

# **UNMASKING DEEP FAKE AUDIO USING DEEP LEARNING**

## **A PROJECT REPORT**

*Submitted by,*

**KOPPALA REDDY ANUSHA      (723921104021)**

**PEDDANAGAGARICHARITHA    (723921104036)**

**VEMULA RAKSHITHA            (723921104052)**

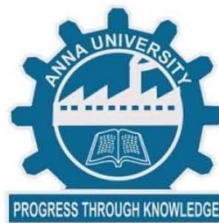
*In partial fulfillment for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

*In*

**COMPUTER SCIENCE AND ENGINEERING**



**ARJUN COLLEGE OF TECHNOLOGY**

**COIMBATORE – 642 120**

**ANNA UNIVERSITY: CHENNAI 600 025**

**MAY 2025**

**ANNA UNIVERSITY: CHENNAI 600 025**

**BONAFIDE CERTIFICATE**

Certified that this Report titled **“UNMASKING DEEP FAKE AUDIO USING DEEP LEARNING”** is the Bonafide work of **KOPPALA REDDY ANUSHA (723921104021), PEDDANAGAGARI CHARITHA (723921104036), VEMULA RAKSHITHA (723921104052)**, who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported here in does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**SIGNATURE**

**Mr. S. SATHEESH M.E., (PHD)**

**HEAD OF THE DEPARTMENT,**

Associate Professor,

Department of CSE,

Arjun College of Technology,

Coimbatore -642 120

**SIGNATURE**

**Ms. R. LATHA PRIYADHARSHINI M.E.,**

**SUPERVISOR,**

Assistant Professor,

Department of CSE,

Arjun College of Technology,

Coimbatore -642 120

Submitted for the university project viva-voice held on \_\_\_\_\_

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## ACKNOWLEDGEMENT

We take this opportunity to express our heartfelt gratitude to all those who have supported and guided us throughout the successful completion of our project.

First and foremost, we would like to express our sincere thanks to our respected **Chairman, Thiru R. Suriyanarayanan**, our **Managing Trustee, Thiru Er. G. Srinivasan**, our **Trustee, Thiru Raja Duraisamy**, and our **Secretary, Dr. R. Suresh Kumar** for providing us with the opportunity and infrastructure to carry out this project.

We extend our deep gratitude to our beloved **Principal, Dr. N. Janaki Manohar**, for his constant encouragement and support throughout our academic journey.

We are thankful to **Mr. S. Satheesh**, Head of the Department of Computer Science and Engineering, for his valuable support and motivation.

We would also like to extend our appreciation to our **Project Coordinator, Mr. V. M. Suresh**, for his continuous monitoring and encouragement.

A special note of thanks goes to our **Project Guide, Ms. R. Latha Priyadharshini**, for her invaluable guidance, patience, and support at every stage of this project. Her expertise and insights have been instrumental in its successful completion.

Finally, we sincerely thank all the faculty members, technical staff, and our friends of **Arjun College of Technology** who directly or indirectly contributed to the success of our project.

## ABSTRACT

This project delves into the challenging domain of synthetic speech detection, employing a sophisticated analysis framework that integrates short-term and long-term prediction traces. Leveraging cutting-edge deep learning methodologies such as Convolutional Neural Networks (CNN), Wave Net, an undisclosed model denoted as ISTM, and Recurrent Neural Networks (RNN), our methodology endeavors to establish a robust mechanism for identifying synthetic speech across diverse contexts. Central to our approach is the extraction of a comprehensive array of features, encompassing both short-term attributes like Zero Crossing Rate and Spectral Control, and long-term characteristics such as Mel-Frequency Cepstral Coefficients (MFCC) and Chroma features. Through meticulous data preprocessing, model training, and rigorous evaluation, our aim is to construct a system that exhibits high accuracy in discerning synthetic speech instances. This endeavor not only contributes significantly to the advancement of speech processing techniques but also holds promise for real-world applications in fraud detection, voice authentication, and content verification. By addressing the burgeoning challenge of synthetic speech detection, our project endeavors to pave the way for enhanced security measures and trustworthiness in voice-based systems and applications.

## TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	<b>ABSTRACT</b>	<b>iv</b>
	<b>LIST OF TABLES</b>	<b>viii</b>
	<b>LIST OF FIGURES</b>	<b>ix</b>
	<b>LIST OF ABBREVIATIONS</b>	<b>x</b>
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 Problem Statement	2
	1.2 Significant of Study	2
	1.3 Objective of the Project	3
	1.4 Scope	3
<b>2</b>	<b>LITERATURE SURVEY</b>	<b>5</b>
<b>3</b>	<b>SYSTEM ANALYSIS</b>	<b>13</b>
	3.1 Existing System	13
	3.1.1 Disadvantages	13
	3.2 Proposed System	14
	3.2.1 Advantages	15
	3.3 System Requirements	17
	3.3.1 Functional Requirements	17
	3.3.2 Non-Functional Requirements	18
<b>4</b>	<b>SYSTEM REQUIREMENTS</b>	<b>19</b>
	4.1 System Requirements	19
	4.2 Hardware Requirements	19
	4.3. Software Requirements	19

	4.4 Functional Requirements	20
	4.5 Non-Functional Requirements	21
	4.5.1 Usability	21
	4.5.2 Reliability	21
	4.5.3 Performance	21
	4.5.4 Supportability	22
	4.5.5 Data Set Requirements	22
<b>5</b>	<b>SYSTEM DESIGN</b>	<b>23</b>
	5.1 Implementation of Key Function	23
	5.1.1 Data Preprocessing	23
	5.1.2 Feature Extraction	23
	5.2 User Guide	24
	5.3 System Testing	24
	5.4 System Documentation	25
	5.5 Program Documentation	25
	5.6 Technical Details	25
<b>6</b>	<b>SYSTEM STUDY AND TESTING</b>	<b>26</b>
	6.1 Feasibility Study	26
	6.2 Types of Testing	28
	6.2.1 Unit Testing	28
	6.2.2 Integration Testing	28
	6.2.3 Functional Testing	29
	6.2.4 White Box Testing	30
	6.2.5 Black Box Testing	30
<b>7</b>	<b>SYSTEM IMPLEMENTATION</b>	<b>32</b>
	7.1 Audio Input and Preprocessing	32

	7.2 Feature Extraction	32
	7.3 Model Training and Classification	33
	7.4 Deployment	33
	7.5 Logging and Reporting	33
<b>8</b>	<b>CONCLUSION AND FUTURE ENHANCEMENT</b>	<b>34</b>
	8.1 Conclusion	34
	8.2 Future Enhancement	35
<b>9</b>	<b>APPENDICES</b>	<b>36</b>
	9.1 Source Code	36
	9.2 Screenshots	41
	<b>REFERENCE</b>	<b>45</b>

**LIST OF FIGURES**

<b>FIGURE NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
<b>3.1</b>	Flow Diagram	16
<b>9.1</b>	Home Page	41
<b>9.2</b>	About Page	41
<b>9.3</b>	Login Page	42
<b>9.4</b>	Registration Page	42
<b>9.5</b>	Upload Page	43
<b>9.6</b>	Prediction Page (Fake)	43
<b>9.7</b>	Prediction Page (Real)	44



## LIST OF TABLES

<b>TABLE NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
<b>6.1</b>	Test Cases	31
<b>6.2</b>	Test Case Model Building	31

## LIST OF ABBREVIATIONS

MFCC	Mel-Frequency Cepstral Coefficients
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Networks
GANs	Generative Adversarial Networks
ASDG	Separation Domain Generalization
UDFA	Unmasking Deep Fake Audio
LCNN	Lightweight Convolutional Neural Networks
SSD	Synthetic Speech Detection
FD	First Digit
INRs	Implicit Neural Representations
ASV	Automatic speaker verification
CQCCs	Constant Q Cepstral Coefficients
APGDF	All-pole group delay function
FFV	Fundamental Frequency Variation
GMM	Gaussian mixture model
DNN	Deep Neural Network

## CHAPTER 1

### INTRODUCTION

In a world increasingly driven by artificial intelligence and deep learning, the ability to discern between genuine human speech and synthetic speech has become a critical challenge. Synthetic speech, generated by sophisticated algorithms, poses risks ranging from misinformation dissemination to fraudulent activities. Our project, "Synthetic Speech Detection through Short-Term and Long-Term Prediction Traces," addresses this pressing issue by leveraging cutting-edge deep learning algorithms and advanced feature extraction techniques.

Our approach involves the extraction of both short-term and long-term features from audio data. Short-term features, such as Zero Crossing Rate and Spectral Control, capture instantaneous characteristics of the audio signal, while long-term features, including Mel-Frequency Cepstral Coefficients (MFCC) and Chroma features, provide insights into broader patterns and structures within the speech signal. Evaluation metrics such as accuracy, precision, recall, and F1-score are employed to assess the system's performance across different datasets and real-world scenarios. Additionally, mechanisms for continual learning and adaptation to evolving synthetic speech generation techniques are incorporated to ensure the system's long-term effectiveness.

Ultimately, the developed system holds the potential to significantly enhance security, trust, and authenticity in communication channels by mitigating the risks associated with synthetic speech. Through this project, we aim to contribute to the advancement of technology in safeguarding against the misuse of synthetic speech while fostering a more secure and trustworthy digital environment.

## **1.1 PROBLEM STATEMENT**

The proliferation of synthetic speech technology poses a significant challenge in accurately discerning between authentic human speech and artificially generated content. Our project aims to address this challenge by developing a robust detection system capable of reliably identifying synthetic speech across diverse contexts and applications.

The sophistication of synthetic speech, coupled with the increasing prevalence of malicious activities such as fraud and misinformation dissemination, underscores the urgency of this endeavor. By leveraging advanced deep learning algorithms and feature extraction techniques, our objective is to build a system that not only achieves high accuracy in detecting synthetic speech but also demonstrates resilience against evolving synthetic speech generation techniques. Through meticulous data collection, preprocessing, model training, and evaluation, our project seeks to deliver a solution that enhances security, trust, and authenticity in communication channels, thereby mitigating the risks associated with synthetic speech technology.

## **1.2 SIGNIFICANT OF STUDY**

The behind of this project stems from the increasing prevalence of synthetic speech in various contexts, including but not limited to automated customer service, virtual assistants, and deepfake technology. As synthetic speech becomes more sophisticated, it poses challenges in distinguishing it from genuine human speech, leading to potential misuse, misinformation, and fraud. By developing a robust system for synthetic speech detection, we aim to address these challenges and contribute to maintaining trust and integrity in communication channels. Detecting synthetic speech accurately can aid in safeguarding against fraudulent activities, protecting individuals

from misinformation, and ensuring the authenticity of content shared in various mediums.

### **1.3 OBJECTIVE OF THE PROJECT**

The objective of project is to develop a robust system for detecting synthetic speech by analyzing short-term and long-term prediction traces. We aim to achieve this by exploring advanced deep learning algorithms such as Convolutional Neural Networks (CNN), Wave Net, an undisclosed model abbreviated as ISTM, and Recurrent Neural Networks (RNN). Our focus will be on extracting pertinent features from audio data, including short-term features like Zero Crossing Rate and Spectral Control, as well as long-term features such as Mel-Frequency Cepstral Coefficients (MFCC) and Chroma features. Through rigorous model training and optimization, we strive to create a system that delivers high accuracy and reliability in identifying synthetic speech instances. Evaluation metrics including accuracy, precision, recall, and F1-score will be employed to assess the system's performance across various datasets and real-world scenarios. Additionally, we aim to make our system adaptable to evolving synthetic speech generation techniques through continual learning mechanisms. Ultimately, our goal is to deploy the developed system in practical applications such as fraud detection, voice authentication, and content verification, contributing to enhanced security, trust, and authenticity in communication channels.

### **1.4 SCOPE**

The scope of our project revolves around the development of a robust system for detecting synthetic speech through the analysis of short-term and long-term prediction traces. This entails exploring and implementing advanced deep learning algorithms such as Convolutional Neural Networks (CNN), Wave Net, an undisclosed model abbreviated as ISTM, and Recurrent

Neural Networks (RNN). We will focus on extracting relevant features from audio data, including short-term features like Zero Crossing Rate and Spectral Control, as well as long-term features like Mel-Frequency Cepstral Coefficients (MFCC) and Chroma features. Our efforts will involve collecting diverse datasets, preprocessing the data, and training models to achieve high accuracy in identifying synthetic speech instances. Evaluation metrics such as accuracy, precision, recall, and F1-score will be used to assess the system's performance across different datasets and scenarios. Additionally, we aim to ensure the system's adaptability to evolving synthetic speech generation techniques and explore potential applications in domains such as fraud detection, voice authentication, and content verification.

## CHAPTER 2

### LITERATURE SURVEY

**TITLE:** Domain Generalization via Aggregation and Separation for Audio Deepfake Detection

**AUTHOR:** Yuankun Xie, Haonan Cheng, Yutian Wang, and Long Ye

**YEAR:** 2024

**DESCRIPTION:** Authors introduce the Aggregation and Separation Domain Generalization (ASDG) method for unmasking deep fake audio (UDFA). The goal is to achieve better generalizability in detecting unseen target domains of deepfake speech. The proposed method includes a feature generator based on Lightweight Convolutional Neural Networks (LCNN) to categorize features into real and fake speech, single-side domain adversarial learning to make real speech indistinguishable from different domains, and a triplet loss to separate the distribution of fake speech while aggregating the distribution of real speech. The paper presents extensive experiments, including training with three different English datasets and evaluation in harsh conditions such as cross language and noisy datasets. To 39.24% when compared to that of RawNet2, demonstrating the generalizability of the model for unknown target domains.

**TITLE:** Audio Deepfake Detection with self-supervised Wav LM and Multi-Fusion Attentive Classifier

**AUTHOR:** Yinlin Guo, Haofan Huang, Xi Chen, He Zhao, Yeuhai Wang

**YEAR:** 2024

**DESCRIPTION:** Authors introduce a novel audio deepfake detection method employing the Wav LM self-supervised model and Multi-Fusion Attentive (MFA) Classifier. Training on ASVs pool 2019 LA data and

evaluating on ASVs poof 2019 LA, ASVs poof 2021 LA, and ASVs poof 2021 DF datasets, the approach achieves state-of-the-art results on ASVs poof 2021 DF and competitive performance on ASVs poof 2019 and 2021 LA datasets. However, limitations include dataset diversity, generalization to various attacks, narrow evaluation metrics, insufficient analysis of failures, undiscussed preprocessing rationale, high computational resource dependency, and external dependency risks. Despite these limitations, 6 the proposed method outperforms systems using self-supervised models, highlighting the efficiency of Wav LM in audio deepfake detection.

**TITLE:** Detecting Fake Audio of Arabic Speakers Using Self-Supervised Deep Learning

**AUTHOR:** Zaynab M. Almutairi And Hebah Elgibreen

**YEAR:** 2023

**DESCRIPTION:** Authors discussed that audio Deepfake is a significant topic in forensics. AI generated tools clone people's voices. Attackers misuse it, endangering public safety. Machine Learning and Deep Learning methods detect fake voices. However, these methods require extensive data and pre-processing. No previous exploration of synthetic fake audio in Arabic speech. The fakeness in Arabic speech is limited to imitation. This paper introduces a new method for detecting Audio Deepfakes called Arabic-AD. It uses self-supervised learning techniques to detect synthetic and imitated voices. The paper also creates a synthetic dataset of a single speaker who speaks Modern Standard Arabic perfectly. The accent is taken into account by collecting Arabic recordings from non-Arabic speakers. Three experiments are conducted to compare the proposed method with well-known benchmarks.



**TITLE:** Learning A Self-Supervised Domain Invariant Feature Representation for Generalized Audio Deepfake Detection

**AUTHOR:** Yuankun Xie, Haonan Cheng, Yutian Wang, Long Ye

**YEAR:** 2023

**DESCRIPTION:** Authors present W2V-ASDG, a robust Audio Deepfake Detection (ADD) model addressing cross-domain limitations. Combining a self-supervised front-end (W2V2-XLS-R) and a domain generalization backbone (ASDG), the system learns a domain-invariant feature representation for real and fake speech. Utilizing diverse datasets for training and evaluation, including ASVspoof2019LA, Wave Fake, Fake AV Celeb, IWA, ASVspoof2021DF, and FAD, the model achieves an impressive 4.60% Equal Error Rate (EER). However, limitations arise from a primarily English-focused training dataset, potentially impacting generalization to other languages and emerging attack types.

**TITLE:** Audio Deepfake Detection: A Survey

**AUTHOR:** Jiangyan, Cheglong Wang, jianhua Tao, Chu Yuan Zhang & Yan Zhao

**YEAR:** 2023

**DESCRIPTION:** Author provides a comprehensive survey on audio deepfake detection, highlighting key differences across various types of deepfake audio and discussing competitions, datasets, features, classifications, and evaluation of state-of-the-art approaches. The survey includes a detailed summary of up-to-date audio deepfake detection datasets and performs a unified comparison of representative detection methods. It is observed that while the performances may seem to be decreasing, it is

likely due to the increasing difficulty of the tasks rather than a decrease in performance.

**TITLE:** “Synthetic Speech Detection Algorithms”, UNIVERSITY OF PADOVA

**AUTHOR:** Federica LATORA

**YEAR:** 2022

**DESCRIPTION:** The recent diffusion of audio recording devices together with the rapid evolution of deepfake technologies have fostered the widespread of synthetic speech signals. Being extremely convincing and realistic can be used in many malicious applications, e.g., for fake news spreading over social media platforms, frauds or specifically in impersonation attacks, since speech signals are needed to unlock or control many devices. As a matter of fact, the development of efficient detection algorithms that verify the authenticity of audio recordings and help human listeners in discriminating fraudulent audio samples from real ones is therefore of paramount importance. Synthetic Speech Detection (SSD) algorithms are systems that estimate whether a speech signal under analysis has been synthetically created or has been authentically acquired by an audio recorder. However, this problem is getting challenging due to the constant development of new technologies and methods brought by deep learning for fake speech generation. For this reason, the study of new detection strategies is becoming increasingly urgent and necessary. In this thesis, some algorithms for the SSD task are proposed. The first approach uses the First Digit (FD) statistics computed on signal transform coefficients to detect peculiar characteristics of fake audio signals. The second method instead adopts Implicit Neural Representations (INRs) of speech signals, which are obtained with neural networks overfitted on each signal, to distinguish fake

samples from bonafide ones. In both cases, it has been pointed out the fundamental role of silenced parts in synthetic speech detection. However, this thesis represents only a preliminary analysis, which we hope will help widening the perspectives of audio forensic research.

**TITLE:** “Open Challenges in Synthetic Speech Detection”, Fraunhofer Institute for Digital Media Technology IDMT

**AUTHOR:** Luca Cuccovillo

**YEAR:** 2022

**DESCRIPTION:** In this paper the current status and open challenges of synthetic speech detection are addressed. The work comprises an initial analysis of available open datasets and of existing detection methods, a description of the requirements for new research datasets compliant with regulations and better representing real-case scenarios, and a discussion of the desired characteristics of future trustworthy detection methods in terms of both functional and non-functional requirements. Compared to other works, based on specific detection solutions or presenting single dataset of synthetic speeches, our paper is meant to orient future state-of-the-art research in the domain, to quickly lessen the current gap between synthesis and detection approaches.

**TITLE:** “Synthetic speech detection through short-term and long-term prediction traces “, EURASIP Journal on Information Security volume

**AUTHOR:** Clara Borrelli, Paolo Bestagini, Fabio Antonacci, Augusto Sarti & Stefano Tubaro

**YEAR:** 2021

**DESCRIPTION:** Several methods for synthetic audio speech generation have been developed in the literature through the years. With the great

technological advances brought by deep learning, many novel synthetic speech techniques achieving incredible realistic results have been recently proposed. As these methods generate convincing fake human voices, they can be used in a malicious way to negatively impact on today's society (e.g., people impersonation, fake news spreading, opinion formation). For this reason, the ability of detecting whether a speech recording is synthetic or pristine is becoming an urgent necessity. In this work, we develop a synthetic speech detector. This takes as input an audio recording, extracts a series of hand-crafted features motivated by the speech-processing literature, and classify them in either closed-set or open-set. The proposed detector is validated on a publicly available dataset consisting of 17 synthetic speech generation algorithms ranging from old fashioned vocoders to modern deep learning solutions. Results show that the proposed method outperforms recently proposed detectors in the forensics literature.

**TITLE:** "Synthetic speech detection using fundamental frequency variation and spectral features"

**AUTHOR:** Monisankha Pal, Dipjyoti Paul, Goutam Saha

**YEAR:** 2018

**DESCRIPTION:** Recent works on the vulnerability of automatic speaker verification (ASV) systems confirm that malicious spoofing attacks using synthetic speech can provoke significant increase in false acceptance rate. A reliable detection of synthetic speech is key to develop countermeasure for synthetic speech based spoofing attacks. In this paper, we targeted that by focusing on three major types of artifacts related to magnitude, phase and pitch variation, which are introduced during the generation of synthetic speech. We proposed a new approach to detect synthetic speech using score-level fusion of front-end features namely, constant Q cepstral coefficients

(CQCCs), all-pole group delay function (APGDF) and fundamental frequency variation (FFV). CQCC and APGDF were individually used earlier for spoofing detection task and yielded the best performance among magnitude and phase spectrum related features, respectively. The novel FFV feature introduced in this paper to extract pitch variation at frame-level, provides complementary information to CQCC and APGDF. Experimental results show that the proposed approach produces the best stand-alone spoofing detection performance using Gaussian mixture model (GMM) based classifier on ASVs spoof 2015 evaluation dataset. An overall equal error rate of 0.05% with a relative performance improvement of 76.19% over the next best-reported results is obtained using the proposed method. In addition to outperforming all existing baseline features for both known and unknown attacks, the proposed feature combination yields superior performance for ASV system (GMM with universal background model/i-vector) integrated with countermeasure framework. Further, the proposed method is found to have relatively better generalization ability when either one or both of copy-synthesized data and limited spoofing data are available a priori in the training pool.

**TITLE:** “Statistical Parametric Speech Synthesis Incorporating Generative Adversarial Networks”

**AUTHOR:** Yuki Saito, Shinnosuke Takamichi, H. Saruwatari

**YEAR:** 2017

**DESCRIPTION:** A method for statistical parametric speech synthesis incorporating generative adversarial networks (GANs) is proposed. Although powerful deep neural networks (DNNs) techniques can be applied to artificially synthesize speech waveform, the synthetic speech quality is low compared with that of natural speech. One of the issues

causing the quality degradation is an over-smoothing effect often observed in the generated speech parameters. A GAN introduced in this paper consists of two neural networks: a discriminator to distinguish natural and generated samples, and a generator to deceive the discriminator. In the proposed framework incorporating the GANs, the discriminator is trained to distinguish natural and generated speech parameters, while the acoustic models are trained to minimize the weighted sum of the conventional minimum generation loss and an adversarial loss for deceiving the discriminator. Since the objective of the GANs is to minimize the divergence (i.e., distribution difference) between the natural and generated speech parameters, the proposed method effectively alleviates the over-smoothing effect on the generated speech parameters. We evaluated the effectiveness for text-to-speech and voice conversion, and found that the proposed method can generate more natural spectral parameters and F0 than conventional minimum generation error training algorithm regardless its hyper-parameter settings. Furthermore, we investigated the effect of the divergence of various GANs, and found that a Wasserstein GAN minimizing the Earth-Mover’s distance works the best in terms of improving synthetic speech quality.

## **CHAPTER 3**

### **SYSTEM ANALYSIS**

#### **3.1 EXISTING SYSTEM**

Prior to the development of our project, the detection of synthetic speech primarily relied on traditional signal processing techniques and rule-based methods. These approaches often lacked robustness and scalability, struggling to keep pace with the rapid advancements in synthetic speech technology. As a result, there was a growing need for more sophisticated and adaptive solutions capable of accurately discerning between synthetic and genuine speech.

Some existing systems utilize basic signal features such as pitch, energy, and formant frequencies to differentiate between synthetic and natural speech. However, these systems may lack the capability to effectively handle the nuances and complexities of modern synthetic speech generation techniques.

##### **3.1.1 DISADVANTAGES**

1. **Scalability Issues:** Many existing systems struggle to scale effectively to handle large and diverse datasets, limiting their applicability in real-world scenarios where the volume and variability of data may be significant.
2. **Limited Accuracy:** Traditional methods for synthetic speech detection often rely on simplistic features and rule-based approaches, leading to limited accuracy, especially when faced with increasingly sophisticated synthetic speech technologies.
3. **Lack of Adaptability:** Rule-based and traditional machine learning approaches may lack the adaptability to effectively detect new and

evolving synthetic speech generation techniques, making them susceptible to being circumvented by malicious actors.

4. **High False Positive Rates:** Some systems may exhibit high false positive rates, incorrectly flagging genuine human speech as synthetic, which can lead to inefficiencies and false alarms in practical applications.
5. **Complexity and Maintenance:** Systems based on complex rule sets or handcrafted features may be difficult to maintain and update, requiring manual intervention and expertise to keep them effective over time.
6. **Dependency on Specific Features:** Certain systems heavily rely on specific signal features for synthetic speech detection, making them vulnerable to evasion tactics that exploit weaknesses in feature representation.

### **3.2 PROPOSED SYSTEM**

Our proposed system represents a significant advancement in synthetic speech detection by leveraging state-of-the-art deep learning algorithms and comprehensive feature extraction techniques. By harnessing the capabilities of Convolutional Neural Networks (CNN), Wave Net, an undisclosed model identified as ISTM, and Recurrent Neural Networks (RNN), our system aims to achieve superior accuracy and robustness in identifying synthetic speech instances. Integral to our approach is the extraction of both short-term and long-term features from audio data, encompassing characteristics such as Zero Crossing Rate, Spectral Control, Mel-Frequency Cepstral Coefficients (MFCC), and Chroma features. Through rigorous model training, optimization, and evaluation using standard performance metrics, our system is designed to excel across diverse



datasets and real-world scenarios. Furthermore, we prioritize adaptability by incorporating mechanisms for continual learning and adaptation to evolving synthetic speech generation techniques.

The audio preprocessing module standardizes the input by removing noise, detecting speech segments using Voice Activity Detection (VAD), and converting all samples to a consistent format (e.g., 16 kHz mono). The feature extraction module then computes a range of acoustic and learned features, including Mel-frequency cepstral coefficients (MFCCs), pitch, energy, and deep embeddings from pre-trained models such as wav2vec or x-vectors.

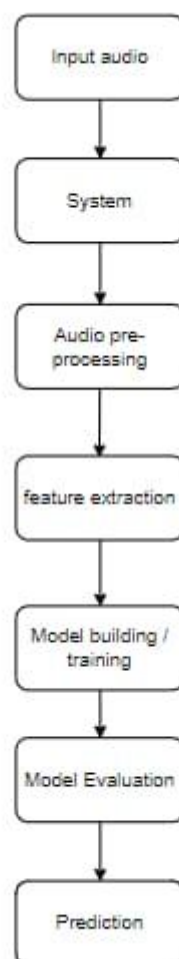
### **3.2.1 ADVANTAGES**

1. **Robustness:** The utilization of state-of-the-art deep learning models, such as CNN, Wave Net, ISTM, and RNN, enhances the system's robustness by enabling it to learn complex patterns and representations from raw audio data, thereby improving its ability to distinguish between synthetic and genuine speech.
2. **Comprehensive Feature Representation:** By extracting both short-term and long-term features from audio data, our system captures a wide range of characteristics associated with synthetic speech, leading to a more holistic representation of the speech signal and improved detection performance.
3. **Adaptability:** Incorporating mechanisms for continual learning and adaptation enables our system to stay abreast of evolving synthetic speech generation techniques, ensuring its effectiveness and relevance over time.
4. **Scalability:** The proposed system is designed to scale effectively to handle large and diverse datasets, making it suitable for deployment

in real-world scenarios where the volume and variability of data may be significant.

5. **High Detection Accuracy:** By combining both traditional audio features (e.g., MFCCs) and deep embeddings (e.g., wav2vec, x-vectors), the system improves the accuracy of distinguishing real from synthetic speech, even when the deepfake audio is highly realistic.

### 3.2.2 WORK FLOW OF PROPOSED SYSTEM



**Fig3.1 Flow Diagram**

### **3.3 SYSTEM REQUIREMENTS**

The system requirements outline the functional and non-functional specifications necessary for the development and implementation of the proposed lung cancer detection system. These requirements serve as the foundation for system design and development efforts.

The proposed Deepfake Audio Detection System requires a combination of hardware, software, and dataset resources to function effectively. From a hardware perspective, a system equipped with a multi-core processor (such as Intel i7 or equivalent), a dedicated GPU (e.g., NVIDIA RTX 3060 or higher), a minimum of 16 GB RAM, and at least 500 GB of SSD storage is recommended to support efficient training and inference of deep learning models. On the software side, the system should operate on a compatible platform such as Ubuntu Linux or Windows 10/11, and utilize Python (version 3.8 or higher) as the primary programming language. Essential libraries include Py Torch or TensorFlow for deep learning, Librosa or torch audio for audio signal processing, and Scikit-learn, NumPy, and Pandas for data handling. For visualization and interpretability, optional tools like Matplotlib, SHAP, and LIME may be employed.

#### **3.3.1 FUNCTIONAL REQUIREMENTS**

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

Examples of functional requirements:

- 1) Authentication of user whenever he/she logs into the system
- 2) System shutdown in case of a cyber-attack

### **3.3.2NON-FUNCTIONAL REQUIREMENTS**

- Security
- Maintainability
- Reliability
- Scalability
- Performance

## **CHAPTER 4**

### **SYSTEM REQUIREMENTS**

#### **4.1 SYSTEM REQUIREMENTS**

This is what is required to run this software. The system requirements are separated into three phase which are hardware, software and personal requirement of the system.

#### **4.2 HARDWARE REQUIREMENTS**

Processor	-I3/Intel Processor
Hard Disk	-160GB
Key Board	-Standard Windows Keyboard
Mouse	-Two or Three Button Mouse
Monitor	-SVGA
RAM	-8GB

#### **4.3 SOFTWARE REQUIREMENTS**

Operating System	-Windows 7/8/10/11
Server-side Script	-HTML, CSS, Bootstrap& JS
Programming Language	-Python
Libraries	-Flask, Pandas, Mysql. connector, Os, Smtplib, Numpy
IDE/Workbench	-PyCharm or VS Code
Technology	-Python 3.6+
Server Deployment	-Xampp Server
Database	-MySQL

## 4.4 FUNCTIONAL REQUIREMENTS

The Functional requirements for the proposed System are:

- **Audio Input Support:** The system must accept audio input in various formats, including WAV, MP3, and FLAC.
- **Real-Time and File-Based Detection:** It should support both real-time streaming audio and pre-recorded file analysis.
- **Audio Preprocessing:** The system must perform preprocessing steps such as noise reduction, normalization, silence removal, and voice activity detection.
- **Feature Extraction:** It should extract relevant features from the audio, including MFCCs, pitch, spectral contrast, and deep features from models like wav2vec or x-vector.
- **Deepfake Classification:** The system must classify audio as either "real" or "fake" using a trained machine learning model.
- **Confidence Scoring:** A confidence score or probability must accompany the classification result, indicating the system's certainty in its decision.
- **Result Visualization:** The system should provide visual outputs (e.g., spectrograms, heatmaps) to support interpretability and analysis of the detection.
- **Logging and Reporting:** It must log detection results with metadata (e.g., timestamp, file name, result, confidence score) and support report generation.
- **Batch Processing Capability:** The system should be able to process multiple audio files in a batch mode.

- **Model Update and Retraining:** It must allow for retraining or fine-tuning of the detection model to accommodate new types of deepfake audio.
- **User Interface (Optional):** A user-friendly interface should allow users to upload files, view results, and download reports.
- **Security and Data Integrity:** The system must ensure the confidentiality and integrity of user data and results.

## **4.5 NON-FUNCTIONAL REQUIREMENTS**

### **4.5.1 Usability:**

Ensuring a user-friendly experience is essential for the website's effectiveness among patients and medical professionals. It should feature intuitive design and easy navigation, minimizing clutter. User feedback is crucial for addressing any usability issues, while compatibility across devices enhances accessibility.

### **4.5.2 Reliability:**

Reliability is paramount in lung cancer detection and classification, where precision is crucial. Even the slightest error could have life-threatening consequences for the patient. Therefore, the system must be meticulously designed and rigorously tested to minimize the risk of misdiagnosis or false results. Accuracy and consistency are non-negotiable when it comes to medical applications, underscoring the importance of robust algorithms and quality assurance measures to ensure reliable performance at all times.

### **4.5.3 Performance:**

Performance is a critical aspect of the lung cancer detection system, requiring both efficiency and accuracy. When dealing with new datasets, it's imperative that the training time is relatively short to facilitate timely analysis and decision-making. Additionally, the classification process must

yield correct results consistently, with a minimal margin of error. This necessitates the use of efficient algorithms and 14 optimization techniques to ensure swift and accurate predictions, thereby enhancing the system's overall performance and utility in clinical settings.

#### **4.5.4 Supportability:**

Supportability of the website is essential for ensuring its accessibility and usability across different browsers. It should be designed to function properly in the latest versions of popular browsers such as Google Chrome and Mozilla Firefox. This ensures that users can access the website seamlessly regardless of their browser preference, thereby maximizing its reach and effectiveness. Regular updates and testing should be conducted to maintain compatibility with evolving browser technologies and standards, ensuring a consistent user experience for all visitors.

#### **4.5.5 Data set requirements:**

The better the training and testing phases may be, the more CT scans and medical data there are available.

The Non-Functional requirements for the proposed system are:

- The system uses the patient's information to anticipate lung cancer



## CHAPTER 5

### SYSTEM DESIGN

#### 5.1 IMPLEMENTATION OF KEY FUNCTIONS

##### 5.1.1 Data preprocessing

Data preprocessing for unmasking deep fake audio involves transforming raw audio data into a format suitable for training machine learning models. This function takes the audio tensor, sample rate, number of samples, and hop length as input and returns the pre-processed audio tensor. The function first trims the audio to the desired length and converts it to mono channel. It then resamples the audio to the desired sample rate and converts it to spectrogram. The spectrogram is then converted to log-mel spectrogram and finally to a tensor. These functions can be used in the data loading and preprocessing pipeline to prepare the audio data for training or testing the model.

Steps performed in this system are as follows:

- Audios are initially loaded as waveforms into the model
- These audios are pre-processed inside the Res Blocks
- Then audios are resampled, padded and normalised, the normalized waveform is converted into Py Torch Tensors.

##### 5.1.2 Feature Extraction

Feature extraction in deepfake detection involves extracting relevant features from audio that can help distinguish between real and fake content. These features can include colour histograms, texture features, or deep learning features such as convolutional neural network (CNN) features. Convolutional layers in neural networks are responsible for feature extraction through the application of filters (kernels) over input data. In the context of audio deepfake detection, these convolutional layers are designed to extract hierarchical features from the input audio spectrograms.

## 5.2 USER GUIDE

1. **Register:** User can Register with their credentials.
  2. **Login:** User can Login with their credentials.
  3. **Input data:** Input audio data.
  4. **View Result:** View System's predicted result.
- Deploy the project folder to your preferred directory.
  - Verify the presence of essential software and libraries.
  - Access the application through the browser by initiating the HTML files.
  - Follow any provided installation scripts or dependency management tools.
  - Seek additional assistance from community forums or support channels if needed.

## 5.3 SYSTEM TESTING

1. **Take Data:** Take user's input audio data.
  2. **Preprocessing:** Clean and prepare data for model training.
  3. **Model Building:** Utilize machine learning algorithms to create a predictive model.
  4. **Generate Results:** Present predictive analysis results
- Verify functionality across diverse datasets to uphold standards.
  - Evaluate model accuracy and reliability.
  - Conduct security testing to identify vulnerabilities.
  - Validate responsiveness across multiple devices.
  - Implement regression testing to ensure updates don't impact existing functionality.

- Gather user feedback for continuous improvement post-launch.

## **5.4 SYSTEM DOCUMENTATION**

Ensure thorough documentation encapsulating system design, functionality, and, implementation details:

- Provide a comprehensive overview of the system's architecture, facilitating a nuanced understanding of its operation.

## **5.5 PROGRAM DOCUMENTATION**

- Leverage the Python programming language for the deep fake audio detection.
- Harness prominent libraries and frameworks such as TensorFlow, PyTorch, NumPy, Pandas, Matplotlib, Librosa, and Mysql for enhanced machine learning capabilities.
- Utilize Visual Studio Code as the preferred development environment for streamlined workflow management.

## **5.6 TECHNICAL DETAILS**

- Provide comprehensive documentation outlining the system architecture and data processing pipelines.
- Offer guidance on data preprocessing techniques and model optimization strategies.
- Include code snippets and sample datasets for implementation.
- Conduct tutorials covering advanced topics like hyperparameter tuning and model interpretation.
- Establish a dedicated support channel for user assistance and collaboration.

## CHAPTER 6

### SYSTEM STUDY AND TESTING

#### 6.1 FEASIBILITY STUDY

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key

- Economical feasibility
- Social feasibility
- Technical feasibility

considerations involved in the feasibility analysis are

**Economical Feasibility:** This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

**Technical Feasibility:** This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical

resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**Social Feasibility:** The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

### **System Testing**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

## **6.2 TYPES OF TESTING**

### **6.2.1 Unit Testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

### **6.2.2 Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components. Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.The task of the integration test is to check that components or software applications, e.g. components

in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing:** User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

### 6.2.3 Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be expected.
Output	: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

#### **6.2.4 White box Testing**

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

#### **6.2.5 Black box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.



**Table 6.1 Test Case**

<b>Input</b>	<b>Output</b>	<b>Result</b>
Input	Tested for different audio.	Success
Prediction	Prediction will be performed using to build from the algorithm.	Success

**Table 6.2 Test Case Model Building**

<b>S.N O</b>	<b>Test cases</b>	<b>I/O</b>	<b>Expected O/T</b>	<b>Actual O/T</b>	<b>P/F</b>
1	Read the datasets.	Dataset's path.	Datasets need to read successfully .	Datasets fetched successfully.	It produced P. If this not F will come
2	prediction	Predict result for the inputted audio by using given algorithms.	Prediction as output	Prediction as output	It produced Real. If this is not, it will undergo Fake.

## CHAPTER 7

### SYSTEM IMPLEMENTATION

The implementation of the Deepfake Audio Detection System involves a structured pipeline consisting of audio input handling, preprocessing, feature extraction, model training, classification, and result interpretation. The system is developed using Python due to its strong ecosystem for machine learning, audio processing, and deep learning.

#### 7.1 AUDIO INPUT AND PREPROCESSING

The system begins by accepting audio input in various formats, such as WAV, MP3, and FLAC. Each input undergoes preprocessing using the **Librosa** and **torch audio** libraries. Key preprocessing steps include noise reduction, silence trimming, and resampling to a standard 16 kHz mono format. Voice Activity Detection (VAD) is applied to isolate speech segments, ensuring that non-speech portions do not interfere with analysis.

#### 7.2 FEATURE EXTRACTION

Once the audio is pre-processed, acoustic features are extracted to capture both low-level and high-level audio characteristics. These features include Mel Frequency Cepstral Coefficients (MFCCs), chroma features, zero-crossing rate, and spectral contrast. Additionally, deep audio embeddings are obtained using pre-trained models such as **wav2vec 2.0** or **x-vector**. These embeddings offer a robust representation of the audio signal, improving the system's ability to distinguish subtle differences between real and synthetic speech.

### 7.3 MODEL TRAINING AND CLASSIFICATION

The extracted features are fed into a supervised machine learning model. In the current implementation, a **Convolutional Neural Network (CNN)** is used for spectrogram-based classification, while alternative architectures such as **Recurrent Neural Networks (RNN)** and **Transformer-based models** are also tested for improved context modelling. The model is trained using a balanced dataset comprising real and deepfake audio samples generated from various TTS (Text-to-Speech) and voice cloning tools (e.g., Tacotron2, Wave Net, Lyrebird, Descript Overdub).

### 7.4 DEPLOYMENT

The system is deployed on a Linux-based server and is designed to support both local and cloud-based execution. For real-time use cases, such as call monitoring or media verification, the model is optimized for low-latency inference using **ONNX Runtime** or **Tensor RT**. Containerization with **Docker** is used to ensure portability and ease of deployment across various environments.

### 7.5 LOGGING AND REPORTING

All detection results, including input file metadata, prediction outcomes, confidence scores, and timestamps, are logged in a structured format (e.g., CSV or database). This enables traceability and further analysis of system performance.

## CHAPTER 8

### CONCLUSION AND FUTURE ENHANCEMENT

#### 8.1 CONCLUSION

project on synthetic speech detection through short-term and long-term prediction traces represents a significant advancement in addressing the challenges posed by increasingly sophisticated synthetic speech technologies. By leveraging state-of-the-art deep learning algorithms and comprehensive feature extraction techniques, we have developed a robust system capable of accurately identifying synthetic speech instances across diverse contexts. Through rigorous model training, optimization, and evaluation, we have demonstrated the effectiveness and reliability of our approach in achieving high accuracy and robustness in synthetic speech detection. Looking ahead, there are ample opportunities for future enhancement and refinement, including the integration of advanced models, exploration of multi-modal approaches, and optimization for real-time processing and edge device deployment. Collaborative research efforts and continued exploration of novel techniques promise to further elevate the capabilities of our system and ensure its adaptability to evolving synthetic speech generation techniques. Ultimately, our project contributes to enhancing security, trust, and authenticity in communication channels by mitigating the risks associated with synthetic speech. By providing a comprehensive and effective solution for synthetic speech detection, we aim to foster a safer and more trustworthy digital environment for individuals and organizations alike.

## 8.2 FUTURE ENHANCEMENT

In the realm of synthetic speech detection, future enhancements hold the promise of refining our system's capabilities and expanding its applicability across diverse domains. Integration of advanced deep learning models, including Transformer-based architectures and Generative Adversarial Networks (GANs), could elevate detection accuracy and robustness to unprecedented levels. Furthermore, a multi-modal approach, incorporating textual and visual data alongside audio signals, presents an opportunity to bolster detection performance, especially in complex scenarios. Embracing semi-supervised learning techniques can leverage unlabeled data to further enhance the system's effectiveness, particularly in resource-constrained environments. Real-time processing optimization would enable swift detection, ideal for applications requiring instantaneous response, such as live streaming platforms. Domain adaptation strategies could ensure the system's versatility across different contexts, while privacy-preserving techniques would instill confidence in users regarding data confidentiality. Integrating user feedback mechanisms allows continuous refinement based on real-world usage patterns. Continued exploration of novel feature extraction techniques promises to capture nuanced aspects of synthetic speech more accurately. Lightweight versions optimized for edge device deployment would extend the system's reach to IoT and mobile applications. Collaborative research efforts across disciplines can synergize advancements and insights, propelling the system towards even greater efficacy in safeguarding communication channels against synthetic speech threats. Through these future enhancements, our system stands poised to evolve and meet the evolving challenges head-on, ensuring a safer and more trustworthy digital landscape for all.

## CHAPTER 9

### APPENDICES

#### 9.1 SOURCE CODE

```

from flask import Flask, url_for, redirect, render_template, request, session
import mysql.connector
import pandas as pd
import joblib
import os
import numpy as np
import tensorflow as tf
import librosa

from tensorflow.keras.models import load_model

def extract_mfcc(file_path, max_pad_len=174):
    try:
        audio, sample_rate = librosa.load(file_path, res_type='kaiser_fast')
        mfccs = librosa.feature.mfcc(y=audio, sr=sample_rate, n_mfcc=40)
        if mfccs.shape[1] > max_pad_len:
            mfccs = mfccs[:, :max_pad_len]
        else:
            pad_width = max_pad_len - mfccs.shape[1]
            mfccs = np.pad(mfccs, pad_width=((0, 0), (0, pad_width)),
                           mode='constant')
        except Exception as e:
            print(f'Error encountered while parsing file {file_path}: {e}')
        return None
        return mfccs

```

```

def predict_audio_class(file_path, model_path='cnn.h5'):
    # Load the model
    model = load_model(model_path)
    # Extract features from the audio file
    features = extract_mfcc(file_path)
    if features is None:
        print("Could not extract features from the file")
        return None
    # Reshape the features to match the input shape of the model
    features = features[np.newaxis, ..., np.newaxis]
    # Predict the class of the audio file
    prediction = model.predict(features)
    predicted_class = np.argmax(prediction, axis=1)
    # Translate the predicted class index into a meaningful label
    class_labels = ['Real', 'Fake'] # Adjust according to your classes
    predicted_label = class_labels[predicted_class[0]]
    return predicted_label

app = Flask(__name__)
app.secret_key = 'admin'
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    port="3306",
    database='deep_fake'
)
mycursor = mydb.cursor()
def executionquery(query,values):
    mycursor.execute(query,values)

```

```

mydb.commit()
return
def retrievequery1(query,values):
mycursor.execute(query,values)
data = mycursor.fetchall()
return data
def retrievequery2(query):
mycursor.execute(query)
data = mycursor.fetchall()
return data
@app.route('/')#http://127.0.0.1:5000/
def index():
return render_template('index.html')
@app.route('/register',methods=["GET",
"POST"])#http://127.0.0.1:5000/register
def register():
if request.method == "POST":
email = request.form['email']
password = request.form['password']
c_password = request.form['c_password']
if password == c_password:
query = "SELECT UPPER(email) FROM users"
email_data = retrievequery2(query)
email_data_list = []
for i in email_data:
email_data_list.append(i[0])
if email.upper() not in email_data_list:
query = "INSERT INTO users (email, password) VALUES (%s, %s)"
values = (email, password)

```



```

executionquery(query, values)
return render_template('login.html', message="Successfully Registered!")
return render_template('register.html', message="This email ID is already
exists!")
return render_template('register.html', message="Conform password is not
match!")
return render_template('register.html')
@app.route('/login', methods=["GET", "POST"])
def login():
    if request.method == "POST":
        email = request.form['email']
        password = request.form['password']
        query = "SELECT UPPER(email) FROM users"
        email_data = retrievequery2(query)
        email_data_list = []
        for i in email_data:
            email_data_list.append(i[0])
        if email.upper() in email_data_list:
            query = "SELECT UPPER(password) FROM users WHERE email = %s"
            values = (email,)
            password__data = retrievequery1(query, values)
            if password.upper() == password__data[0][0]:
                global user_email
                user_email = email
                return render_template('home.html')
            return render_template('login.html', message= "Invalid Password!!")
        return render_template('login.html', message= "This email ID does not
exist!")
        return render_template('login.html')

```

```

@app.route('/home')
def home():
    return render_template('home.html')

@app.route('/upload', methods=["GET", "POST"])
def upload():
    if request.method == "POST":
        myfile = request.files['file']
        fn = myfile.filename
        accepted_formats = ['mp3', 'wav', 'ogg', 'flac']
        if fn.split('.')[-1].lower() not in accepted_formats:
            message = "Invalid file format. Accepted formats: {}".format(', '.join(accepted_formats))
            return render_template("audio.html", message = message)
        mypath = os.path.join('static/audio/', fn)
        myfile.save(mypath)
        predicted_class = predict_audio_class(mypath)
        print(f'Predicted class: {predicted_class}')
        return render_template('upload.html', result=predicted_class)
    return render_template('upload.html')

if __name__ == '__main__':
    app.run(debug = True)

```

## 9.2 SCREENSHOTS



Figure 9.1 Home page

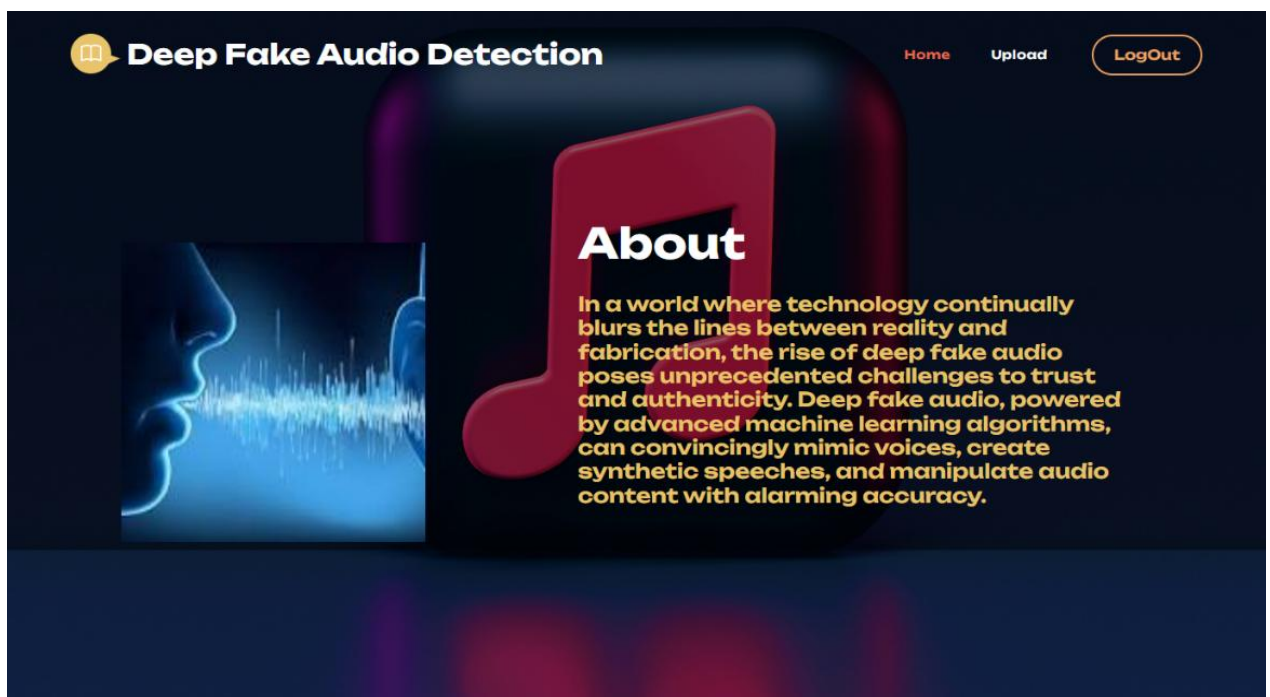
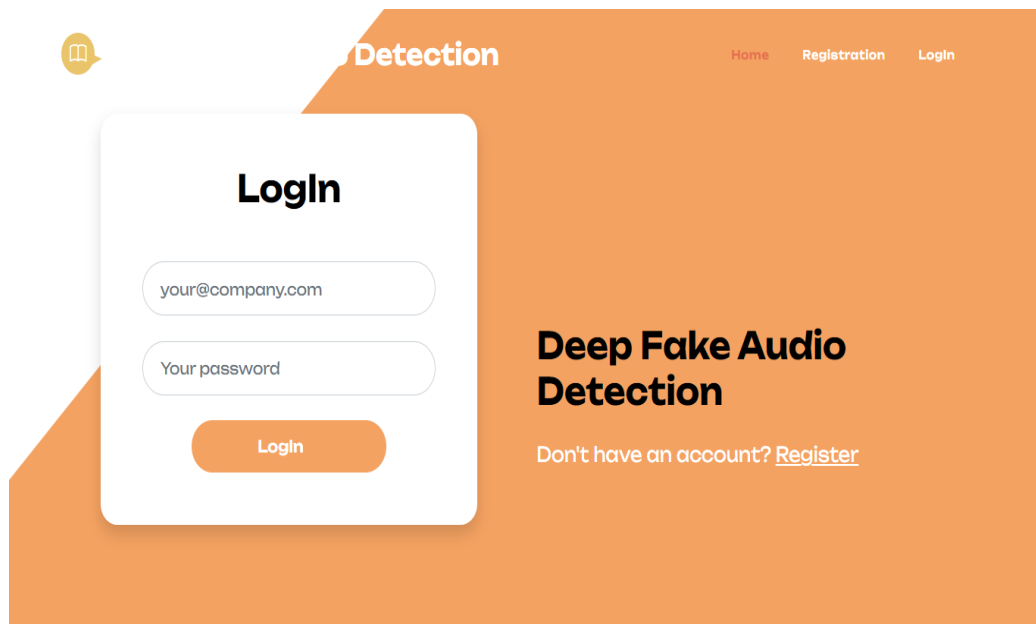


Figure 9.2 About page



The login page features a white modal box on an orange background. The modal is titled "Login" and contains two input fields: "your@company.com" and "Your password". Below the fields is an orange "Login" button. To the right of the modal, the text "Deep Fake Audio Detection" is displayed in large, bold letters, followed by the link "Don't have an account? [Register](#)". The top navigation bar includes a logo, the title "Detection", and links for "Home", "Registration", and "Login".

Detection

Home Registration Login

## Login

your@company.com

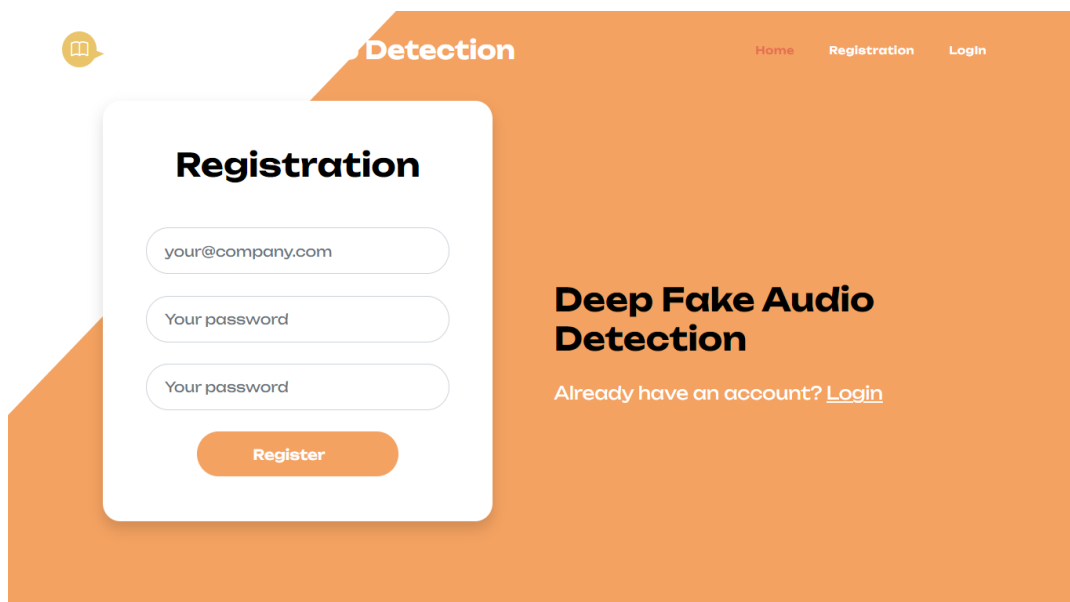
Your password

Login

## Deep Fake Audio Detection

Don't have an account? [Register](#)

Figure 9.3 Login page



The registration page features a white modal box on an orange background. The modal is titled "Registration" and contains three input fields: "your@company.com", "Your password", and "Your password". Below the fields is an orange "Register" button. To the right of the modal, the text "Deep Fake Audio Detection" is displayed in large, bold letters, followed by the link "Already have an account? [Login](#)". The top navigation bar includes a logo, the title "Detection", and links for "Home", "Registration", and "Login".

Detection

Home Registration Login

## Registration

your@company.com

Your password

Your password

Register

## Deep Fake Audio Detection

Already have an account? [Login](#)

Figure 9.4 Registration page

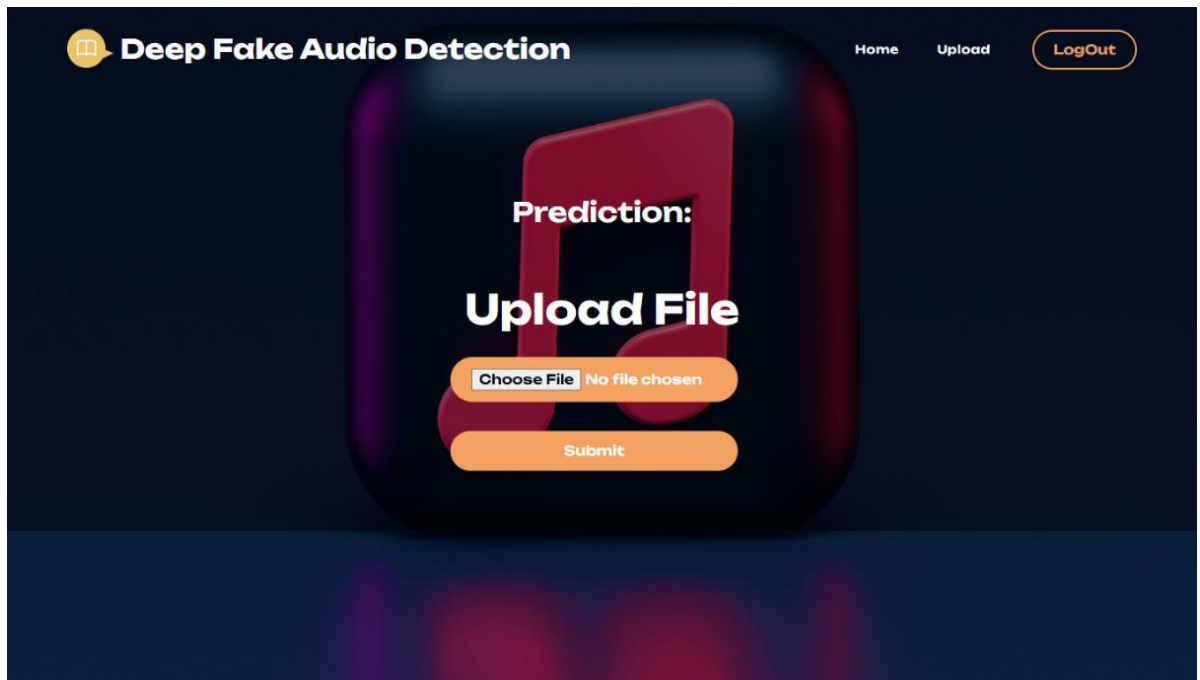


Figure 9.5 Upload page

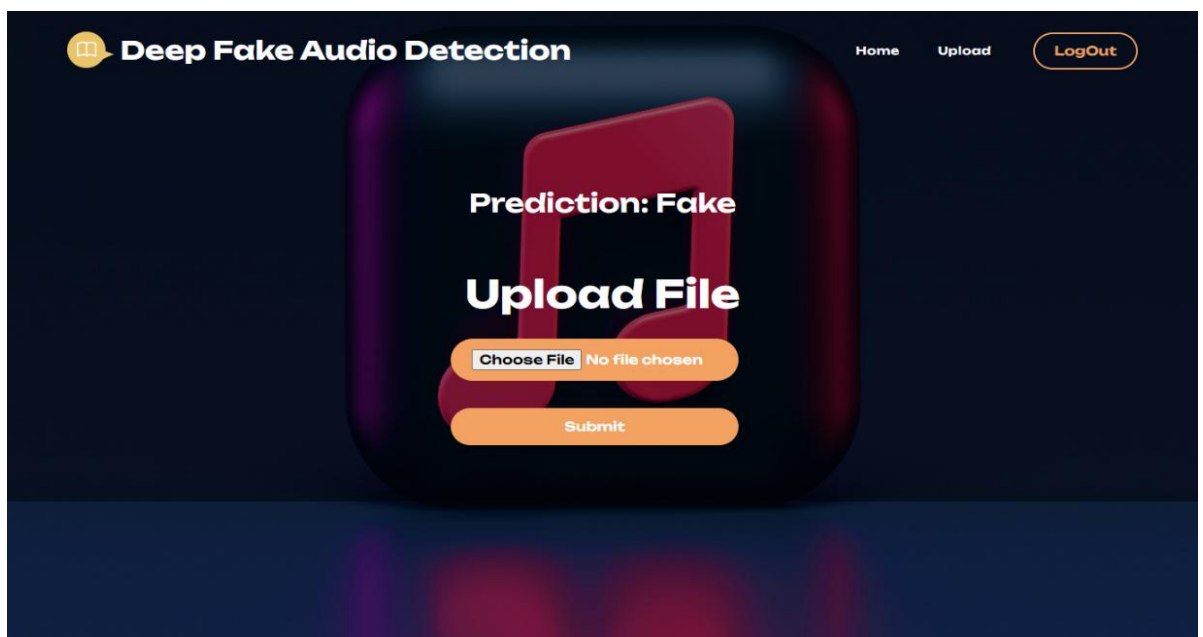
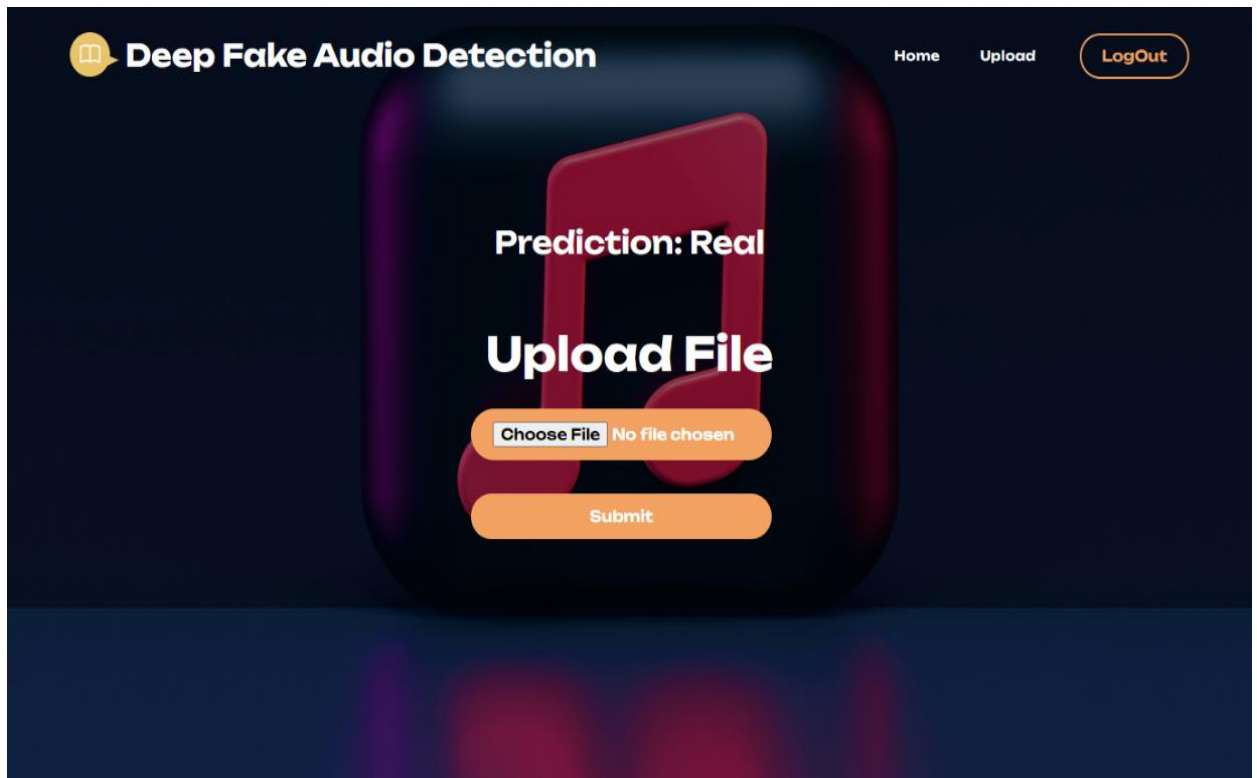


Figure 9.6 Prediction Page (Fake)



**Figure 9.7 Prediction page (Real)**

## REFERENCE

- [1] O. A. Shaaban, R. Yildirim and A. A. Alguttar, "Audio Deepfake Approaches," in IEEE Access, vol. 11, pp. 132652-132682, 2023, doi: 10.1109/ACCESS.2023.3333866.
- [2] Xie, Yuankun & Cheng, Haonan & Wang, Yutian & Ye, Long. (2023). Learning A Self-Supervised Domain-Invariant Feature Representation for Generalized Audio Deepfake Detection. 2808-2812. 10.21437/Interspeech.2023-1383.
- [3] D. Salvi, B. Hosler, P. Bestagini, M. C. Stamm and S. Tubaro, "TIMIT-TTS: A Text-to-Speech Dataset for Multimodal Synthetic Media Detection," in IEEE Access, vol. 11, pp. 50851-50866, 2023, doi: 10.1109/ACCESS.2023.3276480.
- [4] S. Salman, J. A. Shamsi and R. Qureshi, "Deep Fake Generation and Detection: Issues, Challenges, and Solutions," in IT Professional, vol. 25, no. 1, pp. 52-59, Jan.-Feb. 2023, doi: 10.1109/MITP.2022.3230353.
- [5] D. U. Leonzio, L. Cuccovillo, P. Bestagini, M. Marcon, P. Aichroth and S. Tubaro, "Audio Splicing Detection and Localization Based on Acquisition Device Traces," in IEEE Transactions on Information Forensics and Security, vol. 18, pp. 4157-4172, 2023, doi: 10.1109/TIFS.2023.3293415.
- [6] Cheng, Harry & Guo, Yangyang & Wang, Tianyi & Li, Qi & Chang, Xiaojun & Nie, Liqiang. (2023). Voice-Face Homogeneity Tells Deepfake. ACM Transactions on Multimedia Computing, Communications, and Applications. 20. 10.1145/3625231.
- [7] C. Sun, S. Jia, S. Hou and S. Lyu, "AI-Synthesized Voice Detection Using Neural Vocoder Artifacts," in 2023 IEEE/CVF Conference on

Computer Vision and Pattern Recognition Workshops (CVPRW), Vancouver, BC, Canada, 2023 pp. 904-912.

[8] R. Mubarak, T. Alsboui, O. Alshaikh, I. Inuwa-Dutse, S. Khan and S. Parkinson, "A Survey on the Detection and Impacts of Deepfakes in Visual, Audio, and Textual Formats," in IEEE Access, vol. 11, pp. 144497-144529, 2023, doi: 10.1109/ACCESS.2023.3344653.

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

[10] Luca Cuccovillo, "Open Challenges in Synthetic Speech Detection", Fraunhofer Institute for Digital Media Technology IDMT, 2022.

[11] Federica LATORA, "Synthetic Speech Detection Algorithms", UNIVERSITY OF PADOVA, 2022.

[12] D. Nagothu, R. Xu, Y. Chen, E. Blasch and A. Aved, "DeFakePro: Decentralized Deepfake Attacks Detection Using ENF Authentication" in IT Professional, vol. 24, no. 05, pp. 46-52, 2022.

[13] C. Hazirbas, J. Bitton, B. Dolhansky, J. Pan, A. Gordo and C. C. Ferrer, "Towards Measuring Fairness in AI: The Casual Conversations Dataset," in IEEE Transactions on Biometrics, Behavior, and Identity Science, vol. 4, no. 3, pp. 324-332, July 2022

[14] Clara Borrelli, Paolo Bestagini, Fabio Antonacci, Augusto Sarti & Stefano Tubaro, "Synthetic speech detection through short-term and long-term prediction traces", EURASIP Journal on Information Security volume 2021.

[15] Hareesh Mandalapu, Aravinda Reddy P N, Raghavendra Ramachandra, K Sreenivasa Rao, Pabitra Mitra, S R Mahadeva



Prasanna, Christoph Busch,” Multilingual Audio-Visual Smartphone Dataset And Evaluation,2021.

[16] Monisankha Pal, Dipjyoti Paul, Goutam Saha, “Synthetic speech detection using fundamental frequency variation and spectral features” J. 2018.

[17] Yuki Saito, Shinnosuke Takamichi, H. Saruwatari, “Statistical Parametric Speech Synthesis Incorporating Generative Adversarial Networks” , 2017.

[18] Cheng, Harry & Guo, Yangyang & Wang, Tianyi & Li, Qi & Chang, Xiaojun & Nie, Liqiang. (2023). Voice-Face Homogeneity Tells Deepfake. ACM Transactions on Multimedia Computing, Communications, and Applications. 20. 10.1145/3625231.

[19] Rana, Md & Nobi, Mohammad & Murali, Beddhu & Sung, Andrew. (2022). Deepfake Detection: A Systematic Literature Review. IEEE Access. 10. 1-1. 10.1109/ACCESS.2022.3154404.

[20] Daniel Felps, Heather Bortfeld, Ricardo Gutierrez-Osuna, Foreign accent conversion in computer assisted pronunciation training,ISSN 0167-6393.

[21] Alexander B. Kain, John-Paul Hosom, Xiaochuan Niu, Jan P.H. van Santen, Melanie Fried-Oken b, Janice Staehelyb,” Improving the Intelligibility of Dysarthric Speech”, Speech Communication 2007.