concerns and potential for misuse. False positive rates in detection systems could lead to unjust accusations or censorship of legitimate content.
  1. **Research Contributions**

This work contributes to the field of audio deepfake detection through several key innovations and comprehensive evaluations:

**1.4.1 Methodological Contributions**

* Comprehensive comparison of hand-crafted feature approaches versus end-to-end learning methods
* Investigation of both traditional machine learning and deep learning architectures for detection
* Analysis of different audio representations and their effectiveness for detection tasks

**1.4.2 Experimental Contributions**

* Systematic evaluation across multiple synthesis methods and datasets.
* In-depth analysis of model performance under various acoustic conditions.
* Identification of challenging cases and failure modes across different detection approaches.

**1.4.3 Practical Contributions**

* Guidelines for selecting appropriate detection methods based on deployment constraints.
* Analysis of computational trade-offs between accuracy and efficiency.
* Recommendations for improving robustness in real-world scenarios.

#### 1.5 Related Works

Several models have been proposed to detect DeepFake audio and distinguish fake clips from real ones. These approaches range from simple binary classification to more nuanced, technique-specific detection strategies.

* + - 1. **Fine-Grained Supervision** [2]**:** This approach incorporates two classification blocks—one for binary classification (real vs fake), and another to identify the specific type of audio synthesis technique used. This dual-task setup enables the model to learn more detailed distinctions between various kinds of audio manipulations.
      2. **Speaker Recognition Feature Reuse** [3]**:** Instead of training a model from scratch, this method extracts hidden layer activation maps from a pre-trained speaker recognition model. These features are then fed into a lightweight classifier to make the final DeepFake prediction.
      3. **Shallow CNN-Based Detection** [4]**:** A shallower convolutional neural network is proposed to differentiate synthetic voice audio from real audio, aiming for faster inference and lower computational cost without sacrificing performance.
      4. **Fake Span Discovery via Attention** [5]**:** This technique introduces a framework for locating specific fake regions within the audio input. A self-attention mechanism is integrated into the model to enhance its ability to pinpoint and analyze manipulated segments.
      5. **Replay Attack Defense** [6]**:** This work addresses replay attacks, where an attacker replays recorded voices using various playback and recording devices in different environments. The authors propose an end-to-end deep neural network (DNN) to handle such attacks without relying on explicit knowledge-based features.
      6. **Res2Net and Squeeze-and-Excitation Blocks** [7]**:** Another approach combines Res2Net blocks with squeeze-and-excitation mechanisms to improve the model’s ability to detect replay and speech synthesis-based DeepFakes.

# CHAPTER 2 - DATASET

1. **Real Audio Dataset LJSpeech** [8] **-** To source real audio data, we utilize the LJSpeech [8]dataset, which contains 13,100 high-quality short audio clips (ranging from 1 to 10 seconds, totaling 24 hours) of a single female speaker reading passages from seven non-fiction books. The dataset is widely used for tasks involving speech synthesis and speaker recognition due to its clarity and consistency.
2. **DeepFake Audio Dataset (WaveFake** [9]**)-** For synthetic audio, we employ the WaveFake [9]dataset, which is based on LJSpeech and was introduced at NeurIPS 2021. It comprises 117,985 synthetic audio clips, totaling approximately 196 hours of generated speech. The dataset features fake audio produced by six state-of-the-art generative models:
   1. MelGAN [10]
   2. Parallel WaveGAN (PWG) [11]
   3. Multi-band MelGAN (MB-MelGAN) [9]
   4. Full-band MelGAN (FB-MelGAN) [9]
   5. HiFi-GAN [12]
   6. WaveGlow [13]

Each model generates 13,100 synthetic audio clips that correspond directly to the real audio clips in LJSpeech, making it a paired dataset ideal for supervised learning**.**

# CHAPTER 3 – METHODS

# 3.1 Audio Data Preprocessing and Processing Pipeline

# 3.1.1. Audio Signal Preprocessing

# Before applying any detection algorithms, raw audio signals undergo

# several preprocessing steps to ensure consistency and optimal feature extraction. The preprocessing pipeline consists of the following stages:

# Normalization and Standardization: All audio files are first normalized to a consistent amplitude range [-1, 1] to eliminate volume variations that could bias the detection models. This is achieved through peak normalization, where each audio sample is divided by the maximum absolute amplitude in the signal.

# Resampling: To ensure uniform sampling rates across all audio samples, we resample all audio files to a standard sampling rate of 16 kHz. This rate provides sufficient frequency resolution for speech analysis while maintaining computational efficiency. The resampling process uses anti-aliasing filters to prevent frequency domain artifacts.

1. **Windowing and Framing**: The continuous audio signal is segmented into overlapping frames using a Hamming window function. Each frame typically spans 25 milliseconds with a 10-millisecond hop length (60% overlap). This windowing approach balances temporal resolution with frequency domain stability, crucial for subsequent spectral analysis.
2. **Pre-emphasis**: A high-pass pre-emphasis filter with coefficient α = 0.97 is applied to balance the frequency spectrum and compensate for the natural spectral roll-off in speech signals:

s'[n] = s[n] - α × s[n-1] (1)

* + 1. **Silence Detection and Voice Activity Detection (VAD)**

To focus analysis on speech-relevant portions and reduce computational overhead, we implement a Voice Activity Detection system:

1. **Energy-Based VAD**: Frames with energy below a dynamic threshold (calculated as μ - 2σ of the signal's energy distribution) are classified as silence and excluded from feature extraction.
2. **Spectral Centroid Analysis:** Additional validation using spectral centroid shifts helps distinguish between speech and non-speech segments, improving the robustness of voice activity detection.
   * 1. **Audio Augmentation Techniques**

To improve model generalization and robustness against various audio conditions, we apply several data augmentation techniques during training:

1. **Additive Noise**: Gaussian white noise with SNR ratios ranging from 20-40 dB is added to simulate real-world recording conditions.
2. **Speed Perturbation**: Audio playback speed is varied by ±10% while maintaining pitch through time-stretch algorithms, simulating natural speech rate variations.
3. **Pitch Shifting**: Fundamental frequency is modified by ±2 semitones using phase vocoder techniques to account for speaker variability.
4. **Channel Simulation**: Different microphone and channel characteristics are simulated through convolution with room impulse responses and telephone channel filters.

**3.2 Feature Extraction Methodologies**

**3.2.1 Mel-Frequency Cepstral Coefficients (MFCC)**

MFCC features capture the spectral envelope characteristics crucial for speech analysis. Our implementation follows the standard procedure:

1. **Short-Time Fourier Transform (STFT):** Applied to windowed frames to obtain frequency domain representation
2. **Mel-scale Filtering**: 40 triangular filters spaced according to the Mel scale are applied to emphasize perceptually important frequencies
3. **Logarithmic Compression:** Log operation applied to filter bank outputs to compress dynamic range
4. **Discrete Cosine Transform (DCT):** Applied to decorrelate features and reduce dimensionality to 13 coefficients
5. **Delta and Delta-Delta Features**: First and second derivatives computed to capture temporal dynamics

The final MFCC feature vector consists of 39 dimensions (13 static + 13 delta + 13 delta-delta coefficients).

**3.2.2 Linear Frequency Cepstral Coefficients (LFCC)**

Unlike MFCC's perceptual scaling, LFCC maintains linear frequency spacing, potentially capturing subtle artifacts in synthetic speech:

1. **Linear Filter Bank:** 40 linearly-spaced triangular filters applied across the frequency spectrum
2. **Logarithmic Compression and DCT:** Similar to MFCC processing
3. **Feature Dimensionality:** Maintains same 39-dimensional structure as MFCC

# Spectral and Prosodic Features

# Additional features are extracted to capture unique characteristics of synthetic speech:

# Spectral Features:

# Spectral centroid, rolloff, and flux

# Zero-crossing rate variations

# Harmonic-to-noise ratio (HNR)

# Spectral flatness measure

# Prosodic Features:

# Fundamental frequency (F0) contour statistics

# Jitter and shimmer measurements

# Pause duration and rhythm patterns

# Energy distribution across frequency bands

# 3.3 Classification Approaches

# 3.3.1 Gaussian Mixture Model (GMM)

# We develop a GMM-based approach to deepfake audio detection. Specifically, our classifier is composed of 2 Gaussian Mixture Models that are trained individually on real and generated audio samples. Each model is composed of 128 single Gaussian models. The Gaussian distribution is chosen as the probability distribution for the mixture model, given its unique mathematical properties and computational efficiency.

# The two models use the MFCC features extracted from audio files as inputs. The final classification is made according to the likelihood function:

# f(x) = log p(x|θr) − log p(x|θg) (2)

# where θr and θg are the Gaussian parameters for the real and generated audio distributions respectively, with x being the input MFCC feature.

# Model Training: Each GMM is trained using the Expectation-Maximization (EM) algorithm with diagonal covariance matrices to reduce computational complexity while maintaining discriminative power.

# 3.3.2 Vanilla Recurrent Neural Network

# Audio is by nature sequential data, where the waveform manifests different frequencies and amplitudes across timestamps. Vanilla Recurrent Neural Network (RNN) is the simplest architecture for processing sequential data. In our implementation, a hidden state tracks all historical information from previous timestamps. Given an input at timestamp t, the current hidden state is updated by the previous hidden state and current input through fully-connected layers and an activation function. The hidden state at the last timestamp represents the entire audio, which is fed to two fully connected layers for final classification.

*ht* = *tanh*(*Wiit* + *Whht*−1) *(3)*

*ot* = *Wo*2*relu*(*Wo*1*hlast*)*, (4)*

# where *Wi*, *Wh*, *Wo*2, *Wo*1 refer to parameters for the input layer, hidden state transition, and two fully connected output layers respectively.

# 3.3.3 Bi-directional Long Short Term Memory

# While current works including [6] employ Gated Recurrent Unit (GRU), we investigate Long Short-Term Memory (LSTM) performance compared to GRU and vanilla RNN. Since this task relies on temporal information where later information isn't necessarily more important than earlier information, we design a double-layered bidirectional LSTM followed by a 1-dimensional convolution layer and fully-connected layer. The model takes preprocessed LFCC or MFCC features shaped (B, N, T) where B is batch size, N is the number of cepstral coefficients, and T is the number of temporal frames.

# Bidirectional Processing: Forward and backward LSTM layers capture both past and future context, crucial for detecting temporal inconsistencies in synthetic speech.

# 3.3.4 Shallow Convolutional Neural Network

# We adopt a shallow CNN approach inspired by Lieto et al. [4], utilizing CNN applied to 2D-image representation of audio signals. While the original paper used Mel-spectrogram representation, we use MFCC and LFCC features for consistency with other models. MFCC features are obtained through logarithmic scaling of Mel-spectrogram followed by DCT computation. LFCC is similar but derived from DB-scaled linear filtered spectrogram.

# Architecture Details:

# Input: 2D feature maps (frequency × time) derived from MFCC/LFCC

# Convolutional layers with varying filter sizes to capture local patterns

# Max-pooling for translation invariance

# Global average pooling before classification layers

# 

# Rationale for Hand-crafted Features: While neural networks can learn features automatically, MFCC and LFCC incorporate decades of speech processing knowledge. These features emphasize perceptually important characteristics and provide a strong baseline, especially with limited training data.

# 3.3.5 Time Domain Synthetic Speech Detection (TSSD)

# Hua et al. [14] proposed Time-Domain Synthetic Speech Detection (TSSD), an end-to-end framework using deep neural networks for both feature extraction and classification. This lightweight network processes raw audio waveforms without pre-transforms or hand-crafted features.

# The architecture consists of:

# Initial Convolution: 1×7 1D convolutional layer with 16 channels, followed by Batch Normalization, ReLU activation, and max-pooling (kernel size 4)

# ResNet Blocks: M=4 stacked ResNet-style modules with max-pool layers. Each module contains 3 1×3 convolution layers with skip connections, BN, and ReLU. Channel progression: 16→32→64→128

# Classification Head: 3 fully connected layers (128→64→32 dimensions) followed by softmax output.

# 

# Fig 2 TSSD Architecture

# CHAPTER 4 – EXPERIMENTAL SETUPS

# 4.1 Training and Testing Setup

Training and Testing Setup. We choose to conduct experiments using the two most challenging experiment setups described in WaveFake [9].

* **In-distribution Setup**: We used 80% MelGAN + 80% LJSpeech for training, and 20% MelGAN + 20% LJSpeech for testing. i.e., the Real-to-Fake ratio in training is 1:1.
* **Out-of-distribution Setup:** We used everything except MelGAN + 80% LJSpeech for training, and 20% MelGAN + 20% LJSpeech for testing. i.e., the Real-to-Fake ratio in training is roughly 7.4:1.

# Loss Function

We  used the **Binary Cross Entropy Loss** as the training loss for this binary classification problem. In particular we used the  **torch.nn.BCEWithLogitsLoss** function from PyTorch [15], which will combine the Sigmoid function with the BCELoss in a single layer to ensure better numerical stability. When the model is trained in the out-of-distribution setting, we will calculate the ratio of the number of positive samples to the number of negative samples, i.e., pos\_weight, and use this as the additional weight for the loss. By doing so, the loss will act as if there are an equal number of positive and negative samples.

|  |  |
| --- | --- |
| *ln* = −*wn* [*yn* · log*σ*(*xn*) + (1 − *yn*) · log(1 − *σ*(*xn*))] | (5) |
| *ℓ*(*x,y*) = *mean*(*L*) = *mean*({*l*1*,...,lN*}⊤) | (6) |

**4.3 Hyper-parameters**

For the experiments,our models are trained on a NVIDIA RTX3090 graphics processing unit having a batch size of 256. We used the **Adam** [16] **optimizer** and initial **learning rate was 0.0005** and also **weight decay of 0.0001**.

# CHAPTER 5 – RESULTS AND CONCLUSION

**5.1 Results**

In Table 1, we follow WaveFake [10] to report the Equal Error Rate (EER) (a.k.a. Crossover Error Rate), which is commonly used to measure the overall accuracy of a biometric system. EER is defined as the common value where the False Positive Rate (FPR) equals the False Negative Rate (FNR). Lower EER values indicate better performance. Additionally, in Table 2, we also report the F1 score (a.k.a. balanced F-score) and the Area Under the Receiver Operating Characteristic Curve (ROC AUC). Higher F1 score and larger AUC indicate better performance.

Table 1 : Experimental Results reported in EER

|  |  |
| --- | --- |
| Model (input feature type) | EER Score |
| GMM (w/LFCC) | 0.0148 |
| RawNet2 (w/wave) | 0.001 |
| VanillaRNN (w/wave) | 0.350 |
| Bi-LSTM (w/wave) | 0.264 |
| Bi-LSTM (w/LFCC) | 0.040 |
| Bi-LSTM (w/MFCC) | 0.004 |
| ShallowCNN (w/MFCC) | 0.004 |
| ShallowCNN (w/LFCC) | 0.001 |
| TSSD (w/wave) | 0.001 |

GMM (w/ LFCC) and RawNet2 (w/ wave) results are provided by WaveFake[10]. We round all results to three decimal places following [10], hence, 0.000 implies the actual value is less than 0.0005

Table 1 : Experimental Results reported in F1 Score and AUC

|  |  |  |  |
| --- | --- | --- | --- |
| Model (input feature type) | Accuracy | F1 Score | ROC AUC |
| VanillaRNN (w/wave) | 65.26 | 0.649 | 0.653 |
| Bi-LSTM (w/wave) | 75.36 | 0.742 | 0.750 |
| Bi-LSTM (w/LFCC) | 96.50 | 0.960 | 0.960 |
| Bi-LSTM (w/MFCC) | 96.01 | 0.996 | 0.996 |
| ShallowCNN (w/MFCC) | 97.56 | 0.997 | 0.997 |
| ShallowCNN (w/LFCC) | 92.96 | 0.999 | 0.999 |
| TSSD (w/wave) | 95.61 | 0.999 | 0.999 |

GMM (w/ LFCC) and RawNet2 (w/ wave)’s F1 and AUC results are not reported by WaveFake [10]. We round all results to three decimal places, hence, 1.000 implies the actual value is larger than or equal to 0.9995

**5.2 Analysis**

We anticipated that models used in the out-of-distribution setup would require considerably more time for training — and our observations confirmed this. However, when it came to performance, models trained under both in-distribution and out-of-distribution settings exhibited comparable results.

Interestingly, two specific data points, LJ048-0107 and LJ050-0267, stood out as misclassified by several models, making them suitable candidates for deeper analysis. In particular:

* **LJ048-0107** was misclassified :  
  SimpleLSTM\_mfcc, TSSD\_wave, WaveLSTM\_wave, and WaveRNN\_wave
* **LJ050-0267** was misclassified by:  
  ShallowCNN\_mfcc, SimpleLSTM\_mfcc, WaveLSTM\_wave, and WaveRNN.

For a more detailed and interactive examination of these cases, refer to the following page:  
[**https://deluxe-nougat-e21b63.netlify.app/**](https://deluxe-nougat-e21b63.netlify.app/)

As visualized through the interactive GUI, the features extracted from LJ048-0107 and LJ050-0267 are highly alike. Their main structural components are nearly identical, with only subtle variations in finer details, which likely contributed to the consistent misclassifications across models.

**5.3 Conclusion**

In this project, we designed, evaluated, and compared multiple deep learning models aimed at distinguishing between human and synthetic (bot) speech using the WaveFake and LJSpeech datasets as standardized benchmarks.

Our findings indicate that although the networks examined in this study perform reasonably well — even when challenged with high-quality audio generated by advanced GANs — there are still specific audio inputs that can deceive these models. These vulnerabilities, highlighted in our qualitative analysis, suggest that adversaries can strategically design such inputs to bypass detection.

Therefore, there's a clear need for further enhancement to improve the robustness and reliability of these systems. The insights gained from this work may prove valuable in addressing future challenges related to privacy and security brought about by automated speech synthesis and deepfake voice technologies. We hope that our results will motivate continued exploration in the field of deepfake audio detection.

# Bibliography

|  |  |
| --- | --- |
| [1] | Unknown, "Youtube," Youtube, 2022. [Online]. Available: https://www.youtube.com/watch?v=X17yrEV5sl4. [Accessed 15 03 2025]. |
| [2] | N. Subramani and D. Rao, "Learning Efficient Representations for Fake Speech Detection," *Proceedings of the AAAI Conference on Artificial Intelligence,* vol. 34, no. 04, pp. 5859-5866, 2020. |
| [3] | R. Wang, F. Juefei-Xu, Y. Huang, Q. Guo, X. Xie, L. Ma and Y. Liu, "DeepSonar: Towards Effective and Robust Detection of AI-Synthesized Fake Voices," 2020. |
| [4] | A. Lieto, D. Moro, F. Devoti, C. Parera, V. Lipari, P. Bestagini and S. Tubaro, ""Hello? Who Am I Talking to?" A Shallow CNN Approach for Human vs. Bot Speech Classification," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019. |
| [5] | H. Wu, H.-C. Kuo, N. Zheng, K.-H. Hung, H.-Y. Lee, Y. Tsao, H.-M. Wang and H. Meng, "Partially Fake Audio Detection by Self-attention-based Fake Span Discovery," 2022. |
| [6] | H.-j. S. H.-S. H. H.-J. Y. Jee-weon Jung, "Replay attack detection with complementary high-resolution information using end-to-end DNN for the ASVspoof 2019 Challenge," in *ASVspoof 2019 Challenge*, 2019. |
| [7] | N. L. C. W. X. L. D. S. D. Y. H. M. Xu Li, "Replay and synthetic speech detection with res2net architecture," 2020. |
| [8] | K. Ito and L. Johnson, "The LJ Speech Dataset," 2017. [Online]. Available: https://keithito.com/LJ-Speech-Dataset/. [Accessed 18 03 2025]. |
| [9] | L. S. Joel Frank, "A Data Set to Facilitate Audio Deepfake Detection," in *NeurIPS*. |
| [10] | R. K. T. d. B. L. G. W. Z. T. J. S. A. d. B. Y. B. A. C. C. Kundan Kumar, "MelGAN: Generative adversarial networks for conditional waveform synthesis," in *NuerIPS*, 2019. |
| [11] | S. D. H. Z. K. S. O. V. A. G. N. K. A. S. K. K. Aaron van den Oord, "WaveNet: A Generative Model for Raw Audio," in *DeepMind*, 2016. |
| [12] | J. K. a. J. B. Jungil Kong, "Hifi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems,* vol. 33, p. 17022–17033, 2020. |
| [13] | P. D. Durk P. Kingma, "Generative flow with invertible 1x1 convolutions," in *NeurIPS*, 2018. |
| [14] | A. B. J. T. H. Z. Guang Hua, "Towards End-to-End Synthetic Speech Detection," *IEEE Signal Processing Letters,* vol. 28, pp. 1265-1269, 2021. |
| [15] | S. G. F. M. A. L. J. B. G. C. T. K. Z. L. N. G. L. A. A. D. A. K. E. Y. Z. D. M. R. A. T. S. C. Adam Paszke, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Curran Associates, Inc.*, 2019. |
| [16] | J. B. Diederik P. Kingma, "Adam: A Method for Stochastic Optimization," in *International Conference for Learning Representations (ICLR)*, 2015. |