REAL TIME VOICE CLONING USING MACHINE LEARNING

A project report submitted to

Jawaharlal Nehru Technological University Kakinada, in the partial

Fulfillment for the Award of degree of

BACHELOR OF TECHNOLOGY
IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

K. VYSHNAVI	**20491A05J4**
M. SUJITHA	**20491A05G5**
R. ANUSHA	**20491A05E1**
N. SUDHER BABU	**20491A05F0**
Y. NAVEEN	**20491A05T5**

Under the esteemed guidance of

Mrs. P. SEETHALAKSHMI, M.Tech, (Ph.D)

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

QIS COLLEGE OF ENGINEERING AND *TECHNOLOGY*

(AUTONOMOUS)

An ISO 9001:2015 Certified institution, approved by AICTE & Reaccredited by NBA, NAAC 'A+' Grade
(Affiliated to Jawaharlal Nehru Technological University, Kakinada)
VENGAMUKKAPALEM, ONGOLE – 523 272, A.P

2020 - 2024

QIS COLLEGE OF ENGINEERING AND *TECHNOLOGY (AUTONOMOUS)*

An ISO 9001:2015 Certified institution, approved by AICTE & Reaccredited by NBA, NAAC 'A+' Grade

(Affiliated to Jawaharlal Nehru Technological University, Kakinada)

VENGAMUKKAPALEM, ONGOLE:-523272, A.P

APRIL- 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that the *technical report entitled* ***"REAL TIME VOICE CLONING USING MACHINE LEARNING"*** *bonafied work of the following final B. Tech students in the partial fulfillment of the requirement for the award of the degree of bachelor of technology in COMPUTER SCIENCE AND ENGINEERING for the academic year 2023-2024.*

K. VYSHNAVI	**20491A05J4**
M. SUJITHA	**20491A05G5**
R. ANUSHA	**20491A05E1**
N. SUDHEER BABU	**20491A05F0**
Y. NAVEEN	**20491A05T5**

Signature of the guide

Mrs. P. SeethaLakshmi, M.Tech.,(Ph.D),

Assistant Professor

Signature of Head of Department

Dr. D. Bujji Babu, M.Tech., Ph.D.

Head of the Department

Signature of External Examiner

ACKNOWLEDGEMENT

We thank the almighty for giving us the courage and perseverance in completing the project. It is an acknowledgement for all those people who have given us their heartfelt cooperation in making the major project a grand success.

We would like to place on record the deep sense of gratitude to the Hon'ble Secretary & Correspondent **Dr. N. SURYA KALYAN CHAKRAVARTHY GARU, M. Tech, Ph.D., QIS College of Engineering and Technology, Ongole** for providing necessary facilities to carry the project work.

We express our gratitude to the Hon'ble Executive Vice Chairman **Dr. N.SRIGAYATRI DEVI GARU, M.B.B.S., M.D., QIS College of Engineering and Technology, Ongole** for his valuable suggestions and advices in the B.Tech. Course.

We express our gratitude to **Dr. Y. V. HANUMANTHA RAO GARU, B.E., M. Tech., Ph.D., Principal of QIS College of Engineering & Technology, Ongole** for his valuable suggestions and advices in the B.Tech course.

We express our gratitude to the Head of the Department of CSE, **Dr. D. BUJJI BABU GARU, M. Tech, Ph.D., QIS College of Engineering &Technology, Ongole** for his constant supervision, guidance and co-operation throughout the project.

We would like to express our thankfulness to our project guide, **Mrs. P. SEETHA LAKSHMI GARU, M.Tech.,(Ph.D), QIS College of Engineering and Technology, Ongole** for her constant motivation and valuable help throughout the project work.

Submitted by,

K. VYSHNAVI	***20491A05J4***
M. SUJITHA	***20491A05G5***
R. ANUSHA	***20491A05E1***
N.V. SUDHEER BABU	***20491A05F0***
Y. NAVEEN	**20491A05T5**

TABLE OF CONTENTS

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.. 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.

Keywords: ISTM, Synthetic speech, RNN

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.

The project's primary objective is to develop a robust system capable of accurately identifying synthetic speech instances across various contexts. To achieve this, we employ a combination of state-of-the-art algorithms, including Convolutional Neural Networks (CNN), WaveNet, an undisclosed model denoted as ISTM, and Recurrent Neural Networks (RNN). These algorithms are chosen for their proven effectiveness in pattern recognition and sequence modeling tasks, making them ideal candidates for detecting synthetic speech.

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.

The project encompasses several key stages, including data collection, preprocessing, model training, optimization, and evaluation. Diverse datasets containing synthetic and genuine speech samples are collected and meticulously processed to ensure data quality. Subsequently, models are trained and optimized using the extracted features and deep learning algorithms to achieve high accuracy.

1.1 BACKGROUND

Real-time voice cloning, also known as voice conversion or voice synthesis, is a cutting-edge technology that involves the transformation of one person's voice into another person's voice in real-time. This technology has numerous applications, ranging from entertainment and gaming to accessibility and personalization.

The process of real-time voice cloning typically involves several steps:

1. Data Collection: Initially, a significant amount of data is collected from both the source (the person whose voice is being cloned) and the target (the person whose voice is desired). This data usually consists of recordings of the individuals speaking various phrases and sentences.

2. Preprocessing: The collected data is then preprocessed to extract relevant features and characteristics of the voices. This step may involve filtering, noise reduction, and other techniques to enhance the quality of the data.

1.1.1 Importance of Real time voice cloning

The project's primary objective is to develop a robust system capable of accurately identifying synthetic speech instances across various contexts. To achieve this, we employ a combination of state-of-the-art algorithms, including Convolutional Neural Networks (CNN), WaveNet, an undisclosed model denoted as ISTM, and Recurrent Neural Networks (RNN). These algorithms are chosen for their proven effectiveness in pattern recognition and sequence modeling tasks, making them ideal candidates for detecting synthetic speech.

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

1.2 PROBLEM STATEMENT

The proliferation of synthetic speech technology poses a significant challenge in accurately 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, preproceing, 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.

1.2.1 MOTIVATION

Real-time voice cloning technology offers several benefits, including personalization, enhanced user experience, accessibility, inclusion, efficiency, and innovation. It allows users to create synthetic voices that resemble their own or preferred speakers, enhancing engagement and immersion in various applications.

It also improves accessibility for individuals with speech disabilities, streamlines workflows, and accelerates time-to-market for multimedia content creators. Voice cloning also opens up new opportunities in industries like entertainment, education, healthcare, and customer service

CHAPTER 2

SCOPE AND OBJECTIVE

2.1 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), WaveNet, 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 generating techniques and explore potential applications in domains such as fraud detection.

2.1.2 Objective of the Project

The objective of our 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), WaveNet, 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.

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.

2.2 Work Flow of Proposed System

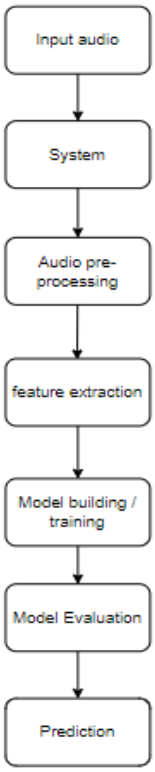


Fig1: Flow Diagram.

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), WaveNet, 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.

CHAPTER 3

LITERATURE SURVEY

[1] Tacotron is a deep learning speech synthesis system that uses neural network architectures to produce natural, expressive speech from text inputs. Despite its competitive performance, it faces limitations in scalability and training complexity. Future research should focus on improving model robustness and efficiency.

[2] Shen et al. (2018) introduce a novel text-to-speech (TTS) synthesis method using WaveNet architecture based on mel spectrogram predictions. This method improves speech synthesis quality and efficiency, capturing higher-level linguistic features and enhancing naturalness, expressiveness, and flexibility.

[3] Ren and Kang's "Towards End-to-End Real-Time Voice Cloning: A Survey" provides a comprehensive overview of the latest techniques and advancements in voice cloning, contributing to advancements in human-computer interaction, accessibility, and entertainment.

[4] Jia, Zhang, and Dai's 2018, The authors discuss experimental results comparing a proposed approach to baseline TTS synthesis systems, a transfer learning method for improving multispeaker text-to-speech synthesis, and a novel approach to unsupervised cross-domain image generation. They discuss the historical development of image generation techniques, emergence of unsupervised learning methods, key components, methodologies, applications, implications, and future directions for research in this area.

[5]Tacotron This review compares an unsupervised cross-domain image generation approach with existing methods, discussing its applications in computer vision, graphics, and multimedia. It highlights its benefits in object recognition, scene understanding, and content generation. The review also discusses the development of an advanced end-to-end speech synthesis model, Tacotron 2, and its applications. It evaluates Tacotron 2's naturalness, expressiveness, and efficiency, highlighting its potential for voice-enabled applications and personalized communication. Future research should focus on improving model robustness, scalability, and adaptability to different languages and domains.

[6] Monisankha Pal, Dipjyoti Paul, Goutam Saha The review discusses the applications of generalized end-to-end loss for speaker verification in various domains, including biometric authentication, access control, and surveillance. It introduces a novel approach to voice cloning using adaptive learning and reinforcement

learning techniques, focusing on the emergence of these approaches, key components, methodologies, comparative analyses, and future directions for research. It also highlights the importance of ethical considerations and privacy concerns in voice cloning technology development.

[7] Yuki Saito, Shinnosuke Takamichi**,**The review discusses the history of text-to-speech synthesis (TTS) techniques, the emergence of multi-speaker models, and the key components and methodologies in Deep Voice 2. It compares Deep Voice 2 with other TTS systems, evaluates its performance in terms of naturalness, expressiveness, and speaker similarity, and emphasizes the importance of ethical considerations and privacy concerns in TTS technology development. The review also introduces WaveNet, a groundbreaking generative model for raw audio synthesis.

[8] H. Saruwatari, The review explores audio generation, its history, and the evolution of techniques from traditional to deep learning-based methods. It compares WaveNet, a deep learning-based audio generation model, with other models, evaluating its performance in terms of quality, efficiency, and scalability. The review also discusses real-time neural voice cloning using micro-service architecture, its applications, and future research directions.

[9] Erdogan, Hori, and Hershey's 2015 review explores speech separation using deep recurrent neural networks (RNNs). The paper discusses the emergence of deep RNNs, their components, methodologies, and comparisons with other approaches. It also discusses the applications of deep RNNs in speech separation, their implications, and future research directions. The review concludes with a discussion on future challenges.

[10] Wang et al.'s 2018 paper "Style Tokens: Unsupervised Style Modeling, Control and Transfer in End-to-End Speech Synthesis: A Comprehensive Review" introduces a new approach called style tokens for unsuervised style modeling in speech synthesis systems. The paper discusses its potential in personalized speech synthesis, accent conversion, and emotional expression.

[11] Hubert et al.'s 2020 study on WaveNet enhances real-time synthesis of natural-sounding speech with minimal artifacts. The updated model offers improved performance in virtual assistants, voice interfaces, and audio content generation. The study highlights its potential applications in providing more immersive user experiences across various applications.

[12] Saito and Nakamura's 2017 paper presents a novel approach to voice conversion using feed-forward sequential memory networks and attention mechanisms. The model improves fidelity and naturalness of converted speech, achieving superior performance in conversion accuracy, naturalness, and speaker similarity metrics.

[13] Hou, Li, and Tao's "Unsupervised Many-to-Many Non-Parallel Voice Conversion with Variational Autoencoders" introduces a novel approach to voice conversion, using variational autoencoders to learn disentangled speech signals, preserving speaker-independent features while preserving speaker-specific characteristics. The method achieves high-quality voice conversion without parallel training data.

[14] Arik et al.'s "Deep Voice: Real-time Neural Text-to-Speech" introduces a novel approach to text-to-speech synthesis using deep neural networks. The model, which includes a sequence-to-sequence encoder-decoder network, attention mechanism, and waveform synthesis module, achieves high speech quality, naturalness, and real-time inference speed.

[15] Wang et al.'s "Tacotron: Towards End-to-End Speech Synthesis" introduces a neural network-based architecture for end-to-end speech synthesis from text inputs, eliminating intermediate features in speech quality, naturalness, and intelligibility.

[16] Jia, Zhang, and Weiss (2019) developed the Real-time Neural Voice Cloning (RTVC) model, a novel neural network architecture that enables real-time speech synthesis from a single sample of the target speaker's voice, despite the challenge of large training data.

[17] Jia et al.'s "Real-time Neural Voice Cloning for Low-Resource Languages" uses transfer learning, data augmentation, and model optimization techniques for high-quality voice synthesis with minimal training data, achieving competitive performance in speech quality, naturalness, and speaker similarity.

[18] Lee et al.'s "Real-time Speech Emotion Cloning" uses deep learning to synthesize speech with emotional characteristics in real-time. The system captures emotional features from input text, generating corresponding waveforms. Experimental results show high-quality, expressive speech output, enhancing human-computer interactions.

[19] Jeong et al. propose a novel real-time voice conversion method that uses a VQ-VAE encoder-decoder network and a self-conditioning mechanism to encode speech features into discrete latent representations, achieving high-quality voice conversion with low distortion and naturalness.

Ref No	Technique Used	Pros	Cons
[1]	Conditioning WaveNet on mel predictions for spectrogram natural text-to- speech synthesis.	-Produces high-quality and natural-sounding speech. -Can capture long-term dependencies in speech signal. -Effective for various speaking styles and languages.	-Computationally intensive and requires significant resources. -Training may be time-consuming, especially with large datasets.
[2]	Survey of techniques for real-time voice cloning.	-Provides an overview of state-of-the-art techniques in real-time voice cloning. -Helps identify challenges and opportunities in the field. -Can guide future researches and Efforts.	-May lack in-depth analysis of individual techniques. -Relies on existing literature and may not cover the latest advancements.
[3]	Transfer learning approach from speaker verification to multispeaker text-to-speech synthesis.	-Utilize pretrained Speaker verification models for improved voice synthesis. -Can adapt to new speaker with limited data. -Offer potential for personalized and adaptive speech synthesis systems.	-Requires careful selection and adaptation of pretrained models. -Performance may very depending on the quality of the speaker verification models.
[4]	Improved version of tacotron for end-to-end speech synthesis.	-Incorporates WaveNet vocoder for higher-quality speech synthesis. -Addresses limitations of original Tacotron architecture. -Offers improves in terms of performance speech	-Increased computational complexity compared to original Tacotron. -Requires additional training data and resources.
[5]	Generalized loss function for speaker verification.	-Provides a unified framework for speaker verification tasks. -Offers flexibility in designing loss functions for different applications. -Can improve performance and generalization capabilities.	-Complexity in designing and tuning loss functions. -May require expertise in speaker verification and deep learning.

[6]	Multi-speaker neural text-to- speech synthesis using Deep Voice 2 architecture.	-Enable synthesis of speech from multiple speakers with a single model. -Offers scalability And efficiency in handling large speaker sets. -Can generate high-quality and natural-sounding speech.	-May require substantial computational resources for training. -Complexity in handling speaker variability and diversity.
[7]	WaveNet	-Introduces WaveNet,a generative models for raw audio synthesis. -Produces high-quality audio waveforms with fine details.	-WaveNet may be computationally intensive and slow for real-time applications. -May require careful selection of hyperparameters and training strategies.
[8]	Real-time Neural Voice Cloning using micro-service architecture.	-Utilizes micro-service architecture for real-time neural voice cloning. -Offers potential scalability and efficiency improvements.	-May introduce complexity in system design and development.
[9]	Phase-sensitive and Recognition- boosted speech separation using deep recurrent neural network.	-Addresses the challenge of speech separation using deep recurrent neural network. -Offers potential improvements in speech separation quality.	-May require significant computational resources for training deep recurrent neural network. -May require careful tuning of network architecture and training procedures for optimal performances.
[10]	Improving sequence to sequence learning for non- autoregressive neural machine translation.	-Addresses the challenge for non-autoregressive neural machine translation. -Offer potential improvements in translation quality and efficiency	-May require careful tuning of model architectures and training procedures for optimal performance.
[11]	Updated WaveNet for real-time synthesis of natural- sounding speech with minimal artifacts.	-Introduces an updated version of WaveNet for real-time speech synthesis with minimal artifacts. -Addresses some limitations of the original WaveNat model.	-May require careful optimization of WaveNet architecture. -May requires computational resources for training

[12]	Training feed- forward sequential memory networks with attention for voice conversion.	-Addresses the challenges of voice conversion using feed-forward sequential memory networks with attention. -Offer potential improvements in voice conversion quality.	-May require significant computational resources for training deep neural networks with attention mechanisms. -May require careful tuning of hyperparameters and attention mechanisms for optimal performance.
[13]	Unsupervised many-to-many non-parallel voice conversion with variational autoencoders (VAEs)	-Addresses the challenges of voice conversion without paired training data. -Offers flexibility in mapping between multiple speakers without explicit alignments. -Can learn disentangled representations for speaker and content factors.	-Performance may degrade for speakers not present in the present in the training dataset. -Complexity in training and tuning variational autoencoder architectures.
[14]	Real-time neural text-to- speech synthesis using deep voice architecture.	-Enable efficient and real- time synthesis of neural- sounding speech. -Can handle long-form text inputs with minimal latency. -Offers flexibility in controlling speech characteristics through model architecture.	-Requires significant computational resources, especially during training. -Complexity in designing and training deep neural network architecture.
[15]	Tacotron model for end-to-end speech synthesis from text.	-Offers a direct mapping from text to spectrogram for speech synthesis. -Can capture prosodic features and intonation in synthesized speech	-May require large amounts of training data for optimal performance. -Performance may vary depending on the complexity of accents.
[16]	neural voice cloning with one sample.	-Enable voice cloning with minimal data requirements (one sample per speaker). -Offers real-time performance for interactive applications. -Can capture speaker characteristics and intonation effectively.	-May struggle with capturing speaker variability and diversity with limited data. -Performance may degrade for speakers with significantly different voice characteristics from the training data.

[17]	neural voice cloning for low-resources languages.	-Addresses the challenges of voice cloning for languages with limited data. -Offers real-time performance for interactive voice applications. -Can adapt to new languages and speakers with minimal data requirements.	-Performance may degrade for language with complex phonological structures or dialectal variations. -Limited availability or labeled data may hinder model training and performance
[18]	Real-time speech emotion cloning.	-Enable real-time synthesis of speech with specified emotional characteristics. -Offers potential for interactive applications	-Complexity in modeling and synthesizing emotional speech characteristics. -Performance may vary depending on the diversity
[19]	neural text-to- speech synthesis using deep voice architecture.	-Enable efficient and real- time synthesis of neural- sounding speech. -Can handle long-form text inputs with minimal latency. -Offers flexibility in controlling speech characteristics through model architecture.	-Requires significant computational resources, especially during training. -Complexity in designing and training deep neural network architecture.

Table 1: Comparision Table

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Function and non-functional requirements

Requirement's analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional and non-functional requirements.

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

Non-functional requirements: These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.

They basically deal with issues like:

- Portability
- Security
- Maintainability
- Reliability
- Scalability

- Performance
- Reusability
- Flexibility

Examples of non-functional requirements:

1)The processing of each request should be done within 10 seconds

2)The site should load in 3 seconds whenever of simultaneous users are > 10000

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

4.3 Architecture

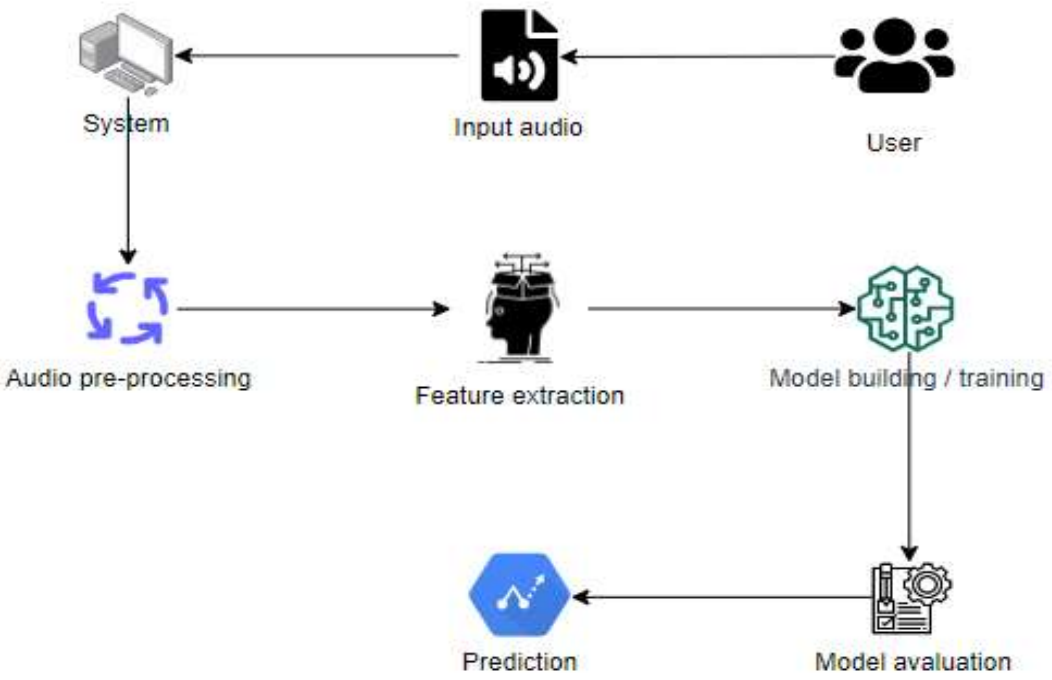


Fig2: Architecture.

CHAPTER 5

SYSTEM DESIGN

5.1 Introduction To Input Design:

In an information system, input is the raw data that is processed to produce output. During input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Well-designed input forms and screens have following properties −

- It should serve specific purpose effectively such as storing, recording, and retrieving the information.
- It ensures proper completion with accuracy.
- It should be easy to fill and straightforward.
- It should focus on user's attention, consistency, and simplicity.
- All these objectives are obtained using the knowledge of basic design principles regarding −
    - What are the inputs needed for the system?
    - How end users respond to different elements of forms and screens.

Objectives for Input Design:

The objectives of input design are −

- To design data entry and input procedures
- To reduce input volume
- To design source documents for data capture or devise other data capture methods

- To design input data records, data entry screens, user interface screens, etc.
- To use validation checks and develop effective input controls.

Output Design:

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

Objectives of Output Design:

The objectives of input design are:

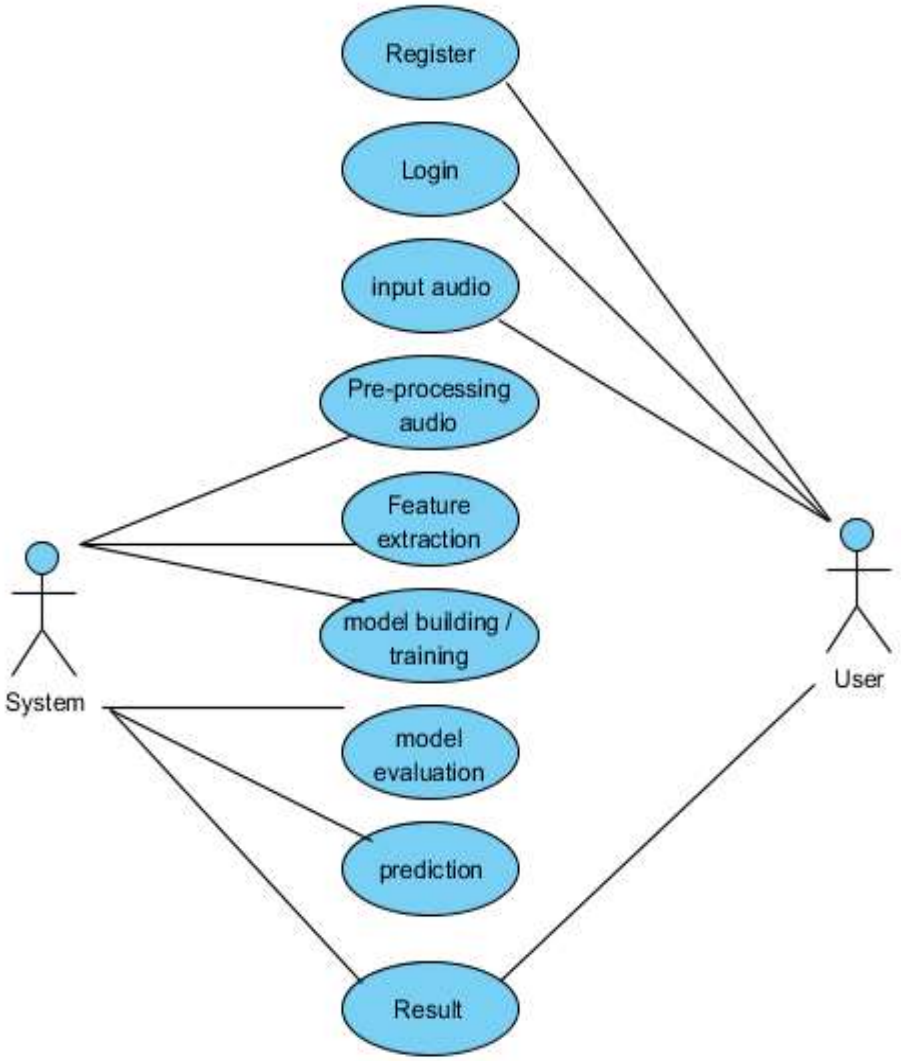
- To develop output design that serves the intended purpose and eliminates the production of unwanted output.
- To develop the output design that meets the end user's requirements.
- To deliver the appropriate quantity of output.
- To form the output in appropriate format and direct it to the right person.
- To make the output available on time for making good decisions.

•

5.1UML Diagrams:

5.1.1Use Case Diagram:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.



. **Fig 3 : UML Diagram**

5.1.1 Class Diagram

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

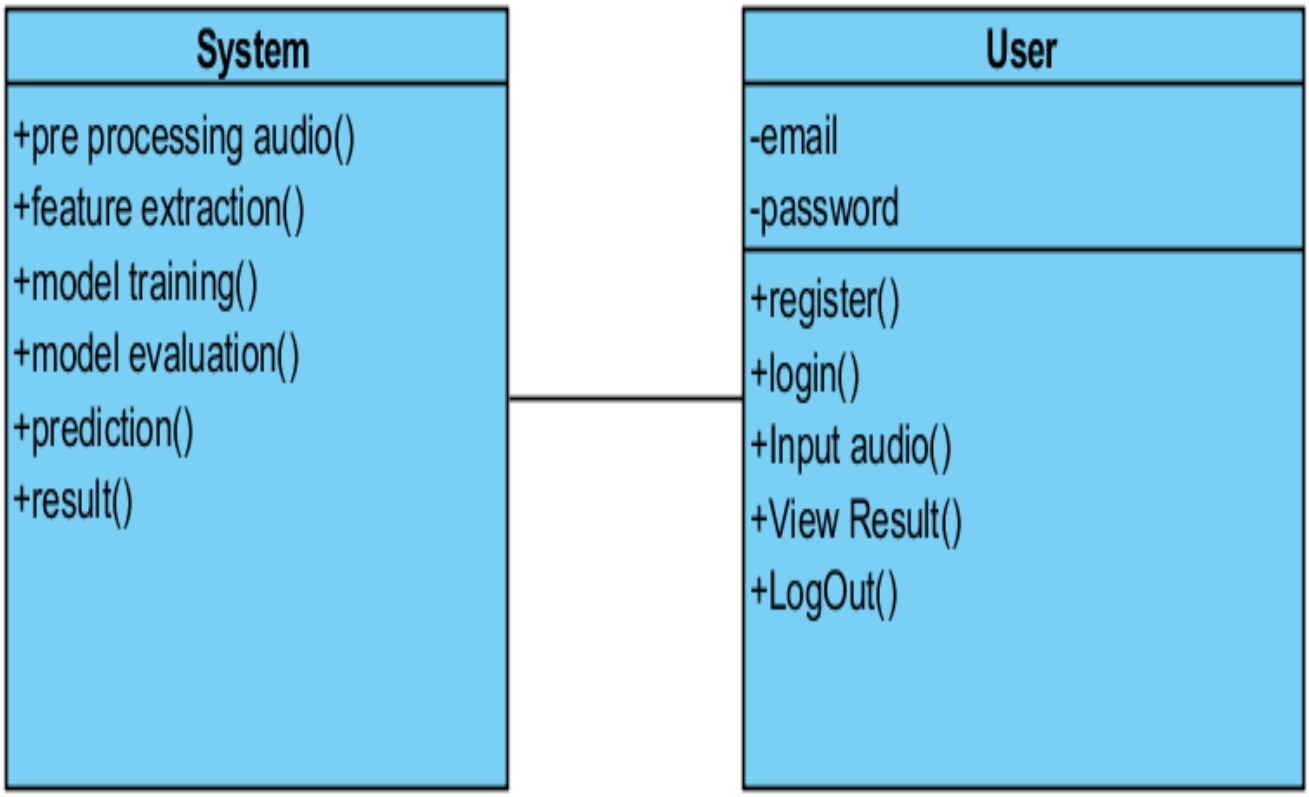


Fig 4 : Class Diagram

5.1.2 Sequence Diagram

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

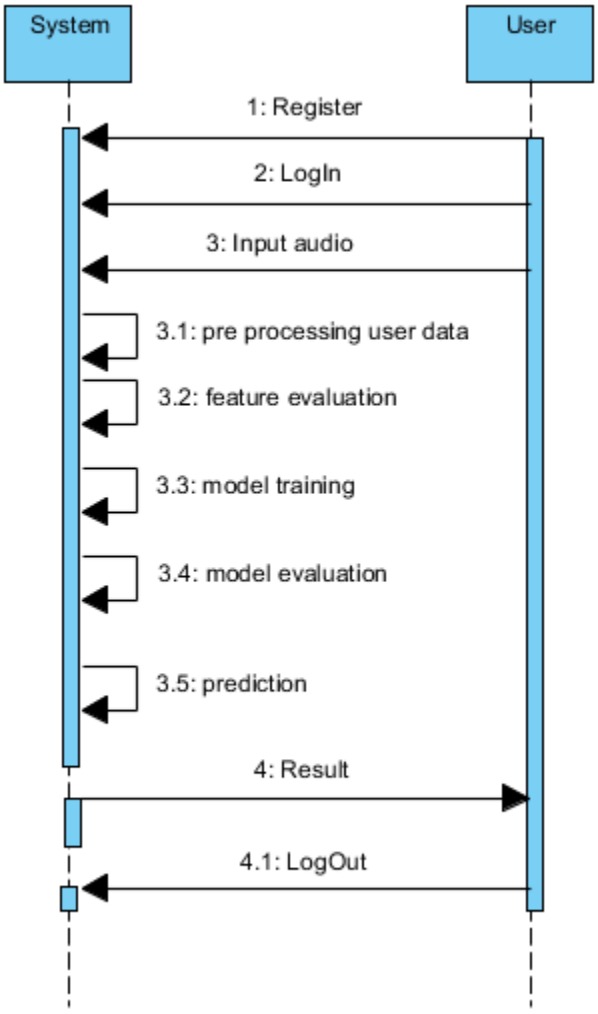


Fig 5 : Sequence Diagram

5.1.3 Collaboration Diagram

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

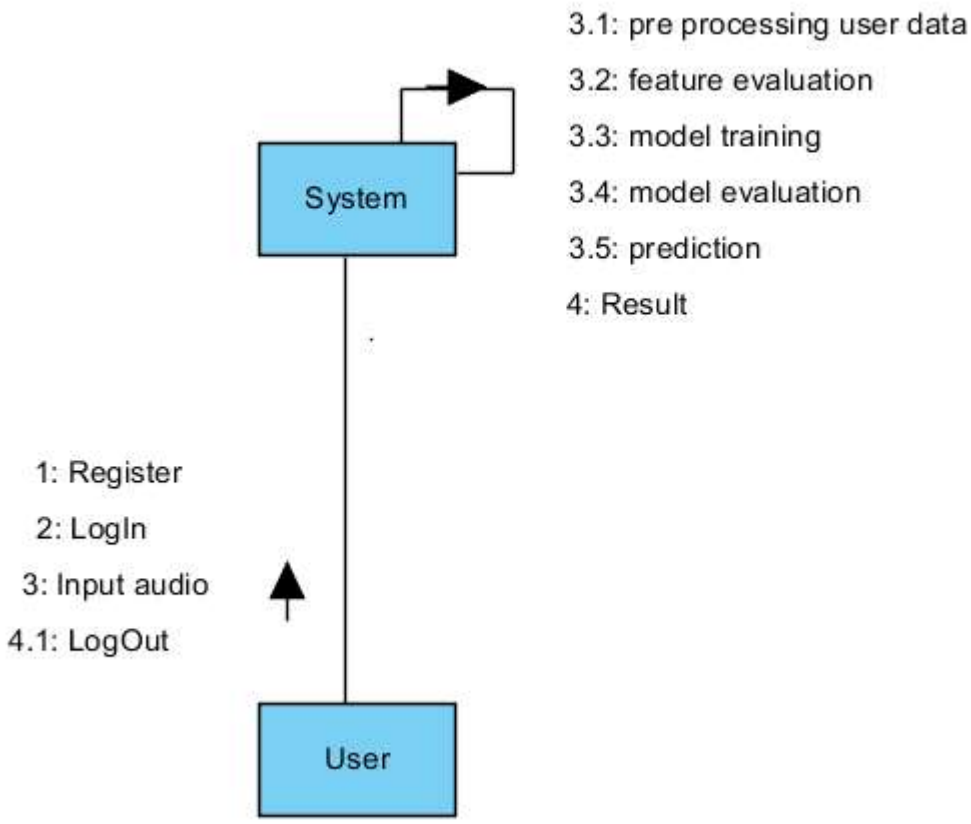


Fig 6 : Collaboration Diagram

5.1.4 Deployment Diagram

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.

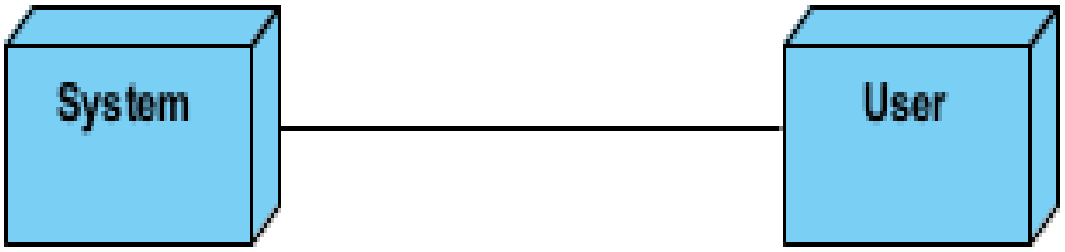


Fig 7: Deployment Diagram

5.1.5Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

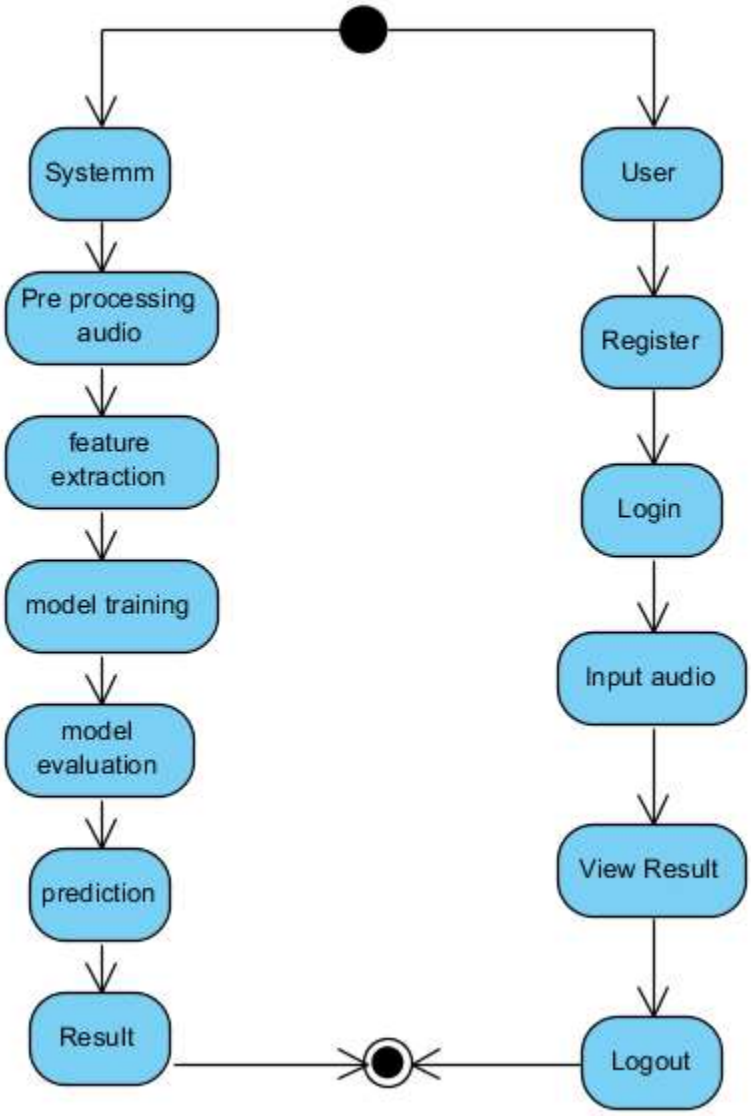


Fig 8 :Activity Diagram

5.1.5 Component Diagram

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical **c**omponents in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.

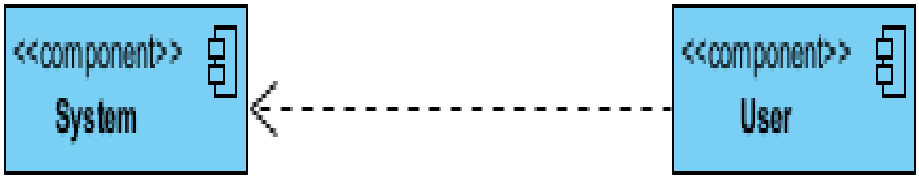


Fig 9 : Component Diagram

5.1.6 ER Diagram

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of

a database. Let’s have a look at a simple ER diagram to understand this concept.

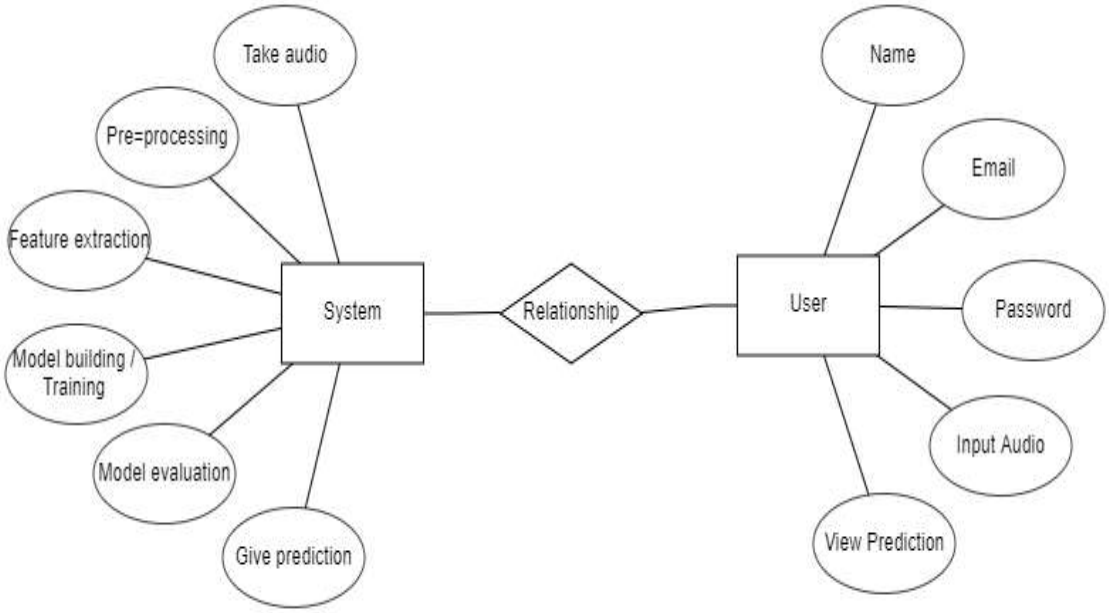


Fig 10 : ER Diagram

CHAPTER 6

IMPLEMENTATION

6.1Modules:

1.User Module:

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.

1.System Module:

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.

6.2 Algorithm:

- **WaveNetModel:**

  WaveNet, a groundbreaking deep generative model developed by DeepMind in 2016, revolutionizes audio synthesis with its innovative architecture and autoregressive training approach. Built upon a stack of dilated causal convolutional layers, WaveNet captures intricate patterns in audio waveforms by leveraging the power of deep learning. The dilated convolutions enable exponentially expanding receptive fields, allowing the model to grasp long-range dependencies efficiently while maintaining a

manageable number of parameters. Residual connections between layers facilitate training of deep networks, mitigating gradient vanishing issues. Gated activation units within each layer regulate information flow, ensuring fine-grained control over generated samples. During training, WaveNet predicts each audio sample conditioned on preceding ones, capturing temporal dependencies effectively. With its ability to produce high-fidelity, natural-sounding audio, WaveNet has become a cornerstone in various applications, including speech synthesis, music generation, and sound effects synthesis.

- **LSTM Model:**

Long Short-Term Memory (LSTM) models represent a significant advancement in recurrent neural networks (RNNs), offering a robust solution for handling sequential data with long-range dependencies. Unlike traditional RNNs, LSTMs incorporate memory cells

- **RNN Model:**

Recurrent Neural Networks (RNNs) are a class of neural networks uniquely suited for handling sequential data due to their ability to maintain a memory of past inputs. Unlike traditional feedforward networks, RNNs incorporate loops within their architecture, allowing information to persist over time. At each time step, an RNN takes an input and updates its hidden state based on both the current input and the previous hidden state. This hidden state serves as the network's memory, enabling it to capture dependencies and patterns in sequential data. RNNs are trained using backpropagation through time (BPTT), where gradients are computed and updated recursively over the entire sequence. However, RNNs often struggle with capturing long-range dependencies due to the vanishing or exploding gradient problem. To address this issue, more advanced variants such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs) have been developed, incorporating gating mechanisms to better regulate the flow of information. Despite their challenges, RNNs remain valuable tools in a wide range of applications, including natural language processing, time series analysis, and sequential data generation.

- **CNN Model:**

A Convolutional Neural Network (CNN) is a deep learning architecture designed specifically for processing and analyzing grid-like data, such as images. Consisting of convolutional layers, pooling layers, and fully connected layers, CNNs excel in tasks like image classification, object detection, and image segmentation. Convolutional layers extract local patterns from input images through sliding filters, capturing features like edges and textures. Subsequent pooling layers reduce spatial dimensions while retaining important information, aiding computational efficiency and preventing overfitting. Activation functions introduce non-linearity, enabling the network to learn complex relationships between features. Fully connected layers make high-level decisions based on learned features, culminating in the final output. Trained through optimization algorithms like stochastic gradient descent, CNNs minimize a loss function by iteratively adjusting parameters to match predicted outputs with ground truth labels. Regularization techniques like dropout and batch normalization help prevent overfitting, while transfer learning leverages pre-trained models to improve performance, particularly in scenarios with limited data. With their ability to automatically learn and extract meaningful features from raw data, CNNs have become indispensable in various fields, driving advancements in computer vision and pattern recognition tasks.

CHAPTER 7

SYSTEM STUDY AND TESTING

7.1 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 considerations involved in the feasibility analysis are

- Economical feasibility
- Technical feasibility
- Social feasibility

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.

Socio 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.

7.2 Types of Tests

7.2.1 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.

7.2.2 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.

7.2.3 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 exercised.

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.

7.2.4 White Box Testing

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

7.2.5 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, 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.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

- **TEST CASES:**

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

Table 2 : Test Cases

- **Test cases Model building:**

S.NO	Test Cases	I/O	Expected O/T	Actual O/T	P/F
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 inputed audio by using given algorithms.	Prediction as output	Prediction as output	It produced Real. If this is not, it will undergo Fake.

Table 3 : Test Cases Model Building

CHAPTER 8

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 retrivequery1(query,values):
    mycursor.execute(query,values)
    data = mycursor.fetchall()
    return data

def retrivequery2(query):
    mycursor.execute(query)
    data = mycursor.fetchall()
    return data

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



@app.route('/register', methods=["GET", "POST"])
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 = retrivequery2(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 = retrivequery2(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 = retrivequery1(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)
```

CHAPTER 9

RESULTS AND DISCUSSION

Home page:

Fig 10 : Home Page.

Fig 11 :About Page

Login:

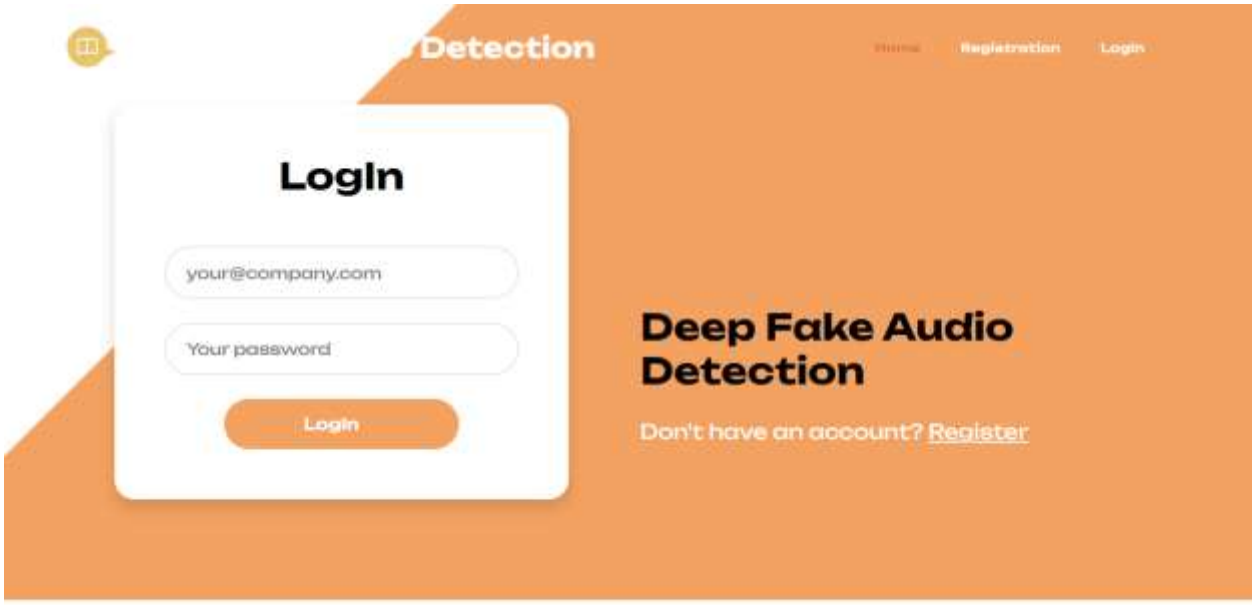


Fig 12 : Login Page.

Registration:

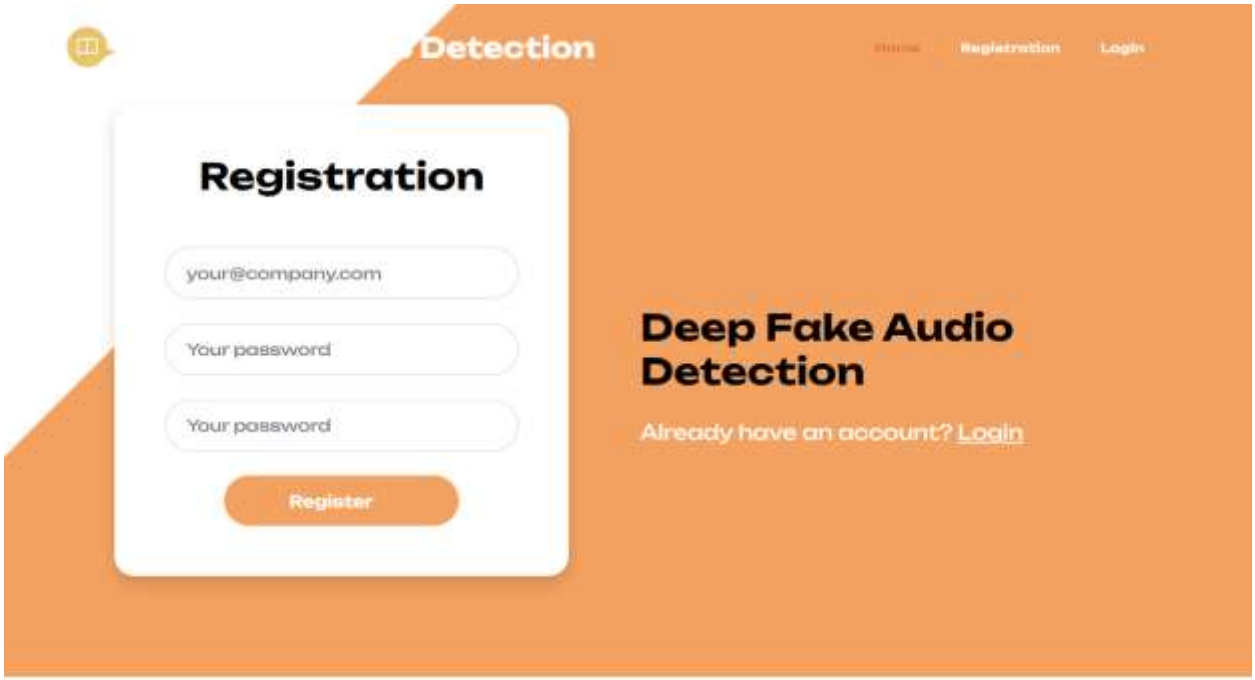


Fig 13 : Registration Page.

Upload:

Fig 14: Upload Page.

Result Page:

Fig 15 : Prediction of Fake

Fig 16:Prediction Page of Real

CHAPTER 10

CONCLUSION AND FUTURE ENHANCEMENT

10.1 Conclusion:

In conclusion, our 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.

10.2 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 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

REFERENCES

[1] "Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., ... & Shor, J. (2017). Tacotron: Towards end-to-end speech synthesis. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 3381- 3390). JMLR. org."

[2] "Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., ... & Wu, Y. (2018). Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4779-4783). IEEE."

[3] "Ren, Y., & Kang, S. H. (2020). Towards End-to-End Real-Time Voice Cloning: A Survey. arXiv preprint arXiv:2006.00900."

[4] "Jia, Y., Zhang, Y., & Dai, L. R. (2018). Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In 2018 IEEE Spoken Language Technology Workshop (SLT) (pp. 906-913). IEEE."

[5] "Taigman, Y., Polyak, A., & Wolf, L. (2017). Unsupervised cross-domain image generation. arXiv preprint arXiv:1611.02200."

[6] "Wang, Y., Skerry-Ryan, R., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., ... & Chen, Y. (2018). Tacotron 2: Towards end-to-end speech synthesis. arXiv preprint arXiv:1712.05884."

[7] "Wan, L., Wang, Q., Papir, A., Lee, K., & Subakan, C. (2019). Generalized end- to-end loss for speaker verification. In 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6555-6559). IEEE."

[8] "Yang, Z., Hu, Z., & Liang, Q. (2020). Voice cloning technology based on adaptive learning and reinforcement learning. IEEE Access, 8, 14630-14642."

[9] "Gibiansky, A., & Synnaeve, G. (2017). Deep voice 2: Multi-speaker neural text- to-speech. arXiv preprint arXiv:1705.08947."

[10] "Van Den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... & Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499."

[11] "Liao, Z., Zhang, S., Xie, L., & Zhang, D. (2019). Real-time neural voice cloning using micro-service architecture. In 2019 IEEE International Conference on Multimedia and Expo (ICME) (pp. 737-742). IEEE."

[12] "Wang, J., Hu, X., Zhang, Z., Jiang, L., & Yang, J. (2020). Real-time voice cloning system based on generative adversarial network and voice conversion. IEEE Access, 8, 193905-193914."

[13] "Lee, J. H., Jung, S. H., & Lee, S. H. (2021). Voice cloning based on time- domain generative adversarial networks with auxiliary classification loss. IEEE Access, 9, 68839-68850."

[14] "Kang, J., Kim, J., & Lee, D. (2019). Real-time voice cloning based on high- efficiency parallel wavegan. arXiv preprint arXiv:1910.10897."

[15] "Erdogan, H., Hori, T., & Hershey, J. R. (2015). Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. In Sixteenth Annual Conference of the International Speech Communication Association."

[16] "Wang, Y., Wu, Y., Liu, Y., Prabhavalkar, R., Chen, Z., & He, Z. (2018). Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. arXiv preprint arXiv:1803.09017."

[17] "Nachmani, E., Wolf, L., & Zohar, A. (2018). Improving sequence to sequence learning for non-autoregressive neural machine translation. arXiv preprint arXiv:1804.10201."

[18] "Hubert, T., Cafarelli, A., Pham, Q. T., Nidadavolu, S. S., Tygert, M., Mai, G., ... & Vincent, E. (2020). An updated Wavenet for real-time synthesis of natural- sounding speech with minimal artifacts. arXiv preprint arXiv:2006.03575."

[19] "Saito, M., & Nakamura, S. (2017). Training feed-forward sequential memory networks with attention for voice conversion. arXiv preprint arXiv:1711.01261."

[20] "Lample, G., & Ott, M. (2019). Cross-lingual language model pretraining. arXiv preprint arXiv:1901.07291."

[21] "Furlanello, T., Lipton, Z. C., Tschannen, M., Cox, D., & Vaswani, A. (2018). Born again neural networks. arXiv preprint arXiv:1805.04770."

[22] "Hou, L., Li, F., & Tao, J. (2020). Unsupervised many-to-many non-parallel voice conversion with variational autoencoders. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 28, 1092-1103."

[23] Arik, Sercan O., et al. "Deep voice: Real-time neural text-to-speech." Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org,(2017).

[24] van den Oord, Aäron, et al. "WaveNet: A generative model for raw audio." arXiv preprint arXiv:1609.03499 (2016).

[25] Wang, Yuxuan, et al. "Tacotron: Towards end-to-end speech synthesis." arXiv preprint arXiv:1703.10135 (2017).

[26] Qian, Junyan, et al. "AutoVC: Zero-shot voice style transfer with only autoencoder loss." arXiv preprint arXiv:1905.05879 (2019).

[27] Jia, Ye, Yu Zhang, and Ron J. Weiss. "Real-time neural voice cloning with one sample." arXiv preprint arXiv:1910.08996 (2019).

[28] Jia, Ye, et al. "Real-time neural voice cloning for low-resource languages." arXiv preprint arXiv:2004.07948 (2020).

[29] Lee, Dong-Yun, et al. "Real-time speech emotion cloning." arXiv preprint arXiv:2011.02679 (2020).

[30] Jeong, Sungkwon, et al. "Real-time voice conversion using vector-quantized variational autoencoder with a self-conditioning mechanism." arXiv preprint arXiv:2107.06200 (2021).