

A Major Project Report

On

**“VoiceChat - Bringing chats to life
using Deep Learning”**

Submitted in partial fulfillment of the

Requirements for the award of the degree of

Bachelor of Technology

In

**Computer Science & Engineering-
Artificial Intelligence & Machine Learning**

By

Laxman Vikas Kommireddi

20R21A6622

Yogeshwar Rayudu

20R21A6644

Pavan Kalyan Aouti

21R25A6601

Mohammed Ikramuddin

20R21A6637

Under the guidance of

Mr. V S.Pavan kumar
Associate professor

Department of Computer Science & Engineering



MLR

INSTITUTE OF TECHNOLOGY

(UGC AUTONOMOUS)

Affiliated to JNTUH, Approved by AICTE
Laxman Reddy Avenue, Dundigal, Hyderabad-500 043, Telangana, India



2024

Department of CSE-AIML

CERTIFICATE

This is to certify that the project entitled “**VoiceChat - Bringing chats to life using Deep Learning**” has been submitted by **Laxman Vikas Kommireddi (20R21A6622)**, **Yogeshwar Rayudu (20R21A6644)**, **PavanKalyan Aouti (21R25A6601)**, **Mohammed Ikramuddin (20R21A6637)** in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering from Jawaharlal Nehru Technological University, Hyderabad. The results embodied in this project have not been submitted to any other University or Institution for the award of any degree or diploma.

Internal Guide

Head of the Department

Project coordinator

External Examiner

Department of CSE-AIML

DECLARATION

We hereby declare that the project entitled “**VoiceChat - Bringing chats to life using Deep Learning**” is the work done during the period from **January 2024 to May 2024** and is submitted in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering from Jawaharlal Nehru Technology University, Hyderabad. The results embodied in this project have not been submitted to any other university or Institution for the award of any degree or diploma.

Laxman Vikas Kommireddi
Yogeshwar Rayudu
Pavan Kalyan Aouti
Mohammed Ikramuddin

20R21A6622
20R21A6644
21R25A6601
20R21A6637

Department of CSE-AIML

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we now have the opportunity to express our guidance for all of them.

First of all, we would like to express my deep gratitude towards our internal guide **Mr. V S.PAVAN KUMAR, Associate professor, Department of CSE -AIML** for his support in the completion of our dissertation. We wish to express our sincere thanks to **Dr. K. SAI PRASAD, HOD, Dept. of CSE-AIML** and principal **Dr. K. SRINIVAS RAO** for providing the facilities to complete the dissertation.

We would like to thank all our faculty and friends for their help and constructive criticism during the project period. Finally, We are very much indebted to our parents for their moral support and encouragement to achieve goals.

Laxman Vikas Kommireddi
Yogeshwar Rayudu
Pavan Kalyan Aouti
Mohammed Ikramuddin

20R21A6622
20R21A6644
21R25A6601
20R21A6637

Department of CSE-AIML

ABSTRACT

This project describes a revolutionary chat application that leverages efficient voice cloning technology to revolutionise digital communication. With the help of this invention text messages can be now personalised and sent as audio messages with the sender's voice. The proposed system captures the authenticity of the sender's speech by smoothly integrating AI algorithms to mimic vocal patterns and nuances. This study develops a trustworthy and adaptable voice replication system by implementing developments in neural networks and speech processing. Using a combination of deep neural networks and chat application development, the project aims to analyse and capture detailed human vocal patterns. The algorithm extracts the subject's vocal characteristics, intonations and speech patterns using as few human voice samples as possible. This method enhances communication quality in general and emotional expression. The application aims to combine application development and deep learning with advanced options, pushing the boundaries of technical innovation. This synopsis focuses on the creation of bridging the text-voice divide, enhancing the accessibility, personalisation, and engagement of digital communication.

APPENDIX-1

TABLE OF CONTENT

INDEX

Certificate	i
Declaration	ii
Acknowledgement	iii
Abstract	iv
Table of Content	
List of Figures	
List of Tables	
List of Abbreviations	
References	
Chapter 1	
Introduction	1
1.1 Overview	1
1.2 Purpose of the project	1
1.3 Motivation	2
Chapter 2	
Literature Survey	3
Chapter 3	
Proposed System	
3.1 Proposed System	93
3.2 Advantages of Proposed System	93
3.3 System Requirements	94
3.3.1 Software Requirements	94
3.3.2 Hardware Requirements	95
3.3.3 Implementation Technologies	96

Chapter 4	
System Design	98
4.1 Proposed System Architecture	98
4.2 UML Diagrams	98
4.2.1 Use case Diagrams	100
4.2.2 Sequence Diagram	101
4.2.3 Activity Diagram	101
Chapter 5	110
5.1 Implementation with Hypothetical Scenarios	110
5.1.1 Working of Tortoise	111
5.2 Source Code	113
Chapter 6	133
Results	133
Chapter 7	134
Conclusion	134
Future Enhancements and Discussions	135

APPENDIX-2

LIST OF FIGURES

LIST OF FIGURES

Fig No	Description of Figure	Page No
Figure 1	Expressive voice cloning model	12
Figure 2	System diagram (a) training procedure and (b) inference procedure with inputs.	20
Figure 3	Working of SparsevWaveRNN	29
Figure 4	Model architecture	33
	Outline of statistical parametric speech	
Figure 5	synthesis	37
Figure 6	Deep voice 3	41
Figure 7	Certificate Generation and Registration	49
Figure 8	Text to speech synthesizer	52
Figure 9	Flowchart of phoneme based text to speech	55
Figure 10	System block diagram	58
	The meta-network generates parameters of	
Figure 11	language-specific convolutional text encoders.	61
Figure 12	Relationship between clients	63
Figure 13	A multi speaker text to speech synthesis	70
	Text to speech conversion with emotion	
Figure 14	detection	73
Figure 15	Providing security using end to end encryption	76
Figure 16	Voice generation using deep learning	80
Figure 17	Process of tortoise	99
Figure 18	Process of text to speech synthesis	104
Figure 19	Function of application	106
Figure 20	Detailed description of conversion of TTS	107
Figure 21	Performance metrics of TTS algorithm	133
Figure 22	Live implementation of voicechat application	134

APPENDIX-3
LIST OF TABLES

LIST OF TABLE

Table No	Description of Figure	Page No
Table. 2.2	Comparision table	81
Table. 2.3	Working of Evaluation Table	85

APPENDIX-4

LIST OF ABBREVIATIONS

ABBREVIATIONS

TTS	Text to Speech
RNN	Recurrent neural networks
CNN	Convolutional neural network
WER	Word error rate
MOS	Mean opinion score

APPENDIX-5

REFERENCES

References

- [1].Paarth Neekhara Shehzeen Hussain, Shlomo Dubnov, Farinaz Koushanfar, Julian McAuley, "Expressive Neural Voice Cloning" Feb 2, 2021.
- [2].Sercan Ö. Arik , "Neural Voice Cloning with a Few Samples", 12 Oct 2018.
- [3].Mike Chrzanowski, "Deep Voice: Real-time Neural Text-to-Speech", 2017.
- [4].Akshata D Vhandale, "AN OVERVIEW OF REAL-TIME CHAT APPLICATION", 2022.
- [5].Daria Diatlova, "EmoSpeech: Guiding FastSpeech2 Towards Emotional Text to Speech", 2023.
- [6].Kainan Peng, Andrew Gibiansky, Sercan O. Arik Ajay Kannan, Sharan Narang,Wei Ping, "DEEP VOICE 3: SCALING TEXT-TO-SPEECH WITH CONVOLUTIONAL SEQUENCE LEARNING" , 22 February 2018.
- [7].Aaron van den Oord, Sander Dieleman, Heiga Zen , Karen Simonyan Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu,"WAVENET: A GENERATIVE MODEL FOR RAW AUDIO" 19 september 2016.
- [8].Nal Kalchbrenner, Erich Elsen ,Karen Simonyan ,Seb Noury, Norman Casagrande ,Edward Lockhart ,Florian Stimberg , Aaron van den Oord, Sander Dieleman, Koray Kavukcuoglu,"Efficient Neural Audio Synthesis"25 june 2018.
- [9].Yuxuan Wang , RJ Skerry-Ryan , Daisy Stanton, Yonghui Wu, Ron J. Weiss , Navdeep Jaitly, Zongheng Yang, Ying Xian , Zhifeng Chen, Samy Bengio , Quoc Le,

- [10]. Yannis Agiomyrgiannakis, Rob Clark, Rif A. Saurous, "TACOTRON: TOWARDS END-TO-END SPEECH SYNTHESIS " 6 april 2017.
- [11]. Raman Singh, Ark NandanSingh Chauhan, Hitesh Tewari , "Blockchain-enabled End-to-End Encryption for Instant Messaging Applications", june 2022
- [12]. Hruthik B Gowda, Karun Datta Ramakumar, Sheethal.V, Sushma M, Dr. Madhusudhan G K "REAL-TIME VOICE CLONING USING DEEP LEARNING: A CASE STUDY"
- [13]. Fahima Khanam , Farha Akhter Munmun , Nadia Afrin Ritu, Alope Kumar Saha , "Text To Speech Conversion Using Different Speech Synthesis"
- [14]. Dr.T NAnitha, Amilio Dsouza, Ashutosh , Akshay Gole , " REAL TIME VOICE CLONING"
- [15]. Tomáš Nekvinda, Ondřej Dušek, "One Model, Many Languages: Meta-learning for Multilingual Text-to-Speech"
- [16]. Dr. Abhay Kasetwar, Ritik Gajbhiye, Gopal Papewar, Rohan Nikhare, Priya Warade , "Development of chat application"
- [17]. Sakith Nalluri, A.Rohan Sai, and M.Saraswati, "Real Time Voice Cloning," vol. 7, pp. 297–302, April 2021
- [18]. Giuseppe Ruggiero, Enrico Zovato, Luigi Di Caro, Vincent Pollet, "Voice Cloning: A Multi-Speaker Text-To-Speech Synthesis Approach Based On Transfer Learning" , vol. 1 , Feb 2021.
- [19]. Anita and Srinivasan, "Text to Speech Conversion with Emotion Detection , " vol. 13, 14, 2018, pp. 11512-11517
- [20]. Noor Sabah, Jamal M. Kadhim and Ban N. Dhannoon, " Developing an End-to-End Secure Chat Application ", IJCSNS International Journal of Computer Science and Network Security, Vol.17 No.11, November 2017.

- [21]. Gonzalo Gómez Sánchez, “ Voice Generation Using Deep Learning,” September 28, 2016
- [22]. Sercan O. Arik, Gregory Diamos, Andrew Gibiansky, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou, “Deep Voice 2: Multi-speaker neural text-to-speech”. In NIPS, 2017b.
- [23]. Mehri, S., Kumar, K., Gulrajani, I., Kumar, R., Jain, S., Sotelo, J., Courville, A. C., and Bengio, Y, “SampleRNN: an unconditional end-to-end neural audio generation model”. CoRR, abs/1612.07837, 2016.
- [24]. Emiliano Masi , Giovanni Cantone , Giuseppe Calavaro, "Mobile Apps Development: A Framework for Technology Decision Making", October 2012.
- [25]. R M Ali¹, S N Alsaad¹, "Instant messaging security and privacy secure instant messenger design", 2020.
- [26]. Jemine, Corentin,” Automatic Multispeaker Voice Cloning”,2019
- [27]. Hema A Murthy, ”Building Speech Synthesis Systems for Indian Languages”, 2017.
- [28]. Y. Lee, A. Rabiee, and S. Lee, “Emotional end-to-end neural speech synthesizer,” CoRR, vol. abs/1711.05447, 2017.
- [29]. Masiello Eric, author, “Mastering React Native. January 11”, 2017. Accessed 1 Jan 2022
- [30]. Naimul Islam Naim, ReactJS: An Open-Source JavaScript library for front-end development, Metropolia University of Applied Sciences, accessed on 1 Jan 2022
- [31]. Stefanov Stoyan, editor. React: Up and Running: Building web Applications

INDEX

Certificate	i
Declaration	ii
Acknowledgement	iii
Abstract	iv
List of Figures and Tables	
Chapter 1	
Introduction	1
1.2 Overview	1
1.2 Purpose of the project	1
1.3 Motivation	2
Chapter 2	
Literature Survey	3
2.1 Existing system	3
2.2 comparision table	77
2.3 Work evaluation table	81
2.4 Disadvantages of Existing System	91
Chapter 3	
Proposed System	93
3.1 Proposed System	93
3.2 Advantages of Proposed System	93
3.3 System Requirements	94
3.3.1 Software Requirements	94
3.3.2 Hardware Requirements	95
3.3.3 Implementation Technologies	96
Chapter 4	
System Design	98
4.1 Proposed System Architecture	98
4.2 UML Diagrams	98
4.2.1 Use case diagram	99

4.2.2 Sequence diagram	100
4.2.3 Activity diagram	101
Chapter 5	
Implementation	110
5.1 Implementation with Hypothetical Scenarios	110
5.1.working Tortoise	111
5.2 Source Code	113
Chapter 6	133
Results	133
Chapter 7	134
Conclusion	134
Future Enhancements and Discussions	135

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The "VoiceChat" project is a revolutionary chat application that utilizes efficient voice cloning technology to transform digital communication. It allows users to personalize text messages and send them as audio messages. The project leverages developments in neural networks and speech processing to develop a trustworthy and adaptable voice replication system. By combining deep neural networks and chat application development.

The project aims to address the limitations of traditional text messaging by infusing conversations with Speech generation of text message. It promotes accessibility for visually impaired individuals and aligns with the evolving landscape of digital communication. The proposed solution is designed to bridge the gap between text and voice, enhancing digital communication by adding a more personal and accessible dimension.

1.2 PURPOSE OF THE PROJECT

The Purpose of the "VoiceChat" initiative is to transform digital communication through the utilization of advanced voice cloning technology. The project seeks to create a messaging platform where users can transmit text messages in the form of audio messages. This groundbreaking concept closes the divide between text-based and voice-based communication, enriching digital interactions with a more intimate and user-friendly experience. The project aims to overcome the constraints of traditional text messaging. Furthermore, it promotes inclusivity by catering to the needs of visually impaired individuals and adapts to the changing dynamics of digital communication.

1.3 MOTIVATION

The goal is to close the divide between text and voice communication, enriching digital interaction with a more intimate and user-friendly approach. This objective is aimed at tackling issues such as limited accessibility for visually impaired individuals and seniors, challenges with multitasking, difficulties in noisy surroundings, and the inconvenience of reading lengthy messages. Through the utilization of efficient voice cloning technology, the project seeks to transform digital communication by enabling the personalization of text messages into audio format. This innovation promises to render communication more authentic and captivating, revolutionizing the way we connect online

CHAPTER 2

LITERATURE SURVEY

An extensive literature survey has been conducted by studying existing systems of Text to speech models and chat applications. A good number of research papers, journals, and publications have also been referred to before formulating this survey.

2.1 EXISTING SYSTEM

Paarth Neekhara [1] aims to address the limitation of current voice cloning methods that cannot control the expressiveness of synthesized audio. This solution is an attempt to synthesize the voice of a speaker with fine control over various style aspects of speech. The system can generate speech that is not only accurate but also expressive, capturing variations in tone, speaking rate, emphasis, and emotions. This capability has several applications, such as voice-overs in animated films, synthesizing realistic and expressive speech for videos.

Sercan Ö. Arık [2] discusses the solution for the problem of voice cloning, which involves learning the voice of a speaker from a limited amount of data and generating speech that sounds like it is pronounced by the target speaker. The goal of this system is to synthesize a person's voice from only a few audio samples. This solution uses a technique called 'Speaker adaptation' that allows fine-tuning a pre-trained [21] multi-speaker model for an unseen speaker.

Mike Chrzanowski [3] They developed a text-to-speech (TTS) system that solves the problem of generating high-quality and natural-sounding speech from text in real-time. It focuses on optimizing the different components of a TTS system. The authors built a fully neural system that can generate speech in real-time, offering adjustable trade-off between synthesis speed and audio quality. They optimized the inference process to achieve faster-than-real-time speeds, making the system usable for various applications. One of the key contributions of this work is the development of efficient WaveNet inference.

Akshata D Vhandale [4] They introduced a solution that enables users to communicate with each other seamlessly. The chat application provides a platform for users to connect and exchange messages in real-time, regardless of their location. It is designed to facilitate instant messaging and improve communication among users. They used MongoDB, Express.js, [28] React and node.js to build the chat application. This work contributes to the field of [30] web development by showcasing the capabilities of the MERN stack in building real-time chat applications. This solution promotes user friendliness, secure communication, instant communication and scalability.

Daria Diatlova [5] The authors of EmoSpeech proposed a solution that aims to solve the problem of synthesizing speech with desired emotions. The EmoSpeech model extends the

FastSpeech2 architecture with modifications that enable conditioning on a given emotion while maintaining fast inference speed. This solution involves extracting phonemes and punctuation from text using the grapheme-to-phoneme (GTP) model. It also includes extracting durations of phonemes and pitch from waveforms. They also introduced a style control mechanism that allows users to specify the desired emotion in the synthesized speech.

Sercan Ö. Arık [6] The aim of Deep Voice 3 is to create a text-to-speech (TTS) system that is both effective and of good quality. It seeks to find a solution to the challenge of generating coherent, natural voice from textual input. By employing an attention-based sequence-to-sequence model that prevents frequent attention problems and enables a more compact architecture, Deep Voice 3 overcomes the drawbacks of earlier TTS systems. In order to make TTS feasible for production systems, it also focuses on enhancing inference efficiency and training speed. Deep Voice 3 contributes to the advancement of TTS technology by addressing common errors, providing flexibility in waveform synthesis, supporting multispeaker synthesis, and delivering superior audio quality.

Aaron van den Oord [7] Used a deep neural network that aims to produce unprocessed audio waveforms. It attempts to address the issue of producing audio that sounds natural and of excellent quality, including music and conversation. As a result of WaveNet's fully probabilistic and autoregressive architecture, every audio sample's prediction is dependent on every other sample. Because of this, WaveNet is able to produce realistic audio with natural intonation while also capturing the traits of many speakers.

Nal Kalchbrenner [8] used the sparse waveRNN method that resulted in reducing the computing demands for audio synthesis without sacrificing sound quality. It attempts to address the issues of enabling audio synthesis on low-power mobile CPUs and achieving real-time or faster audio synthesis on GPUs.

Yuxuan Wang [9] The Tacotron approach aims to provide a generative text-to-speech (TTS) model that can generate speech from characters in an end-to-end fashion. It solves the complexity and monotony of traditional TTS pipelines, which often need extensive domain expertise and multiple phases. Tacotron uses many models trained on text-audio pairs and rich conditioning on multiple attributes to try to automate TTS. As a result, feature engineering is not necessary.

Raman Singh [10] proposed blockchain-based end-to-end encryption (E2EE) system aims to provide instant messaging applications with a true end-to-end encrypted messaging service. Your data is protected and no one can access it except the intended recipient.

Hruthik B Gowda [11] The aim is to create a system that can make someone's voice from written words. The system has different parts, like TTS Synthesis, a WaveNet Vocoder, and Neural Network Training. It copies voices using Speaker Encoding, TTS Synthesis, and the

Vocoder. It can produce good text-to-speech output, useful for things like voice assistants. But, it may not work well if there's noise in the input audio.

Fahima Khanam [12] The main goal is to turn written text into natural and understandable speech using Text-to-Speech (TTS) technology. The system involves Natural Language Processing (NLP) for text analysis, phonetic conversion, and prosodic phrasing, along with Digital Signal Processing (DSP). It combines NLP and DSP, using unit selection speech synthesis to pick the best-sounding units from a speech database. This approach produces natural and clear speech, benefiting people with visual impairments and finding applications in teaching aids, text reading, and interactive media. However, there's a limitation as it may have discontinuities in phoneme transitions, affecting the smoothness and naturalness of the synthesized speech. The end result is a synthesized speech waveform that matches the input text.

Dr.T NAnitha [13] The "Real Time Voice Cloning" project aims to quickly copy a person's voice using a three-stage deep learning system. It includes deep learning models, a vocoder, reference speech, and training data. The system works through Feature Extraction, an Acoustic Model, and a Vocoder, featuring a personalized voice interface. A positive aspect is its ability to clone unheard voices with just a few seconds of reference speech. However, a drawback is that the cloned voice may lack naturalness and accurate replication of accents compared to the original human voice, affecting authenticity. In the end, the system produces a cloned voice similar to the reference audio, but there might be differences in naturalness and accent.

Tomáš Nekvinda [14] This multilingual text-to-speech (TTS) model aims to make natural-sounding speech in many languages with minimal training data. Its parts include an Input Text Encoder, Decoder, Convolutional Encoder, and a Parameter Generator Network. It uses a special Model Architecture that relies on Convolutional Encoders and Encoder Parameter Generation. A unique feature is how it uses multilingual training batches to get the most out of the design. Benefits of this model include improved voice cloning abilities and support for code-switching between languages, making it more adaptable across different languages. However, a downside is that it might not generate speech accurately in complex scripts like Chinese. In the end, the system produces natural-sounding speech in multiple languages, showing how well it achieves its goal.

Dr. Abhay Kasetwar [15] The goal of the chat app project is to create a reliable and adaptable system for instant communication. It uses JavaScript, needs internet connection, has an App Registration Page, and a message editor with a keyboard. This system is great for making communication faster and easier. However, building a chat app is complex and takes time. Also, it relies on an internet connection for use. In the end, it shows messages and lets users send texts successfully.

Sakith Nalluri [16] The aim is to Create an advanced real-time voice cloning framework, enhancing accuracy and realism through deep learning aiding visually impaired or reading-disabled users with help of Front-end and back-end of a text-to-speech system, NLP module and DSP module in the synthesis process. The application has a simple and user-friendly interface, making it easy for users to input text and convert it into speech, take input from user and save user input audio in cloud. This can be beneficial for individuals with visual impairments as it allows them to access and consume large volumes of text more easily.

Giuseppe Ruggiero [17] Advanced Gru Network solution is to build a Text-to-Speech (TTS) system that can generate natural speech for a wide variety of speakers. "Advanced Gru Network" model is a sequence of mel spectrogram frames. It is built with Text-to-speech (TTS) which generate data efficient manner natural speech for a wide variety of speaker and not necessarily seen during the training phase. It consists of 1 Conv1D layer and 3 GRU layers, each followed by a linear projection layer. The advanced GRU network was able to create a robust space of internal features that effectively separated speakers based on their utterances. It achieved the best Speaker Verification Equal Error Rate (SV-EER) on the test set compared to other models.

Anita and Srinivasan [18] The Aim is to create a text-to-speech conversion, analyze the emotions present in textual data which help in understanding the emotional context of the textual data. It ensures proper sentence structure and improves the quality of the speech output. The techniques used involve analyzing sentence patterns, assigning emotion constants to words, and matching words with audio files in the multimedia database based on their assigned emotions. The system components for text to speech conversion are Emotion Detection, Grammar Identification, Text-to Speech Conversion,

Noor Sabah [19] The goal is to solve the problem of security and privacy concerns, To provide [24] end to-end security for users to safely exchange private information without worrying about data leakage. The Text messages, images, or files that they want to send to another user. Provides a secure container to store the local storage key, making it difficult for unauthorized access. It ensure that messages exchanged between users are encrypted and can only be read by the sender and receiver, without the involvement of any third party. Users authenticate themselves by providing their email and password.

Gonzalo Gómez Sánchez [20] The "Voice Generation Using Deep Learning" solution is to develop a system for voice generation using deep learning. It explores deep learning for voice generation, proposing architectures with limitations. Parallelization reduces computational costs, suggesting avenues for future improvements in audio quality and alternative architectures. The audio signal in [27] speech synthesis can lead to improved quality, simplified system architecture, parallelization, and the potential for more advanced text-to-speech systems.

1		
Reference in APA format	Expressive Neural Voice Cloning	
URL of the Reference	Authors Names and Emails	Keywords in this Reference
https://www.researchgate.net/publication/348958580_Expressive_Neural_Voice_Cloning	Paarth Neekhara Shehzeen Hussain, Shlomo Dubnov Farinaz Koushanfar Julian McAuley	Transfer learning, speaker verification, multi-speaker text-to-speech synthesis, Deep Voice 3
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	The Goal (Objective) of this Solution & What is the problem that need to be solved	What are the components of it?

Expressive Neural Voice Cloning	<p>To synthesize the voice of a speaker with fine control over various style aspects of speech.</p> <p>It aims to address the limitation of current voice cloning methods that lack the ability to control the expressiveness of synthesized audio.</p>	<p>Speaker Encoding</p> <p>Style Conditioning</p> <p>Global Style Tokens (GST)</p> <p>Multi-Speaker Text-to-Speech (TTS) Model</p>
The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process		

The "Expressive Neural Voice Cloning" solution aims to synthesize the voice of a speaker using only a few reference audio samples.

The steps involved in the process are as follows:

	Process Steps	Advantage	Disadvantage (Limitation)
1	Speaker Encoding: This encoding captures the unique characteristics of the speaker's voice.	Precise Control: The solution allows for precise control over various style aspects of the synthesized speech, such as tone, speaking rate, and emphasis.	Chance of Overfitting
2	Training the Model: A multi-speaker Text-to-Speech (TTS) model is trained using the speaker encodings and other conditioning factors.	Few Samples Required: The system can generate a speaker's voice using only a few reference audio samples, making it applicable in scenarios where limited data is available.	None
3	Cloning Tasks: The trained model can be used for different voice cloning tasks. These tasks include synthesizing speech directly from text and style control.	Expressive Cloning: The model can capture and reproduce the unique characteristics of a speaker's voice, enabling expressive voice cloning.	Ethical Concerns: The technology can be potentially misused for unethical purposes

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
Mel spectrogram Attention map	Text Speaker encoding Pitch contour	None	None

Relationship Among The Above 4 Variables in This article

Mel spectrogram and attention map are the visualization techniques whose result is affected by the independent variables like text, speaker encoding and Pitch contour.

Input and Output		Feature of This Solution	Contribution & The Value of This Work				
<table><tr><th>Input</th><th>Output</th></tr><tr><td>Text, multi-speaker dataset.</td><td>Mel-spectrogram, synthesised audio.</td></tr></table>		Input	Output	Text, multi-speaker dataset.	Mel-spectrogram, synthesised audio.	Controllable voice cloning Speech synthesis from text Precise style control	The contribution of this solution is the ability to control the expressiveness of the synthesized audio, including variation in tone, speaking rate, emphasis, and emotions.
Input	Output						
Text, multi-speaker dataset.	Mel-spectrogram, synthesised audio.						
Positive Impact of this Solution in This Project Domain			Negative Impact of this Solution in This Project Domain				

<p>The system can generate speech that is not only accurate but also expressive, capturing variations in tone, speaking rate, emphasis, and emotions.</p> <p>This capability has several applications, such as voice-overs in animated films, synthesizing realistic and expressive speech for videos.</p>	<p>Misuse and unethical use:</p> <p>The technology can be abused for creating inappropriate content, spreading misinformation, or generating voice-overs for DeepFake videos. This can have negative consequences for individuals and society.</p>	
<p>Analyse This Work By Critical Thinking</p>	<p>The Tools That Assessed this Work</p>	<p>What is the Structure of this Paper</p>

<p>This work demonstrates a promising approach to expressive neural voice cloning, but further research and comparisons with existing methods are needed to establish its effectiveness and applicability in different contexts.</p>	<p>Style-MOS</p>	<p>1.Introduction 2.Related Work 3.Methodology 4.Experiments 5.Results 6.Discussion 7.Conclusion 8.Future Work 9.References</p>
--	------------------	---

Diagram/Flowchart

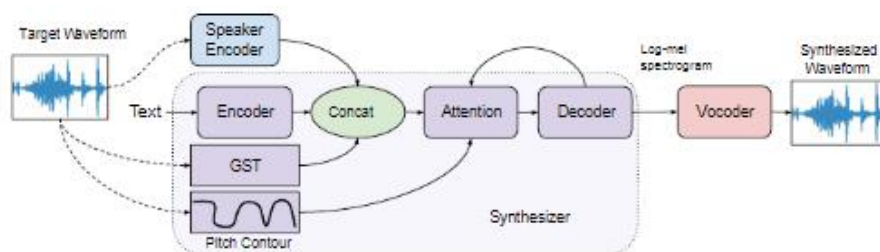


Figure 1: Expressive voice cloning model: Tacotron-2 TTS model conditioned on speaker and style characteristics derived from the target audio of a given text. At inference time, the model can be provided independent references for style and speaker encodings to achieve expressive voice cloning.

---End of Paper 1---

2	Neural Voice Cloning with a Few Samples	
Reference in APA format		
URL of the Reference	Authors Names and Emails	Keywords in this Reference

1802.06006v3.pdf (arxiv.org)	<p>Sercan Ö. Arik - sercanarik@baidu.com</p> <p>Jitong Chen - chenjitong01@baidu.com</p> <p>Kainan Peng - pengkainan@baidu.com</p> <p>Wei Ping - pingwei01@baidu.com</p> <p>Yanqi Zhou - yanqiz@baidu.com</p>	<p>Voice cloning</p> <p>Sequence-to-sequence neural speech synthesis systems</p> <p>Speaker adaptation</p> <p>Speaker encoding</p> <p>Voice morphing</p>
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	The Goal (Objective) of this Solution & What is the problem that need to be solved	What are the components of it?
voice cloning in sequence-to-sequence neural speech synthesis systems	<p>The goal of the voice cloning system is to synthesize a person's voice from only a few audio samples. The system aims to solve the problem of voice cloning, which involves learning the voice of a speaker from a limited amount of data and generating speech that sounds like it is pronounced by the target speaker.</p>	<p>Speaker adaptation: This approach involves fine-tuning a multi-speaker generative model for a speaker using a few audio-text pairs.</p> <p>Speaker encoding: In this approach, a separate model called the speaker encoder is trained to directly estimate the speaker embedding from audio samples of an unseen speaker. This model does not require fine-tuning during voice cloning.</p>
The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process		

The voice cloning system developed by Baidu Research addresses the challenge of learning speaker characteristics from limited data and generating voice for unseen texts through two approaches: speaker adaptation and speaker encoding.

	Process Steps	Advantage
1	<p>Speaker Adaptation:</p> <p>This approach involves fine-tuning a pre-trained multi-speaker model using a few samples from an unseen speaker.</p> <p>By adapting the model to the specific characteristics of the new speaker, the system can generate voice that closely resembles the target speaker.</p>	<p>Good cloning quality</p> <p>Naturalness</p> <p>Fast cloning</p>
2	<p>Speaker Encoding:</p> <p>This encoding model takes only a few audio samples from the target speaker as input.</p> <p>This approach offers a more efficient and resource-friendly solution for voice cloning.</p>	<p>Compact representation</p> <p>Better performance with more cloning audios</p>

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
Evaluation methods	Cloning audios Training dataset Model architecture and hyperparameters	Sample Count Voice cloning approach Cloning Time	Computational resource requirements

Relationship Among The Above 4 Variables in This article

The independent variables I.e, cloning audios, training dataset, model architecture and hyper parameters have an effect on Evaluation methods. This effect is increased or decreased by the moderating variables(sample count, voice cloning approach and cloning time). Mediating variables such as computational resource requirements act as a bridge between the independent and dependent variables.

Input and Output	Feature of This Solution	Contribution in This Work
------------------	--------------------------	---------------------------

		<p>Speaker Adaptation: It allows fine-tuning a pre-trained multi-speaker model for an unseen speaker using a few samples.</p> <p>Automated Evaluation Methods: The solution introduces automated evaluation methods for voice cloning.</p>	<p>They show that fine-tuning a pre-trained multi-speaker model with a few samples from an unseen speaker can achieve good cloning quality.</p> <p>The authors propose automated evaluation methods for voice cloning.</p>
Input	Output		
cloning audios and the text that needs to be synthesized.	synthesized audio that mimics the voice of the speaker in the cloning audios.		
Positive Impact of this Solution in This Project Domain		Negative Impact of this Solution in This Project Domain	
With this solution, it becomes possible to synthesize a person's voice from just a few audio samples. This can be beneficial in various applications such as voice assistants and audiobooks		Misuse of Cloned Voices: Voice cloning technology can be misused for fraudulent activities such as impersonation, identity theft, or creating fake audio evidence. This can have serious legal and ethical implications.	
Analyse This Work By Critical Thinking		The Tools That Assessed this Work	What is the Structure of this Paper
The paper presents an approach to produce synthesized audio outputs from limited number of samples. It proposes automated evaluation methods. These contributions contribute to the advancement of voice cloning technology.		Human evaluations	<p>Introduction</p> <p>Related Work</p> <p>Speaker Cloning Approaches</p> <p>Automated Evaluation Methods</p> <p>Voice Morphing</p> <p>Experimental Setup</p> <p>Results and Analysis</p> <p>Conclusion</p>
Diagram/Flowchart			
None			

--End of Paper 2--

3		
Reference in APA format	Deep Voice: Real-time Neural Text-to-Speech	
URL of the Reference	Authors Names and Emails	Keywords in this Reference

https://arxiv.org/pdf/1702.07825.pdf	<p>Sercan O. Arık † SERCANARIK@BAIDU.COM</p> <p>Mike Chrzanowski† MIKECHRZANOWSKI@BAIDU.COM</p> <p>Adam Coates† ADAMCOATES@BAIDU.COM</p> <p>Gregory Diamos† GREGDIAMOS@BAIDU.COM</p> <p>Andrew Gibiansky† GIBIANSKYANDREW@BAIDU.COM</p> <p>Yongguo Kang† KANGYONGGUO@BAIDU.COM</p> <p>Xian Li† LIXIAN05@BAIDU.COM</p> <p>John Miller† MILLERJOHN@BAIDU.COM</p> <p>Andrew Ng† ANDREWNG@BAIDU.COM</p> <p>Jonathan Raiman† JONATHANRAIMAN@BAIDU.COM</p> <p>Shubho Sengupta† SSENGUPTA@BAIDU.COM</p> <p>Mohammad Shoeybi† MOHAMMAD@BAIDU.COM</p>	<p>Deep Voice</p> <p>Real time</p> <p>Neural Text-to-Speech</p> <p>Grapheme-to-phoneme conversion</p> <p>Audio synthesis</p> <p>Neural networks</p> <p>WaveNet</p>
<p>The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)</p>	<p>The Goal (Objective) of this Solution & What is the problem that need to be solved</p>	<p>What are the components of it?</p>

Deep Voice: Real-time Neural TTS	To develop a text-to-speech (TTS) system that solves the problem of generating high-quality and natural-sounding speech from text in real-time. It focuses on optimizing the different components of a TTS system.	Grapheme-to-phoneme model Segmentation model Phoneme duration model Fundamental frequency model Audio synthesis model
----------------------------------	--	---

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process

	Process Steps	Advantage
1	Training the Model: The model is trained using a dataset of short audio clips and corresponding textual transcripts. This step allows the model to learn the mapping between text and speech.	It minimizes the use of hand-engineered features, making it easier to train and reproduce the system.
2	WaveNet Inference: WaveNet is an autoregressive model that generates speech waveform samples one at a time.	Real-time synthesis, making the system usable for various applications.

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	
Phoneme durations Audio synthesis quality Mean Opinion Score (MOS) ratings	Model architecture Phoneme duration	None	
<div>Relationship Among The Above 4 Variables in This article</div> <div>The model architecture is expected to have an impact on the quality of audio synthesis. The different components of the model work together to generate the final output, and the quality of each component can affect the overall audio synthesis quality.</div>			
Input and Output		Features of this Solution	Contribution & The Value of This Work
		Standalone System Real-time Inference Efficient WaveNet Inference	The authors built a fully neural system that can generate speech in real-time, offering adjustable trade-off between synthesis speed and audio quality. They optimized the inference process to achieve faster-than-real-time speeds, making the system usable for various applications.
Input	Output		
Dataset of short audio clips Corresponding textual transcripts	High-quality synthesized audio		
Positive Impact of this Solution in This Project Domain			Negative Impact of this Solution in This Project Domain
It offers real-time inference, which is essential for practical applications of TTS. It simplifies the process of creating TTS systems and opens up new possibilities for exploration in the field.			The use of AI-generated voices could raise ethical concerns, particularly in cases where the technology is used to manipulate individuals. For example, malicious actors could use the technology to create fake audio recordings for fraudulent purposes.

Analyse This Work By Critical Thinking	The Tools That Assessed this Work	What is the Structure of this Paper
<p>One of the key contributions of this work is the development of efficient, real-time WaveNet inference.</p> <p>While the work demonstrates impressive results, there are still areas for further exploration.</p> <p>Overall, the work presents a significant advancement in the field of TTS systems, showcasing the potential of neural networks for generating high-quality speech in real-time.</p>	<p>MOS (Mean Opinion Score)</p> <p>Blizzard 2013 Dataset</p> <p>Performance Model</p>	<p>1.Introduction</p> <p>2.Related Work</p> <p>3.Model Architecture</p> <p style="padding-left: 40px;">a)WaveNet Model</p> <p style="padding-left: 40px;">b)Conditioning Network</p> <p>4.Training</p> <p style="padding-left: 40px;">a)Segmentation Results</p> <p style="padding-left: 40px;">b)Grapheme-to-Phoneme Results</p> <p style="padding-left: 40px;">c)Phoneme Duration and Fundamental Frequency Results</p> <p style="padding-left: 40px;">d)Audio Synthesis Results</p> <p>5.Conclusion</p> <p>6.References</p>
Diagram/Flowchart		

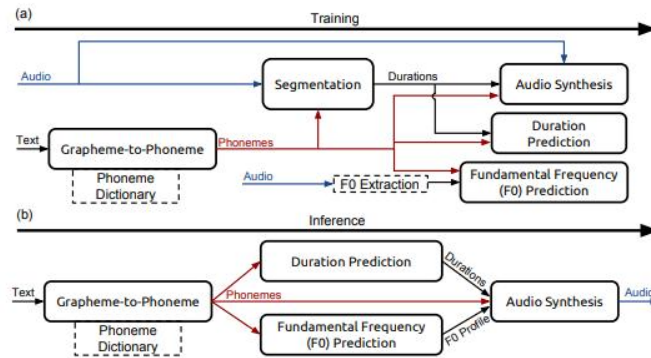


Figure 2. system diagram depicting (a) training procedure and (b) inference procedure, with inputs on the left and outputs on the right.

--End of Paper 3--

4		
Reference in APA format	AN OVERVIEW OF REAL-TIME CHAT APPLICATION	
URL of the Reference	Authors Names and Emails	Keywords in this Reference
https://www.ijrti.org/papers/IJRTI2206316.pdf	Akshata D Vhandale, Sayam N Gandhak,Saundarya A Karhale, Sandipkumar R Prasad, Prof. Sudhesh A Bachwani	Chat application Python MongoDB Express JS Node.js
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	The Goal (Objective) of this Solution & What is the problem that need to be solved	What are the components of it?

Real-time Chat application	<p>The goal of the current solution is to enable users to communicate with each other seamlessly.</p> <p>The chat application provides a platform for users to connect and exchange messages in real-time, regardless of their location.</p> <p>It is designed to facilitate instant messaging and improve communication among users.</p>	<p>Server</p> <p>Database</p> <p>User Interface</p> <p>Direct Messages</p>
----------------------------	---	--

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process

The MERN stack is a combination of four technologies: MongoDB, Express.js, React, and Node.js.

	Process Steps	Advantage
1	MongoDB: MongoDB is a NoSQL database used for storing and managing data. It offers flexibility in handling large amounts of data by using a document-oriented structure.	Ability to handle unstructured data efficiently
2	Express.js: Express.js is a web application framework for Node.js. It simplifies the process of building the backend of a web application by providing a set of tools and middleware.	Allows for asynchronous programming and follows a single-threaded architecture, which improves performance.
3	React: React is a JavaScript library used for building user interfaces. It breaks down the UI into reusable components, making it easier to develop and maintain complex applications.	React offers a virtual DOM, which enhances performance by efficiently updating only the necessary parts of the UI.
	Node.js: Node.js is a JavaScript runtime environment that allows developers to run JavaScript on the server-side.	Suitable for building scalable and high-performance applications.

Major Impact Factors in this Work							
Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable				
None	None	None	None				
Relationship Among The Above 4 Variables in This article							
NA							
Input and Output		Feature of This Solution	Contribution & The Value of This Work				
<table><tr><th>Input</th><th>Output</th></tr><tr><td>User details, messages, user actions</td><td>User interface of the chat application</td></tr></table>		Input	Output	User details, messages, user actions	User interface of the chat application	User-friendly GUI Real-time messaging Contextual enrichment E2EE secured chats	The contribution of this work is the development of a web-based chat application using the MERN stack (MongoDB, Express.js, React.js, and Node.js). This work contributes to the field of web development by showcasing the capabilities of the MERN stack in building real-time chat applications.
Input	Output						
User details, messages, user actions	User interface of the chat application						
Positive Impact of this Solution in This Project Domain			Negative Impact of this Solution in This Project Domain				
User friendliness Secure communication Instant communication Scalability			The increased usage of chat applications may lead to Social isolation and reduced face-to-face interaction.				
Analyse This Work By Critical Thinking		The Tools That Assessed this Work	What is the Structure of this Paper				

<p>The overview emphasizes the need for a real-time chat application with multi-platform support.</p> <p>The choice of the MERN stack reflects a strategic decision, leveraging MongoDB, Express.js, React, and Node.js for a comprehensive and efficient development process.</p>	<p>MongoDB</p> <p>Express.js</p> <p>React.js</p>	<p>The paper is structured as follows:</p> <p>Introduction</p> <p>Node.js</p> <p>Implementation of Real-Time Chat Application</p> <p>MongoDB</p> <p>MERN Stack</p> <p>Performance Optimization</p> <p>Conclusion</p>
Diagram/Flowchart		
None		

--End of Paper 4--

5		
Reference in APA format	EmoSpeech: Guiding FastSpeech2 Towards Emotional Text to Speech	
URL of the Reference	Authors Names and Emails	Keywords in this Reference
https://arxiv.org/pdf/2307.00024.pdf	<p>Daria Diatlova - d.dyatlova@vk.team</p> <p>Vitaly Shutov - vi.shutov@corp.vk.com</p>	<p>EmoSpeech</p> <p>Speech synthesis</p> <p>Emotion</p> <p>Multi-speaker</p> <p>Lightweight solution</p>
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	The Goal (Objective) of this Solution & What is the problem that need to be solved	What are the components of it?

EmoSpeech	It aims to solve the problem of synthesizing speech with desired emotions. The EmoSpeech model extends the FastSpeech2 architecture with modifications that enable conditioning on a given emotion while maintaining fast inference speed.	This step involves extracting phonemes and punctuation from text using the grapheme-to-phoneme (GTP) model. It also includes extracting durations of phonemes and pitch from waveforms.
-----------	---	---

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process

The process involves training a speech synthesis model called EmoSpeech using a dataset of emotional speech recordings.

Here are the steps involved:

	Process Steps	Advantage
1	Data Preprocessing: The dataset of emotional speech recordings is preprocessed to extract features such as mel spectrograms and eGeMAPS features. These features capture the acoustic characteristics of the speech.	Helps in capturing the necessary information for synthesizing emotional speech.
2	Model Architecture: The EmoSpeech model is designed with a multi-speaker and multi-emotional setup. It uses a conditioning discriminator which helps in controlling the emotions of the generated speech.	Helps in controlling the emotions of the generated speech.
3	Training: The EmoSpeech model is trained using reconstruction loss and an adversarial loss.	Reconstruction loss ensures that the generated speech resembles the original input. Adversarial loss helps in improving the quality and emotional expression of the generated speech.

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
MOS and NIQSA scores	Emotion-unlabeled dataset Style control techniques	None	None

Relationship Among The Above 4 Variables in This article

MOS score is affected by the quality of the synthesised speech whereas NIQSA score is affected by the naturalness of the speech.

Input and Output		Feature of this Solution	Contribution & The Value of This Work			
<table><tr><th>Input</th><th>Output</th></tr><tr><td>Input text</td><td>Synthesised audio with emotions</td></tr></table>	Input	Output	Input text	Synthesised audio with emotions	<div>-Extension of FastSpeech2 -Conditioning Mechanism -Lightweight Solution</div>	<p>The contribution of this work is the development of a text-to-speech (TTS) system that can generate expressive speech with controllable emotions.</p> <p>They also introduce a style control mechanism that allows users to specify the desired emotion in the synthesized speech.</p> <p>The value of this work lies in its potential applications in various domains such as virtual assistants, audiobooks etc.</p>
Input	Output					
Input text	Synthesised audio with emotions					

Positive Impact of this Solution in This Project Domain	Negative Impact of this Solution in This Project Domain
<div>-Improved Naturalness -Potential applications such as virtual assistants, audiobooks etc.</div>	<div>Overtraining could result in synthesized speech that sounds exaggerated or unnatural.</div>

Analyse This Work By Critical Thinking	The Tools That Assessed this Work	What is the Structure of this Paper
<p>The work proposes a great approach to enhance emotional speech synthesis.</p> <p>The experimental results demonstrate its effectiveness, but further research is needed to find the potential for improvement.</p>	openSMILE toolkit, CatBoost classifier	<p>Introduction</p> <p>Previous approach</p> <p>Methodology</p> <p>Experimental Setup</p> <p>Results and Analysis</p> <p>Limitations</p> <p>Conclusion</p> <p>References</p>
Diagram/Flowchart		
None		

6		
Reference in APA format	Efficient Neural Audio Synthesis	
URL of the Reference	Authors Names and Emails	Keywords in this Reference
https://arxiv.org/pdf/1802.08435.pdf	Nal Kalchbrenner , Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman, Koray Kavukcuoglu	WaveRNN, Weight Pruning, Subscale Dependency Scheme
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	The Goal (Objective) of this Solution & What is the problem that need to be solved	What are the components of it?
Sparse WaveRNN	The goal of the Sparse WaveRNN solution is to reduce the computational requirements for audio synthesis while maintaining high audio quality. It aims to solve the problem of real-time or faster audio synthesis on GPUs, as well as enabling audio synthesis on low-power mobile CPUs.	<p>Wave Recurrent Neural Networks (WaveRNN)</p> <p>Sparse WaveRNN</p> <p>Subscale Dependency Scheme</p> <p>Batched Sampling</p>
The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process		

The Sparse WaveRNN solution is a method for efficient audio synthesis using neural networks. It addresses the problem of high computational requirements and memory bandwidth limitations in traditional WaveRNN models.

	Process Steps	Advantage	Disadvantage (Limitation)
1	Sparsity: The first step is to introduce sparsity in the model by reducing the number of connections between neurons. This reduces the overall parameter count and computational requirements.	It allows for larger models with better audio quality within the same computational budget	It requires specialized techniques and algorithms to handle sparse matrices efficiently
2	Block-Sparse Matrix-Vector Product: To handle the sparse matrices, the solution uses high-performance block-sparse matrix-vector product operations. These operations optimize the computation and memory access patterns, improving the efficiency of the model.	It reduces the memory bandwidth requirements and speeds up the computation.	Implementing these operations can be complex and requires specialized knowledge.
3	Subscale Dependency Scheme: The solution introduces a subscale dependency scheme to generate multiple samples per step. This allows for batched computation and parallelization, increasing the overall sampling speed.	It improves the throughput and efficiency of the model, especially when using multiple GPU devices.	It requires careful management of dependencies and may introduce a slight sampling lag.

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
the audio samples generated by the model.	the sparsity level of the weight matrices.	The sparsity of the weight matrices.	The subscale dependency scheme

Relationship Among The Above 4 Variables in This article

In the Sparse WaveRNN solution for audio synthesis discussed in this article, the dependent variable is the audio samples that are stuff synthesized. The independent variable is the sparsity level of the weight matrices, which refers to the stratum of sparsity or density of the weight matrices used in the model. The moderating variable is moreover the sparsity of weight matrices, which influences the relationship between the independent variable (sparsity level) and the dependent variable (audio samples). The mediating or intervening variable is the subscale dependency scheme, which is a generation process based on subsampling.

Input and Output		Feature of This Solution	Contribution & The Value of This Work			
<table><tr><th>Input</th><th>Output</th></tr><tr><td>a sequence of audio samples</td><td>generated sequence of audio samples</td></tr></table>	Input	Output	a sequence of audio samples	generated sequence of audio samples	the Sparse WaveRNN solution offers an efficient and effective approach to neural audio synthesis.	Overall, the Sparse WaveRNN solution provides a high-quality, efficient, and scalable approach to audio synthesis, making it a valuable contribution to the field.
Input	Output					
a sequence of audio samples	generated sequence of audio samples					
Positive Impact of this Solution in This Project Domain		Negative Impact of this Solution in This Project Domain				
the Sparse WaveRNN enables real-time on-device audio synthesis with high quality, making it a significant advancement in the field.		While the Sparse WaveRNN solution offers advantages in terms of efficiency and resource usage, there are potential drawbacks in terms of audio quality, limited applicability, implementation complexity, and compatibility.				
Analyse This Work By Critical Thinking	The Tools That Assessed this Work	What is the Structure of this Paper				
Sparse WaveRNN offers real-time audio synthesis with high quality, but it may have limitations in capturing long-range dependencies and requires specific hardware for optimal performance.	Negative Log-Likelihood (NLL) metric and the Mean Opinion Score (MOS) metric. These metrics were used to evaluate the performance and quality of the Sparse WaveRNN model in comparison to other models.	<div>Abstract</div> <div><div>I. Introduction</div><div>II. Wave Recurrent Neural Networks</div><div>III. WaveRNN Sampling on GPU</div><div>IV. Sparse WaveRNN</div><div>V. Subscale Dependency Scheme</div><div>VI. Batched Sampling</div><div>VII. Results</div><div>VIII. Conclusion</div></div>				

Diagram/Flowchart

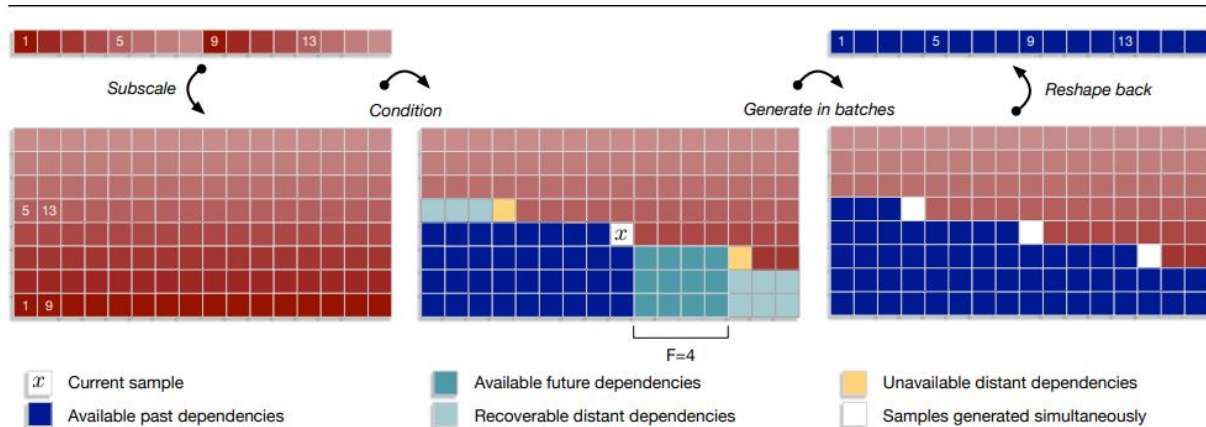


Figure 3. working of Sparse WaveRNN

---End of Paper 6

7		
Reference in APA format	TACOTRON: TOWARDS END-TO-END SPEECH SYNTHESIS	
URL of the Reference	Authors Names and Emails	Keywords in this Reference
https://arxiv.org/pdf/1703.10135.pdf	Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrghiannakis, Rob Clark, and Rif A. Saurous.	Tacotron, end-to-end, speech synthesis, text-to-speech, generative model, mean opinion score, parametric system, concatenative system, CBHG module, attention mechanism, encoder, decoder, spectrogram, waveform synthesis.
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	The Goal (Objective) of this Solution & What is the problem that need to be solved	What are the components of it?
Tacotron: Towards End-to-End Speech Synthesis	The goal of the Tacotron solution is to develop an end-to-end generative text-to-speech (TTS) model that can synthesize speech	1. Text analysis frontend: This component extracts various linguistic features from the

	<p>directly from characters. The problem it aims to solve is the complexity and laborious nature of traditional TTS pipelines, which typically consist of multiple stages and require extensive domain expertise. Tacotron aims to simplify the TTS process by training a single model on <text, audio> pairs, eliminating the need for feature engineering and allowing for rich conditioning on various attributes.</p>	<p>input text.</p> <ol style="list-style-type: none"> 2. Acoustic model: The acoustic model predicts the acoustic features of the speech based on the linguistic features extracted by the frontend. 3. Audio synthesis module: This module synthesizes the speech waveform from the predicted acoustic features. 4. Sequence-to-sequence framework: Tacotron uses a sequence-to-sequence model to generate the speech spectrogram directly from the input characters. 5. Waveform synthesis: Tacotron uses a simple waveform synthesis technique to convert the generated spectrogram into a speech waveform.
--	---	--

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process

	Process Steps	Advantage	Disadvantage (Limitation)
1	Encoder: The encoder takes characters as input and converts them into a high-level representation. It uses a sequence of convolutional layers followed by a bidirectional gated recurrent unit (GRU) to capture the contextual information of the input characters. The advantage of the encoder is that it can effectively extract meaningful representations from the character-	End-to-end Training: Tacotron can be trained completely from scratch with random initialization, eliminating the need for extensive domain expertise and manual feature engineering.	Training Data Requirements: Tacotron requires a large amount of paired <text, audio> data for training, which may be challenging to obtain in some cases.

	level inputs.		
2	<p>Attention-based Decoder: The attention-based decoder takes the high-level representation from the encoder and generates the spectrogram frames. It uses an attention mechanism to align the generated frames with the input characters. This allows the model to focus on relevant parts of the input during the synthesis process. The advantage of the attention mechanism is that it enables the model to handle long input sequences and capture the dependencies between characters and spectrogram frames</p>	<p>Faster Generation: Since Tacotron generates speech at the frame level, it is substantially faster than sample-level autoregressive methods like WaveNet.</p>	<p>Alignment Learning: The alignment between characters and spectrogram frames is learned by the model, which can be challenging and may result in imperfect alignments.</p>
3	<p>Post-processing Network: The post-processing network takes the generated spectrogram frames and converts them into waveforms. It applies a series of convolutional layers to refine the spectrogram frames and then uses an inverse Fourier transform to obtain the waveforms. The advantage of the post-processing network is that it improves the quality and naturalness of the synthesized speech.</p>	<p>Robustness: The end-to-end nature of Tacotron makes it more robust compared to multi-stage models, as errors from each component do not compound.</p>	<p>Limited Experimental Results: The experimental results of Tacotron are mainly evaluated on US English, and the performance on other languages or datasets may vary</p>

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
spectrogram frames	the character sequence inputs	CBHG module	The post-processing net

Relationship Among The Above 4 Variables in This article		
<p>The encoder, attention-based decoder, post-processing net, and CBHG module work together in the Tacotron model to convert text input into synthesized speech. The encoder extracts sequential representations of the text, the attention-based decoder generates spectrogram frames, the post-processing net converts the frames into waveforms, and the CBHG module helps improve the capability of the model to extract features from the text input.</p>		

Input and Output	Feature of This Solution	Contribution in This Work
	Tacotron offers a	It eliminates the need for

<table><tr><th>Input</th><th>Output</th></tr><tr><td>Characters</td><td>spectrogram frames</td></tr></table>	Input	Output	Characters	spectrogram frames	<p>simpler and more efficient approach to text-to-speech synthesis, with improved naturalness and faster generation speed.</p>	<p>multiple stages in traditional TTS systems, such as a text analysis frontend, an acoustic model, and an audio synthesis module. Tacotron can be trained completely from scratch with random initialization, making it easier to build TTS systems without extensive domain expertise.</p>
Input	Output					
Characters	spectrogram frames					
Positive Impact of this Solution in This Project Domain		Negative Impact of this Solution in This Project Domain				
<p>It provides an end-to-end generative text-to-speech model that can synthesize speech directly from characters. This eliminates the need for multiple stages and complex components in traditional TTS pipelines, reducing the engineering efforts required to build a new system.</p>		<ul style="list-style-type: none">• Computational Resources: Tacotron, like other deep learning models, can be computationally expensive to train and deploy. It may require powerful hardware and infrastructure to achieve real-time performance or to scale for large-scale applications.• Generalization to Other Languages and Accents: The Tacotron model described in the context is trained on US English data. It may not generalize well to other languages or accents, requiring additional training data and modifications to the model.• Training Data Requirements: Tacotron requires a large amount of training data, specifically <text, audio> pairs, to achieve optimal performance. Collecting and curating such datasets can be time-consuming and resource-intensive.				
Analyse This Work By Critical Thinking		The Tools That Assessed this Work	What is the Structure of this Paper			
Tacotron is an integrated end-to-end generative TTS		Mean	Abstract			

<p>model that shows promising results in terms of naturalness and speed. It eliminates the need for hand-engineered linguistic features and complex components like an HMM aligner. However, there are still areas for improvement, such as the output layer, attention module, loss function, and waveform synthesizer. The authors are working on improving the quality of the waveform synthesis and exploring advancements in learned text normalization.</p>	<p>opinion score (MOS) tests.</p>	<p>I. Introduction II. Related Work III. Model Architecture IV. Model details V. Experiment Results VI. Discussions VII. References</p>
<p>Diagram/Flowchart</p>		
<p>Figure 4. Model architecture. The model takes characters as input and outputs the corresponding raw spectrogram, which is then fed to the Griffin-Lim reconstruction algorithm to synthesize speech.</p>		

--End of Paper 7--

8		
Reference in APA format	WAVENET: A GENERATIVE MODEL FOR RAW AUDIO	
URL of the Reference	Authors Names and Emails	Keywords in this Reference
https://arxiv.org/pdf/1609.03499.pdf	Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu.	Pulse Code Modulation, language modeling, hidden Markov models, linear prediction, text-to-speech conversion, pitch-adaptive time-frequency smoothing, speech synthesis, WaveNet, multi-speaker speech generation, mean opinion score, statistical parametric speech

		synthesis, vocoder, deep convolutional nets, fully connected CRFs, acoustic theory of speech production	
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	The Goal (Objective) of this Solution & What is the problem that need to be solved	What are the components of it?	
WaveNet	The goal of WaveNet is to generate raw audio waveforms using a deep neural network. It aims to solve the problem of generating high-quality and natural-sounding audio, such as speech and music. WaveNet is designed to be fully probabilistic and autoregressive, meaning that the prediction for each audio sample is conditioned on all previous samples. This allows WaveNet to capture the characteristics of different speakers and generate realistic audio with smooth intonations. Additionally, WaveNet can be used for tasks like text-to-speech and phoneme recognition.	Dilated Causal Convolutions, Autoregressive , Modeling Conditional Modeling , Global Conditioning ,Local Conditioning are the components that allow WaveNet to generate raw audio waveforms with high fidelity and capture the characteristics of different speakers or audio.	
The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process			
WaveNet is a generative model that operates directly on the raw audio waveform. It models the joint probability of a waveform by factorizing it into conditional probabilities. The model consists of a stack of convolutional layers, with no pooling layers, and the output has the same time dimensionality as the input.			
	Process Steps	Advantage	Disadvantage (Limitation)
1	Dilated Causal Convolutions: WaveNet uses dilated causal convolutions to capture long-range temporal dependencies in audio signals. These convolutions have exponentially growing receptive fields, allowing the model to capture information from a large context window..	WaveNet operates directly on the raw audio waveform, allowing it to capture fine-grained details and produce high-quality audio samples.	Designing the filters for WaveNet can be challenging, as it requires training the model from data to learn the optimal filters.

2	<p>Conditioning:</p> <p>WaveNet can be conditioned on other inputs in a global or local way. Global conditioning involves providing additional inputs, such as speaker identity, to the model. Local conditioning involves transforming a separate time series, such as linguistic features, using transposed convolutional networks or 1x1 convolutions.</p>	<p>The use of dilated causal convolutions enables WaveNet to model long-range temporal dependencies, which is crucial for generating realistic audio.</p>	<p>WaveNet can be computationally expensive, especially when using larger context stacks or conditioning on multiple inputs.</p>
3	<p>Context Stacks:</p> <p>To further increase the receptive field size, WaveNet can use separate context stacks that process a long part of the audio signal and locally condition a larger WaveNet. Multiple context stacks with varying lengths and numbers of hidden units can be used.</p>	<p>WaveNet can be conditioned on other inputs, such as speaker identity or linguistic features, allowing for control over the generated audio</p>	<p>The training process for WaveNet can be time-consuming, as it requires a large amount of audio data to learn the complex patterns in the waveform.</p>

Major Impact Factors in this Work

The major impact factors in this work include the use of WaveNet, a deep generative model for raw audio, for tasks such as multi-speaker speech generation, text-to-speech synthesis, and music audio modeling. The authors also explore the use of conditioning techniques to control the output of the model based on different inputs, such as speaker ID or linguistic features. Additionally, the use of context stacks to increase the receptive field size of the model is another important factor in this work.

Dependent Variable	Independent Variable	Moderating variable
categorical distribution	Raw audio waveform	None

Relationship Among The Above 4 Variables in This article

These variables are interconnected in the equations and operations .

Input and Output		Feature of This Solution	Contribution & The Value of This Work
Input	Output	<p>Learning the speech front-end with raw waveform Deep auto-encoder based low-dimensional feature extraction from FFT spectral envelopes</p> <p>Postfilters to modify the modulation spectrum for</p>	<p>The value of this work is in the advancement of speech synthesis techniques and the potential for generating high-quality, customizable music samples.</p>
Raw audio waveform	Categorical distribution over next audio sample indicating probability of possible value for next sample.		

	<p>statistical parametric speech synthesis.</p> <p>Generative image modeling using spatial LSTMs</p> <p>Speech parameter generation algorithm considering global variance for HMM-based speech synthesis</p>	
Positive Impact of this Solution in This Project Domain		Negative Impact of this Solution in This Project Domain
it improves the quality and naturalness of speech synthesis.		Computational complexity, training data requirements, or performance issues.
Analyse This Work By Critical Thinking	The Tools That Assessed this Work	What is the Structure of this Paper
These observations highlight the strengths and limitations of WaveNet in speech synthesis, particularly in terms of naturalness, prosody, and speaker modeling.	<p>Subjective paired comparison tests and mean opinion score (MOS) tests were conducted.</p> <p>These evaluations were used to compare the performance of the WaveNet TTS system with baseline statistical parametric and concatenative speech synthesizers.</p>	<p>Abstract</p> <ul style="list-style-type: none"> • Introduction • Background • Methodology • Evaluation • Results • Discussion • Conclusion • References
Diagram/Flowchart		

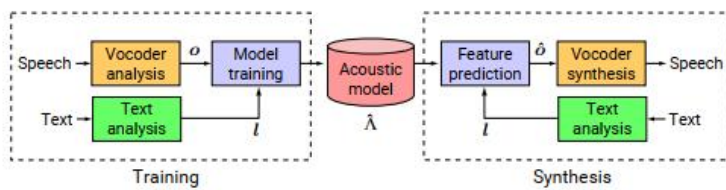


Figure 5. Outline of statistical parametric speech synthesis.

--End of Paper 8--

9		
Reference in APA format	DEEP VOICE 3: SCALING TEXT-TO-SPEECH WITH CONVOLUTIONAL SEQUENCE LEARNING	
URL of the Reference	Authors Names and Emails	Keywords in this Reference
https://in.docworkspace.com/d/sILbDueGLAdbWw6oG	Sercan Ö. Arik, Gregory Diamos, Andrew Gibiansky, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou.	Deep Voice 3, neural text-to-speech system, fully-convolutional sequence-to-sequence acoustic model, position-augmented attention mechanism, waveform synthesis, Griffin-Lim spectrogram inversion, WaveNet, WORLD vocoder, multi-speaker speech synthesis, trainable speaker embeddings, text normalization, performance characteristics, MOS evaluations.
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	The Goal (Objective) of this Solution & What is the problem that need to be solved	What are the components of it?
Deep Voice 3.	The goal of Deep Voice 3 is to develop a high-quality and efficient text-to-speech (TTS) system. It aims to solve the problem of synthesizing natural and intelligible speech from text input. Deep Voice 3	Encoder: This component is responsible for converting textual features into an internal learned representation. It uses a fully-convolutional

	addresses the limitations of previous TTS systems by using an attention-based sequence-to-sequence model, which allows for a more compact architecture and avoids common attention errors. It also focuses on improving training speed and inference efficiency to make TTS feasible for production systems.	<p>architecture.</p> <p>Decoder: The decoder decodes the learned representation using a multi-hop convolutional attention mechanism. It generates a low-dimensional audio representation.</p> <p>Converter: The converter is a post-processing network that predicts the final vocoder parameters from the hidden states of the decoder. Unlike the decoder, the converter is non-causal and can depend on future context information.</p>
--	--	--

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process

Deep Voice 3 is a fully-convolutional attention-based neural text-to-speech (TTS) system that converts written language into human speech. It consists of three main components: the encoder, the decoder, and the converter.

	Process Steps	Advantage	Disadvantage (Limitation)
1	Encoder: The encoder takes the input text and converts it into an internal learned representation using a fully-convolutional architecture. This allows for parallel computation and faster training compared to recurrent architectures.	Ability to process text quickly and efficiently.	It may not capture long-range dependencies as effectively as recurrent models.
2	Decoder: The decoder uses a multi-hop convolutional attention mechanism to decode the learned representation from the encoder into a low-dimensional audio representation, specifically mel-scale spectrograms. It does this in an autoregressive manner, predicting one timestep at a time	Ability to generate high-quality spectrograms.	It may suffer from attention errors, such as repeated words, mispronunciations, or skipped words. artifacts or distortions in the synthesized speech.
3	Converter: The converter network takes the hidden states from the decoder and predicts the vocoder parameters for waveform synthesis. It is a non-causal network, meaning it can depend	Ability to generate the final waveform parameters	It may introduce artifacts or distortions in the synthesized speech.

on future context information.			
Major Impact Factors in this Work			
Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
mel-scale log magnitude spectrograms,	waveform synthesis method.	Speaker embeddings	Converter
Relationship Among The Above 4 Variables in This article			
mel-scale log magnitude spectrograms are generated by the decoder, the waveform synthesis method is determined by the chosen technique, the speaker embedding allows for multi-speaker synthesis, and the converter network predicts the parameters required for the waveform synthesis method.			
Input and Output		Feature of This Solution	Contribution & The Value of This Work
Input	Output	Encoder,Decoder ,Converter, work together to optimize the overall objective function, which is a combination of losses from the decoder and the converter. The model also includes text preprocessing steps to normalize the input text for better performance.	Deep Voice 3 contributes to the field of speech synthesis by introducing a more efficient and scalable architecture, demonstrating the effectiveness of attention mechanisms in TTS, and providing high-quality speech synthesis with the ability to handle large-scale deployment.
textual features	low-dimensional audio representation in the form of mel-scale spectrograms		

Positive Impact of this Solution in This Project Domain		Negative Impact of this Solution in This Project Domain
Deep Voice 3 contributes to the advancement of TTS technology by addressing common errors, providing flexibility in waveform synthesis, supporting multispeaker synthesis, and delivering superior audio quality.		It's important to note that Deep Voice 3 addresses some of these issues, such as monotonic attention behavior, and offers scalability and faster training compared to previous systems. However, these potential negative impacts should be considered when using Deep Voice 3 or any other TTS system.
Analyse This Work By Critical Thinking	The Tools That Assessed this Work	What is the Structure of this Paper
The work presents a novel approach to neural TTS and demonstrates promising results in terms of speech quality and scalability. However, it also acknowledges areas for future improvement, such as optimizing inference further and exploring smaller models and fixed-precision arithmetic.	CrowdMOS toolkit for Mean Opinion Score (MOS) ratings and the Gentle toolkit for preprocessing and splitting long utterances.	<p>Abstract</p> <ul style="list-style-type: none"> • Introduction • Related Work • Deep Voice 3 Model • Error Modes in Sequence-to-Sequence Speech Synthesis • Waveform Synthesis Methods • Multispeaker Speech Synthesis • Deep Voice 3 System • Experimental Setup • Results • Conclusion • Reference
Diagram/Flowchart		
Deep Voice 3 uses residual convolutional layers to encode text into per-timestep <i>key</i> and <i>value</i> vectors for an attention-based decoder. The decoder uses these to predict the mel-scale log		

magnitude spectrograms that correspond to the output audio. (Light blue dotted arrows depict the autoregressive process during inference.) The hidden states of the decoder are then fed to a converter network to predict the vocoder parameters for waveform synthesis

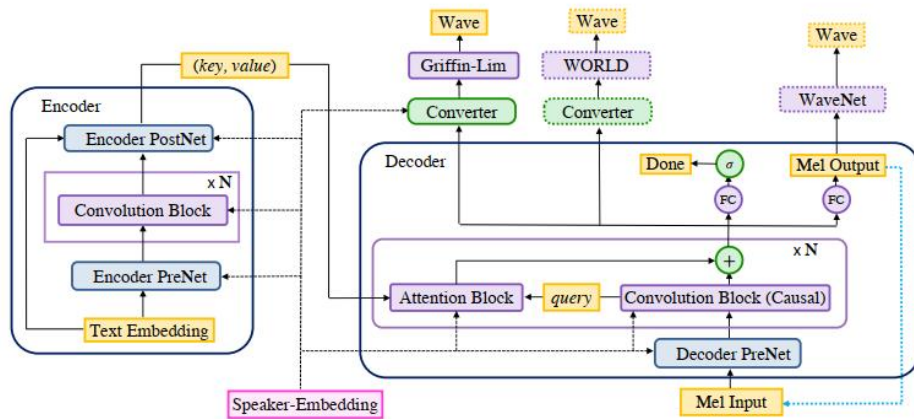


Figure 6. Deep Voice 3

--End of Paper 9--

10		
Reference in APA format	Blockchain-enabled End-to-End Encryption for Instant Messaging Applications	
URL of the Reference	Authors Names and Emails	Keywords in this Reference
https://in.docworkspace.com/d/sIKjDueGLAcbh w6oG	Raman Singh and Hitesh Tewari.	blockchain-enabled, end-to-end encryption, instant messaging applications.
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	The Goal (Objective) of this Solution & What is the problem that need to be solved	What are the components of it?
blockchain-based end-to-end encryption (E2EE) framework.	The goal of the proposed blockchain-based end-to-end encryption (E2EE) framework is to provide a real end-to-end encrypted	Mobile User: The user's device generates a public/private key pair during the application installation phase.

	<p>messaging service for instant messaging applications.</p>	<p>Mobile Network Operator (MNO): The MNO acts as a trusted third-party and issues a digital certificate for the user based on the information provided by them.</p> <p>Blockchain: The public-key digital certificates are stored on the blockchain. The blockchain maintains the validity of the certificates and allows users to fetch certificates for other users.</p> <p>Instant Messaging (IM) Server: The IM server verifies the user's certificate from the blockchain network. It does not control the encryption/decryption process and does not store any keys.</p> <p>Encryption/Decryption: Users encrypt and decrypt messages using their own public/private keys. The sender can fetch the receiver's digital certificate from the blockchain before encrypting a message.</p> <p>Backup and Restoration: Users generate their backup decryption key using a known secret and store the backup data on their cloud drive. When changing devices, users can</p>
--	--	--

		download their backups and decrypt them using the backup key.
--	--	---

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process

The proposed blockchain-based end-to-end encryption (E2EE) framework aims to address the privacy issues associated with current messaging applications by providing a secure and confidential communication environment.

	Process Steps	Advantage	Disadvantage (Limitation)
1	User Key Generation: During the installation phase of the messaging application, a mobile user creates a public/private key pair on their device. This step ensures that the user has control over their encryption keys.	Users have full control over their encryption keys, enhancing the security and privacy of their messages.	If the user loses their private key, they may lose access to their encrypted messages.
2	Digital Certificate Creation: A trusted third-party, such as a mobile network operator, creates a digital certificate for the user based on the information provided by them. This certificate serves as proof of the user's identity.	The involvement of a trusted third-party adds an additional layer of authentication and verification to the user's identity.	If the trusted third-party is compromised, it could lead to the compromise of the user's digital certificate and potentially their privacy.
3	Storing Certificates on the Blockchain: The user's public-key digital certificate is stored on the blockchain, ensuring its immutability and accessibility.	Storing certificates on the blockchain provides a decentralized and tamper-proof storage solution, enhancing the security and reliability of the certificates.	Storing large amounts of data on the blockchain can be resource-intensive and may impact scalability.
4	Certificate Verification: The messaging application can verify a user's certificate from the blockchain network before establishing a secure communication channel.	Certificate verification from the blockchain ensures the authenticity and integrity of the user's certificate, reducing the risk of impersonation or unauthorized access.	The verification process may introduce additional latency in establishing secure communication channels.

5	End-to-End Encryption: Once the sender fetches the receiver's digital certificate from the blockchain, they can encrypt the message using the receiver's public key.	End-to-end encryption ensures that only the intended recipient can decrypt and read the message, providing strong confidentiality.	If the sender's private key is compromised, an attacker may be able to decrypt the messages.

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable

Relationship Among The Above 4 Variables in This article

--

Input and Output		Feature of This Solution	Contribution & The Value of This Work						
<table><tr><th>Input</th><th>Output</th></tr><tr><td>Mobile user's device: The user creates a public/private key pair on their device during the application installation phase.</td><td>Encrypted message: The sender fetches the receiver's digital certificate from the blockchain and encrypts the message using the recipient's public key.</td></tr><tr><td>Digital certificate: A trusted third-party, such as a mobile network operator, creates a digital certificate for the user based on the</td><td>Decrypted message: The receiver decrypts the message using their private key.</td></tr></table>	Input	Output	Mobile user's device: The user creates a public/private key pair on their device during the application installation phase.	Encrypted message: The sender fetches the receiver's digital certificate from the blockchain and encrypts the message using the recipient's public key.	Digital certificate: A trusted third-party, such as a mobile network operator, creates a digital certificate for the user based on the	Decrypted message: The receiver decrypts the message using their private key.		Real End-to-End Encryption: Unlike some messaging apps that claim to provide end-to-end encryption but still involve servers in the encryption/decrypti on process, this framework ensures true end-to-end encryption. The server does not store any keys or participate in the encryption/decrypti	The contribution of this work is the proposal of a blockchain-enabled end-to-end encryption framework for instant messaging applications. The framework aims to provide secure and private communication by leveraging the decentralized nature of blockchain technology. It
Input	Output								
Mobile user's device: The user creates a public/private key pair on their device during the application installation phase.	Encrypted message: The sender fetches the receiver's digital certificate from the blockchain and encrypts the message using the recipient's public key.								
Digital certificate: A trusted third-party, such as a mobile network operator, creates a digital certificate for the user based on the	Decrypted message: The receiver decrypts the message using their private key.								

<p>information provided by them.</p> <p>Blockchain network: The user sends their public-key digital certificate to the blockchain network.</p>		<p>on process.</p> <p>Public Key Infrastructure (PKI): The framework utilizes a blockchain-based PKI system to provide secure and scalable digital certificates for users. This eliminates the need for a centralized authority and reduces the cost of issuing and maintaining digital certificates.</p> <p>Secure Messaging: The framework allows users to securely exchange messages by generating message keys using cryptographic mechanisms. The message keys are derived from chain keys, ensuring secure communication between users.</p> <p>Backup and Restoration: The framework provides a mechanism for users to backup and restore their data. Unlike traditional messaging apps where backup decryption keys are stored on application servers, this framework</p>	<p>eliminates the need for a centralized server to store encryption keys and ensures that only the intended recipients can decrypt the messages. The use of blockchain allows for the implementation of a large-scale public key infrastructure (PKI) system at a low cost. The value of this work lies in its potential to enhance the security and privacy of instant messaging applications, offering users a more secure communication experience.</p>
--	--	--	--

	<p>advises users to generate their own backup keys, enhancing the security of user data.</p> <p>Group Messaging: The proposed framework also supports secure group messaging. Group administrators can generate group keys and share them with group members over encrypted channels. Group messages are encrypted using the group key, ensuring confidentiality and authentication.</p>	
Positive Impact of this Solution in This Project Domain		Negative Impact of this Solution in This Project Domain
<p>the blockchain-enabled end-to-end encryption framework has the potential to significantly improve the security and privacy of instant messaging applications, providing users with a more secure and trustworthy communication platform.</p>		<p>Scalability: Blockchain technology is known to have scalability limitations, especially when it comes to handling a large number of transactions and blocks. The proposed framework would need to be tested for its scalability in a blockchain-based environment.</p> <p>Performance: Blockchain transactions can be slower compared to traditional centralized systems. The encryption and decryption processes may experience delays due to the involvement of blockchain</p>

	<p>operations, potentially impacting the real-time nature of instant messaging applications.</p> <p>User Experience: Implementing a blockchain-based framework may introduce additional complexity for users. They may need to understand and manage public/private key pairs, digital certificates, and blockchain operations, which could potentially impact the user experience and adoption of the application.</p> <p>Governance and Trust: While blockchain technology provides decentralized and transparent mechanisms, the governance and trust aspects of the proposed framework need to be carefully considered. The trustworthiness of the mobile network operator as a trusted third-party and the security of the blockchain network itself are crucial factors for the success of the framework.</p> <p>Compatibility and Interoperability: Integrating the proposed framework with existing messaging applications and platforms may pose compatibility and interoperability challenges. Ensuring seamless communication between users of different messaging applications could be a complex task.</p>	
Analyse This Work By Critical Thinking	The Tools That	What is the Structure of this

	Assessed this Work	Paper
<p>The work presents a critical analysis of the privacy concerns in online communications and proposes a blockchain-enabled solution to provide real E2EE. It addresses the limitations of the current PKI model and highlights the importance of user privacy in messaging applications.</p>	<p>Google Firebase: Used to implement the IM server in the proposed framework.</p> <p>Ethereum: Implemente d on the Docker platform to provide blockchain functionalit y.</p> <p>Docker Container: Used as a platform to implement the Ethereum blockchain.</p> <p>Android Emulators: Used to measure the performance of the AES256 encryption, HMAC calculation, and total encryption time in the Android application.</p>	<p>Abstract</p> <ul style="list-style-type: none"> ● Introduction ● Related Work ● System Architecture ● Phases of the Security Framework ● Implementation Details ● Evaluation ● Conclusion
Diagram/Flowchart		

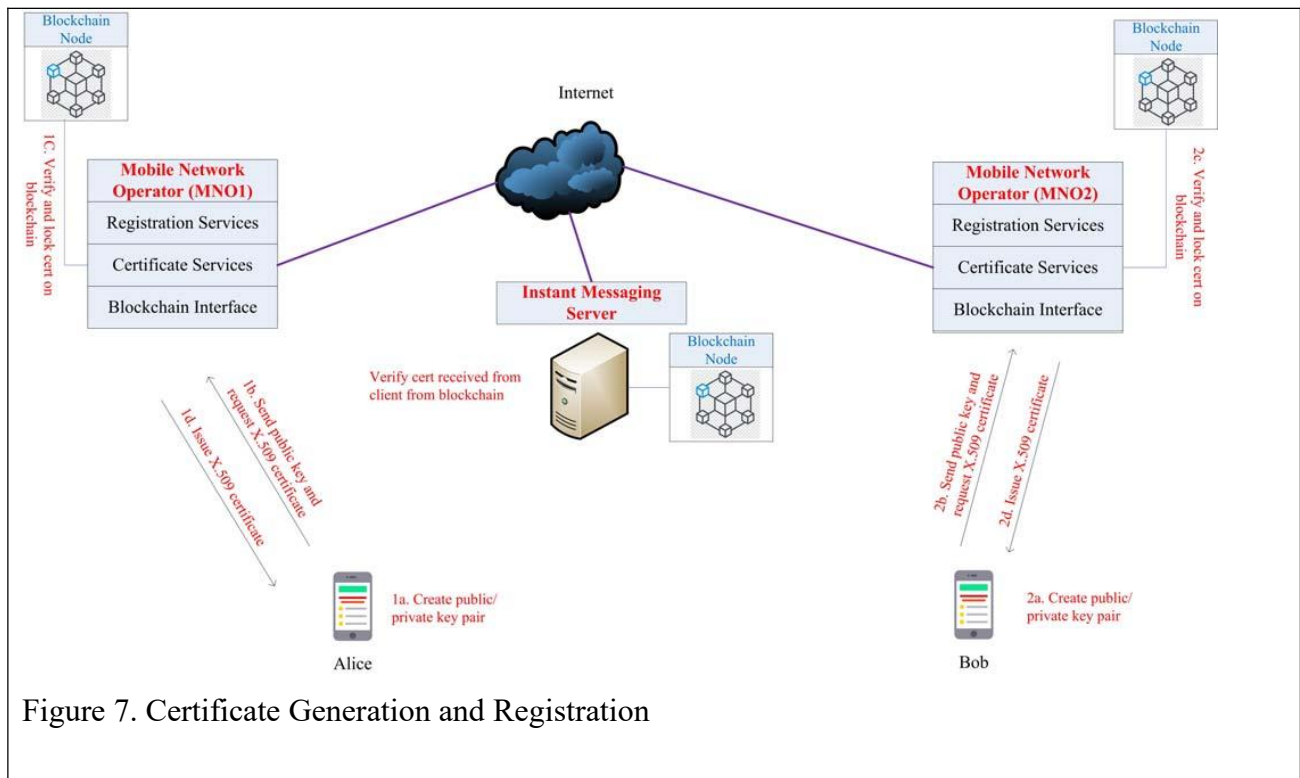


Figure 7. Certificate Generation and Registration

--End of Paper 10--

11		
Reference in APA format	REAL-TIME VOICE CLONING USING DEEP LEARNING: A CASE STUDY	
URL of the Reference	Authors Names and Emails	Keywords in this Reference
https://ijcrt.org/papers/IJCRT2305749.pdf	Hruthik B Gowda,Karun Datta Ramakumar,Sheethal.V,Sushma M,Dr. Madhusudhan G K	Voice Cloning, Deep Learning, Text to Speech Synthesis, Dimension Reduction ,Voice Morphing
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	The Goal (Objective) of this Solution & What is the problem that need to be solved	What are the components of it?
The approach includes three stages: voice cloning, text-to-speech synthesis, and speaker encoding	The aim is to develop a system that can produce speech in any given speaker's voice from given text input data.	Author used TTS Synthesis, WaveNet Vocoder, Neural Network Training to create a voice cloning system that can replicate speech in any given speaker's voice from

		supplied text input
The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process		

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable			
The dependent variables in this work are the synthesized voice products	The independent variables in this work are the speaker encoder network, the Tacotron 2 inspired sequence to sequence TTS synthesis network, and the Wave Net powered neural vocoder. These components are combined to create high-quality multispeaker TTS output.	none	none			
Relationship Among The Above 4 Variables in This article						
The synthesized voice products are dependent variables, influenced by and responsive to changes in the independent variables, which consist of the speaker encoder network, the Tacotron 2 inspired sequence to sequence TTS synthesis network, and the WaveNet-powered neural vocoder, as they collectively contribute to the creation of high-quality TTS output.						
Input and Output		Feature of This Solution	Contribution & The Value of This Work			
<table><tr><th>Input</th><th>Output</th></tr><tr><td>Textual data and sample audio</td><td>Speaker's voice generated from the supplied text input.</td></tr></table>	Input	Output	Textual data and sample audio	Speaker's voice generated from the supplied text input.	The key features of this solution include its ability to produce high-quality text-to-speech (TTS) output	This work is mostly lies in advancing the field of TTS by combining Sequence to sequence synthesis network.
Input	Output					
Textual data and sample audio	Speaker's voice generated from the supplied text input.					
Positive Impact of this Solution in This Project Domain		Negative Impact of this Solution in This Project Domain				
The positive impact of this solution is there is a better scope in advancement of technology on applications like voice assistants.		The major negative aspect of this solution is this model may give unsatisfactory results if the given audio has noise it.				
Analyse This Work By Critical Thinking		The Tools That Assessed this Work	What is the Structure of this Paper			

This work is good, as they tried to develop a voice cloning technique by utilizing Text to speech synthesis for speech generation	automated evaluation,Tacotr on2	<p>Abstract</p> <p>I. Introduction</p> <p>II. Literature review</p> <p>III. Methodology</p> <p>IV. Conclusion</p> <p>V. Reference</p>
---	---------------------------------	---

Diagram/Flowchart

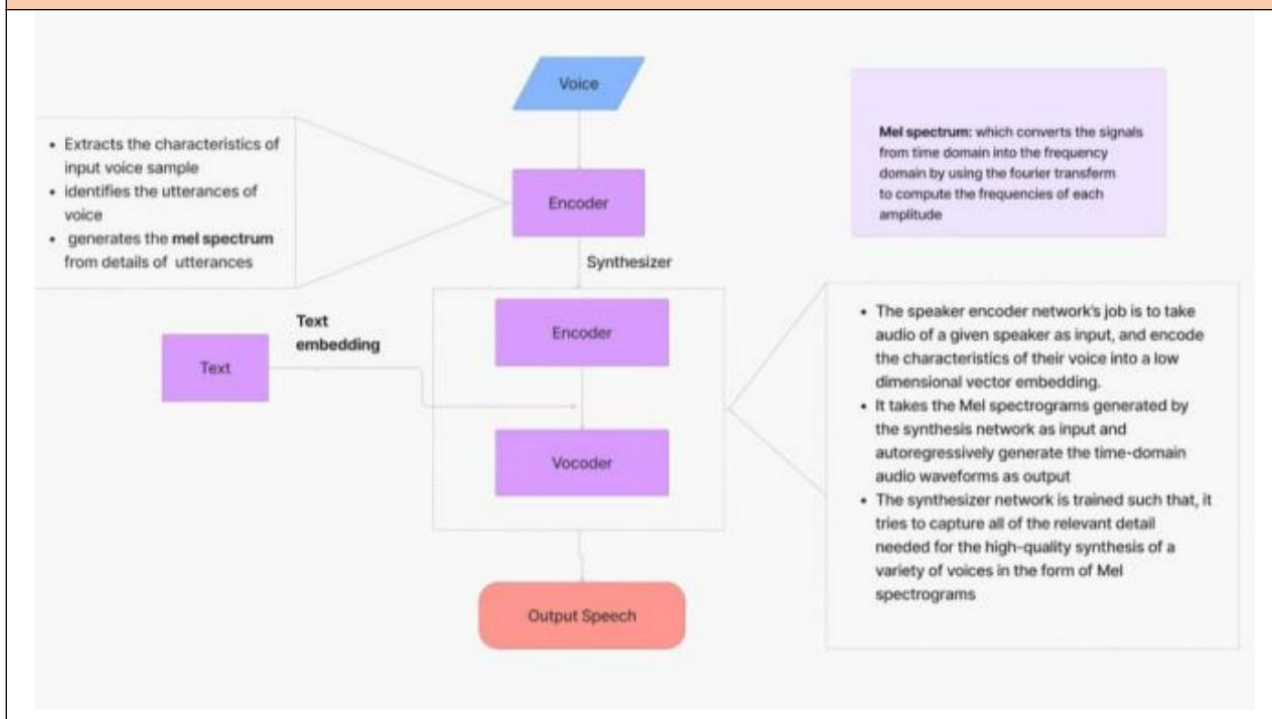


Figure 8. Text to speech Synthesizer

---End of Paper 11-

12		
Reference in APA format	Text To Speech Conversion Using Different Speech Synthesis	
URL of the Reference	Authors Names and Emails	Keywords in this Reference
file:///C:/Users/yogir/Downloads/Text-To-Speech-Conversion-Using-Different-Speech-Synthesis%20(1).pdf	Fahima Khanam , Farha Akhter Munmun , Nadia Afrin Ritu, Aloke Kumar Saha ,	Text to Speech (TTS), Domain specific synthesis, Phoneme based synthesis, Unit selection synthesis.
The Name of the Current Solution	The Goal	What are the components

(Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	(Objective) of this Solution & What is the problem that need to be solved	of it?
TTS(Text to Speech) Synthesizer for Speech Generation	The goal is to convert written text into natural and intelligible speech by using TTS.	NLP-The NLP includes text analysis, phonetic conversion, and prosodic phrasing.

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process

	Process Steps	Advantage	Disadvantage (Limitation)
1	<p>Natural Language Processing (NLP):</p> <p>Text Analysis: The input sentence is segmented into tokens, and each word is determined as part of speech (POS) using techniques like segmentation, text normalization, and POS tagging.</p> <p>Phonetic Conversion: Phonetic transcription is assigned to each word. This can be done using rule-based or dictionary-based approaches.</p> <p>Prosodic Analysis: This step determines the intonation, amplitude, and duration modelling of speech, which describes the speaker's emotion.</p>	NLP helps in analysing and understanding the input text, enabling accurate phonetic conversion and prosodic analysis.	NLP can be challenging for complex sentences, and errors in text analysis can affect the quality of speech.
2	<p>Digital Signal Processing (DSP):</p> <p>Speech Synthesis: This part focuses on generating the actual speech waveform. There are different technologies for this:</p> <p>Concatenative Synthesis: This technology uses a database of pre-recorded natural sounds and concatenates them to produce speech.</p> <p>Formant Synthesis: This technique does not have any database of</p>	DSP techniques like concatenative synthesis can produce natural-sounding speech	Concatenative synthesis may require a large database and memory capacity, while formant synthesis can sound artificial.

	speech samples so it sounds artificial and robotic.					
	Articulatory Synthesis: This technique synthesizes speech based on models of the human vocal tract. It can produce more natural speech but requires complex modelling.					
Major Impact Factors in this Work						
Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable			
The dependent variable discussed in this article are the output speech quality	The independent variables discussed in this article are: Text analysis, Phonetic conversion And Prosodic phrasing	none	none			
Relationship Among The Above 4 Variables in This article						
The variations in text analysis, phonetic conversion, and prosodic phrasing are expected to influence the output speech quality						
Input and Output		Feature of This Solution	Contribution in This Work			
<table><tr><td>Input</td><td>Output</td></tr><tr><td>Textual Data</td><td>Synthesized speech waveform that corresponds to the input text.</td></tr></table>	Input	Output	Textual Data	Synthesized speech waveform that corresponds to the input text.	One of the features of this solution is the use of unit selection speech synthesis, which selects an optimum set of sounding units from a speech database	The contribution of this work is the development of a text-to-speech (TTS) system that can generate natural and intelligible speech for numbers, words, and sentences.
Input	Output					
Textual Data	Synthesized speech waveform that corresponds to the input text.					
Positive Impact of this Solution in This Project Domain			Negative Impact of this Solution in This Project Domain			
One positive impact of this solution in the project domain is that the output speech is natural and intelligible, making it			One negative impact of this solution in the project domain is			

13	
Reference in APA format	REAL TIME VOICE CLONING
easier for blind people to access information through speech. Additionally, the system can be used in various applications such as teaching aids, text reading, and talking books/toys.	that the output speech for words may have discontinuities between transitions of phonemes. This can affect the overall smoothness and naturalness of the speech output.
Analyse This Work By Critical Thinking	<div>The Tools That Assessed this Work</div> <div>What is the Structure of this Paper</div>
The work presents an analysis of different synthesis methods for TTS systems and discusses the strengths and weaknesses of each approach.	<div>Domain specific synthesis, Phoneme based speech synthesis and Unit selection synthesis</div> <div> Abstract I. Introduction II. Methodology III. Implementation IV. Simulation Result V. Conclusion </div>
Diagram/Flowchart	
<pre> graph TD Start([Start]) --> InputWord[/Input Word/] InputWord --> PhoneticConversion[Phonetic Conversion] DictionaryBase[(Dictionary base)] --> PhoneticConversion PhoneticConversion --> CompareConcatenate[Compare and concatenate phoneme] RecordedPhonemeSounds[(Recorded phoneme sounds)] --> CompareConcatenate CompareConcatenate --> OutputSpeech([Output Speech]) OutputSpeech --> End([End]) </pre> <p>Figure 9. Flowchart of phoneme based text to speech</p>	

URL of the Reference	Authors Names and Emails	Keywords in this Reference
https://ijrti.org/papers/IJRTI2206107.pdf	Dr.T NAnitha, Amilio Dsouza, Ashutosh , Akshay Gole	Neural Networking, 3-stage pipeline, Text-to-Speech, Deep Learning.
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	The Goal (Objective) of this Solution & What is the problem that need to be solved	What are the components of it?
The current solution is called Real Time Voice Cloning.	The goal of the "Real Time Voice Cloning" is to develop a three-stage deep learning system that can clone a person's voice in real-time.	Three-level pipeline,Deep learning models,vocoder,reference speech,training data
The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process		
	Process Steps	Advantage
1	Feature Extraction: The first step is to extract features from the reference audio, which better represent the sound of the voice produced by the model.	The feature extractor provides data that better represents the voice produced by the model. It helps in capturing the important characteristics of the voice.
2	Acoustic Model: The acoustic model is like a smart tool that figures out how the features we calculate are connected to the actual sounds produced for a given piece of text.	It helps in generating speech that closely resembles the target speaker's voice.
3	Vocoder: The vocoder is a system that reconstructs audio waveforms from the acoustic features generated by the acoustic model. It takes the acoustic features as input and generates the corresponding audio waveform, which closely resembles the voice of the target speaker.	The vocoder is responsible for reconstructing audio waveforms from the acoustic features generated by the acoustic model. It helps in generating high-quality and natural-sounding speech.
Major Impact Factors in this Work		

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
The dependent variable mentioned in this paper is naturalness.	The Independent variables mentioned in this paper are TTS Pipeline, Training data	none	none

Relationship Among The Above 4 Variables in This article

The study explores the impact of TTS pipeline, and training data as independent variables on the perceived naturalness, the dependent variable, of the synthesized audio.

Input and Output		Feature of This Solution	Contribution & The Value of This Work			
<table><tr><th>Input</th><th>Output</th></tr><tr><td>Reference audio</td><td>Cloned voice of reference audio</td></tr></table>	Input	Output	Reference audio	Cloned voice of reference audio	One of the feature of this solution is Personalised voice interface	This work's key contribution lies in its development of a real-time voice cloning framework via text-to-speech synthesis, offering the valuable potential to advance voice cloning technology.
Input	Output					
Reference audio	Cloned voice of reference audio					

Positive Impact of this Solution in This Project Domain	Negative Impact of this Solution in This Project Domain
One positive impact of solution in this project domain is that it allows for the replication of unseen voices from just a few seconds of reference speech. This means that with minimal input, the system is able to generate highly natural sounding cloned voices. This can be beneficial in various applications such as personalized voice interfaces.	One negative impact of solution in this project domain is that the cloned voice may lack naturalness and accent. This means that the cloned voice may not sound as natural or have the same accent as the original human voice.

Analyse This Work By Critical Thinking	The Tools That Assessed this Work	What is the Structure of this Paper
--	-----------------------------------	-------------------------------------

This work on voice cloning appears to be a promising and innovative approach in the field of text-to-speech synthesis. The authors have identified the limitations of existing models and have proposed a three-stage pipeline to address these limitations.	The tools used to assess this work include the SV2TTS database, Google Colab.	Abstract I. Introduction II. Literature survey III. Proposed system IV. Implementation V. Result and Conclusion
--	---	--

Diagram/Flowchart

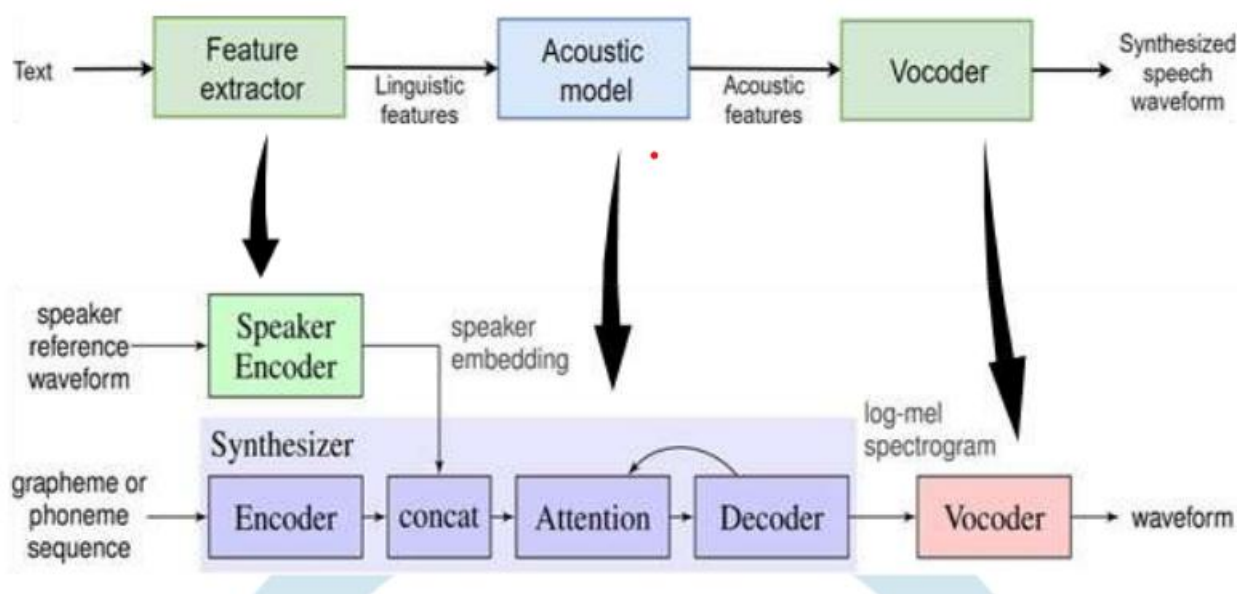


Figure 10. System Block Diagram

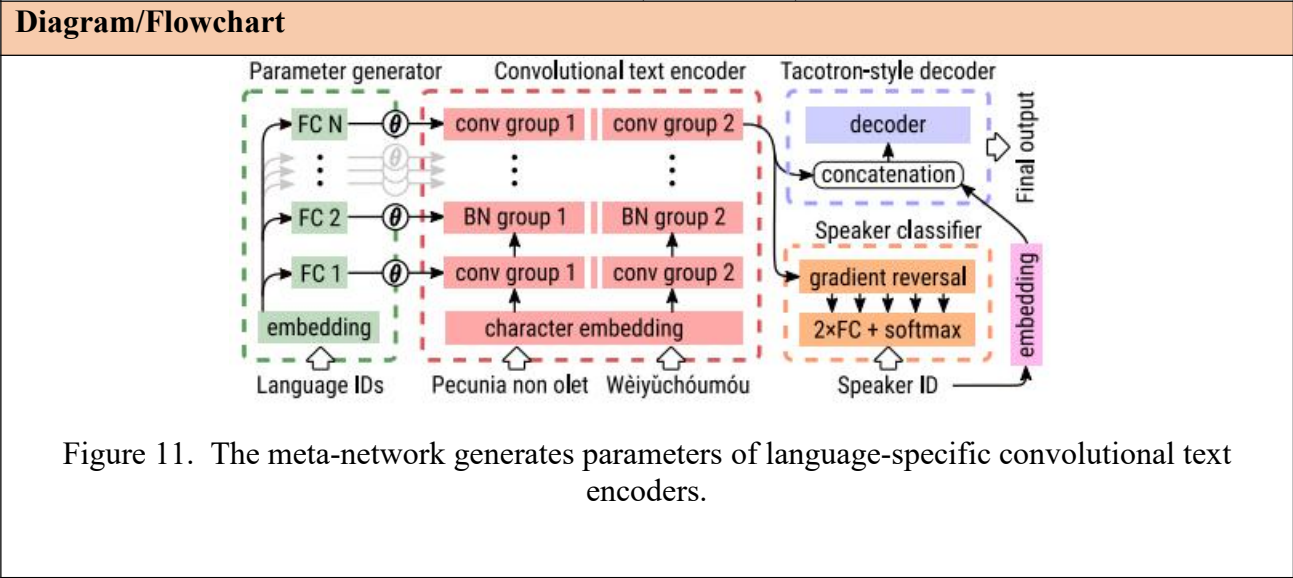
--End of Paper 13--

4		
Reference in APA format	One Model, Many Languages: Meta-learning for Multilingual Text-to-Speech	
URL of the Reference	Authors Names and Emails	Keywords in this Reference
https://paperswithcode.com/paper/one-model-many-languages-meta-learning-for	Tomáš Nekvinda, Ondřej Dušek	text-to-speech, speech synthesis, multilinguality, code-switching, meta-learning, domain-adversarial training
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/	The Goal (Objective) of this Solution & What is	What are the components of it?

Framework/ ... etc)	the problem that need to be solved	
The current solution discussed in the article is a multilingual text-to-speech (TTS) model.	The goal of the multilingual text-to-speech (TTS) model discussed in the document is to enable natural-sounding speech synthesis in multiple languages using less training data.	The multilingual text-to-speech (TTS) model discussed in the document consists of the following components : Input Text Encoder,Decoder,Convolutional Encoder,Parameter Generator Network
The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process		
	Process Steps	Advantage
1	Model Architecture: The researchers base their experiments on Tacotron 2, a TTS model. They introduced some changes to the architecture, including the use of convolutional encoders, parameter generation conditioned on language embeddings, and training with multilingual batches.	The advantage of using Tacotron 2 as the base model is that it provides a solid foundation for TTS
2	Convolutional Encoders: Instead of having separate encoders for each language, the researchers use multiple language-specific input text encoders as they are used for better Batch Normalization	The advantage of using convolutional encoders is that they can effectively process the input text
3	Encoder Parameter Generation: To enable cross-lingual knowledge sharing, parameters of the encoders are generated using a separate network conditioned on language embeddings. This allows for controllable cross-lingual parameter sharing.	The advantage of generating parameters conditioned on language embeddings is that it enables cross-lingual knowledge sharing.
Major Impact Factors in this Work		

Dependent Variable		Independent Variable		Moderating variable
The dependent variables discussed in this article are pronunciation accuracy, voice quality.		Grapheme Encoder Network(GEN)		none
Relationship Among The Above 4 Variables in This article				
In this article, the dependent variable is the performance of the multilingual text-to-speech (TTS) pronunciation accuracy. The independent variables are the different models and approaches used (GEN) model .The article explain how these independent variables affect the dependent variable different models and approaches.				
Input and Output		Feature of This Solution	Contribution & The Value of This Work	
Input	Output	One of the features of this solution is the use of multilingual training batches to fully utilize the potential of the architecture.	This work contributes to the advancement of multilingual TTS synthesis and provides insights into the effectiveness of different approaches for handling multilingual data.	
Text in specific language	Natural sounding Multilingual Speech			
Positive Impact of this Solution in This Project Domain			Negative Impact of this Solution in This Project Domain	
This solution enables better knowledge sharing, expands voice cloning capabilities, and facilitates code switching, leading to improved performance and versatility in the project domain.			The major negative aspect in this solution is this model is prone to error generation of speech while working with complex language scripts like Chinese	
Analyse This Work By Critical Thinking		The Tools That Assessed this Work	What is the Structure of this Paper	
The authors highlight the need for cross-lingual knowledge-sharing in multilingual text-to-speech (TTS) systems. They mention that previous work in this area is limited, and they aim to address this gap by proposing a scalable grapheme-based model.		The tools used to assess this work include	Abstract I. Introduction II. Related Work III. Model Architecture IV. Dataset	

	character error rate (CER) evaluation and Google Cloud Platform.	V. Experiments VI. Conclusion and Acknowledgment
--	--	---



--End of Paper 14--

15	Development of chat application	
Reference in APA format	Authors Names and Emails	Keywords in this Reference
URL of the Reference		
https://www.ijraset.com/research-paper/development-of-chat-application	Dr. Abhay Kasetwar, Ritik Gajbhiye, Gopal Papewar, Rohan Nikhare, Priya Warade	Javascript , React.js, Internet
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	The Goal (Objective) of this Solution & What is the problem that need to be solved	What are the components of it?
To Build a chat Application	The goal of the chat application is to provide a reliable and flexible chat system that allows users to communicate in real-time.	Java script,Internet,Application Registration Page,Message editing field with keyboard
The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage		



& Disadvantage of Each Step in This Process

	Process Steps	Advantage	Disadvantage (Limitation)
1	Define the Purpose and Goals	This helps to determine the purpose of the app	
2	Plan the features and functionalities	This Helps to make a list of features and functionalities that the app will have. Prioritize them based on their importance and feasibility.	
3	Design the User Interface: Design the user interface (UI) of the app.	A well-designed and intuitive UI enhances the overall user experience, making the app more enjoyable and engaging.	
4	Backend development: Set up the server, database, and APIs required for the app's functionality.	Backend development provide overall Functionalities for Smooth operation of app	More Complex and time Taking
5	Front end Development: Develop the frontend of your app using programming languages like HTML, CSS, and JavaScript. Implement the UI design and integrate it with the backend	Frontend development enhances user experience and engagement by creating visually appealing and interactive interfaces for seamless interaction with the app.	Need to be compatible with both app and the browser
6	Test and debug Conduct thorough testing to identify and fix any bugs or issues in the app.	Testing and debugging are used for identifying and fixing potential issues early in the development process, ensuring a more reliable and robust application.	Complex and time consuming

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
none	none	none	none

Relationship Among The Above 4 Variables in This article

NA							
Input and Output		Feature of This Solution	Contribution & The Value of This Work				
<table><tr><th>Input</th><th>Output</th></tr><tr><td>User registration details, login credentials, text messages, and search queries</td><td>Display of messages, and the transmission of text messages to other users</td></tr></table>		Input	Output	User registration details, login credentials, text messages, and search queries	Display of messages, and the transmission of text messages to other users	This is a Two way communication system which includes basic app functionalities like Notifications and statistics for unread messages, Saving past messages etc.	The contribution of this work is the development of a real-time messaging app using modern web technologies. Unlike most chat apps available in the market, this app focuses on developers and aims to increase their productivity.
Input	Output						
User registration details, login credentials, text messages, and search queries	Display of messages, and the transmission of text messages to other users						
Positive Impact of this Solution in This Project Domain			Negative Impact of this Solution in This Project Domain				
The positive impact of this chat application in the project domain is that it focuses on developers and aims to increase their productivity. By providing a real-time messaging platform, it allows for easier and faster communication between developers			Development of Chatting application is Complex and Time taking and another limitation is Internet is required in order to use the application				
Analyse This Work By Critical Thinking		The Tools That Assessed this Work	What is the Structure of this Paper				
The work seems to be a well-planned project that addresses the needs of developers and aims to provide a reliable and secure chat system.		JavaScript, React.js, MongoDB, Express.js, and Node.js	Abstract I. Introduction II. Problem Statements III. Aim IV. Objective V. Components for App development VI. Conclusion and Future Scope				
Diagram/Flowchart							
<div><div> sender</div><div>→ message →</div><div>Chat service: 1. store message 2. relay message</div><div>→ message →</div><div> receiver</div></div>							
Figure 12. Shows the relationship between clients(server and receiver) and the chat services.							

16		
Reference in APA format	Real Time Voice Cloning	
URL of the Reference	Authors Names and Emails	Keywords in this Reference
https://ijirt.org/Article?manuscript=151003	Sakith Nalluri, A.Rohan Sai, M.Saraswati	Text-to-speech synthesis, Natural Language Processing, Digital Signal Processing.
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	The Goal (Objective) of this Solution & What is the problem that need to be solved	What are the components of it?
TextToSpeech Robot	The TextToSpeech Robot solution addresses accessibility, aiding visually impaired or reading-disabled users by converting text to speech. It also allows saving converted text as audio files locally.	The Main Application Module manages GUI, basic operations, and parameter input through file, keyboard, or browser. Integrated with it, the Conversion Engine Module utilizes the free TTS API for text-to-speech synthesis.
The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process		

The TextToSpeech Robot solution converts text into speech using a text-to-speech (TTS) engine. The process involves two main phases: text analysis and speech waveform generation.

	Process Steps	Advantage	
1	Text Input: The user can input text either by typing it into the text field or by copying it from an external document and pasting it into the application.	Accessibility: It provides a useful tool for people with visual impairments, allowing them to convert text into speech and listen to it.	qual synt depe Som spee robo
2	Conversion: The input text is converted into speech using the TTS (Text-to-Speech) functionality. The application uses the open-source API called c# for this conversion.	Ease of use: The application has a simple and user-friendly interface, making it easy for users to input text and convert it into speech.	Lim and proc punc to p outp

Major Impact Factors in this Work

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable

Relationship Among The Above 4 Variables in This article

Input and Output	Feature of This Solution	Contribution & The Value of This Work
------------------	--------------------------	---------------------------------------

		Our software is called the TextToSpeech Robot, a simple application with the text to speech functionality.The main features of the TextToSpeech Robot solution include:	The TextToSpeech Robot solution aims to improve the similarity and naturalness of generated speech, and it can be further enhanced and expanded upon in future developments.
Input	Output	<ul style="list-style-type: none">● Text-to-speech functionality● User-friendly interface● Input options● Saving audio files	The solution utilizes deep learning networks and advanced audio processing techniques to generate natural-sounding speech
The input of the TextToSp eech Robot solution is text, which can be entered directly into the applicatio n's text field or copied from an external document .		The output of the solution is speech, where the text is convert ed into synthesi zed speech that can be heard by the user.	
Positive Impact of this Solution in This Project Domain			Negative Impact of this Solution in This Project Domain
This can be beneficial for individuals with visual impairments as it allows them to access and consume large volumes of text more easily			The generated speech may not sound completely natural or human-like, which can affect the user experience and make it less engaging or enjoyable.
Analyse This Work By Critical Thinking		The Tools That Assessed this Work	What is the Structure of this Paper
It mainly discusses the development of a framework for real-time voice cloning using deep learning networks and text-to-speech synthesis. It mentions the successful implementation of the framework and the potential for further improvement		NONE	Abstract I. INTRODUCTION II. LITERATURER SURVEY III. STRUCTURE OF A TEXT-TO-SPEECH IV. PROPOSED WORK V. METHODOLOGY VI. CONCLUSION
Diagram/Flowchart			

---End of Paper 16---

17		
Reference in APA format	VOICE CLONING: A MULTI-SPEAKER TEXT-TO-SPEECH SYNTHESIS APPROACH BASED ON TRANSFER LEARNING	
URL of the Reference	Authors Names and Emails	Keywords in this Reference
https://arxiv.org/pdf/2102.05630.pdf	Giuseppe Ruggiero, Enrico Zovato, Luigi Di Caro, Vincent Pollet	End-to-end text-dependent speaker verification, deep neural networks, 1D convolution neural networks, gated recurrent neural networks, neural audio synthesis.
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	The Goal (Objective) of this Solution & What is the problem that need to be solved	What are the components of it?
Advanced Gru Network.	The goal of the "Advanced Gru Network" solution is to build a Text-to-Speech (TTS) system that can generate natural speech for a wide variety of speakers.	<ul style="list-style-type: none"> • 1 Conv1D layer • 3 GRU layers • Linear projection layers
The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process		

	Process Steps	Advantage	Disadvantage (Limitation)
1	Input Processing: The input texts are converted into phoneme sequences, and the target mel spectrogram features are computed on 50 ms signal windows, shifted by 12.5 ms, and passed through an 80-channel mel-scale filterbank.	Faster Training: Compared to the GRU network, the "Advanced Gru Network" is faster during training, which can save computational time.	The linear projection layer may introduce additional computational complexity to the model and increase the number of parameters that need to be trained.
2	Model Architecture: The Advanced Gru Network consists of 1 Conv1D layer and 3 GRU layers, each followed by a linear projection layer. The Conv1D layer performs convolutional operations on the input features, and the GRU layers capture the temporal dependencies in the data.	Improved Speaker Verification: The "Advanced Gru Network" achieved the best Speaker Verification Equal Error Rate (SV-EER) on the test set, indicating its effectiveness in accurately verifying speakers.	The main disadvantage of the Conv1D layer is that it may not capture long-range dependencies effectively, as it operates on local regions of the input.
3	GRU Layers: The network then includes three GRU layers, which are recurrent neural network layers that can capture long-term dependencies in sequential data. Each GRU layer is followed by a linear projection layer.		GRU layers may struggle with capturing very long-term dependencies in the data. They may also suffer from the vanishing gradient problem, which can affect the training process.
Major Impact Factors in this Work			
Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
synthesized speech waveform.	mel spectrogram feature vector	None	None

Relationship Among The Above 4 Variables in This article

The dependent variable is the synthesized speech waveform, this means that in study the synthesized speech waveform is the output or result that you are specifically interested in analyzing, measuring, or evaluating.

The independent variable is the mel spectrogram feature vector, this means that the mel spectrogram feature vector is the input data or parameter that you are manipulating or varying in order to observe its effect on the synthesized speech waveform, which is the dependent variable, commonly used in speech processing tasks.

The moderating variable is the speaker encoder, which computes an embedding vector that characterizes the voice of the speaker and conditions the synthesis process.

The mediating (intervening) variable is the synthesizer, which takes the input text and the embedding vector and predicts the mel spectrogram.

Input and Output		Feature of This Solution	Contribution in This Work			
<table><tr><th>Input</th><th>Output</th></tr><tr><td>The input of the "Advanced Gru Network" model is a sequence of mel spectrogram frames</td><td>the output is a fixed-dimensional embedding vector that represents the speaker characteristics in the transformed space.</td></tr></table>	Input	Output	The input of the "Advanced Gru Network" model is a sequence of mel spectrogram frames	the output is a fixed-dimensional embedding vector that represents the speaker characteristics in the transformed space.	<p>The "Advanced GRU Network" solution is a speaker encoder model that combines the advantages of convolution and GRU networks. It consists of 1 Conv1D layer and 3 GRU layers, each followed by a linear projection layer. It achieved the best Speaker Verification Equal Error Rate (SV-EER) on the test set compared to other models.</p>	<p>This model was found to be faster during training compared to the GRU network and achieved the best Speaker Verification Equal Error Rate (SV-EER) on the test set. The advanced GRU network was able to create a robust space of internal features that effectively separated speakers based on their utterances.</p>
Input	Output					
The input of the "Advanced Gru Network" model is a sequence of mel spectrogram frames	the output is a fixed-dimensional embedding vector that represents the speaker characteristics in the transformed space.					
Positive Impact of this Solution in This Project Domain		Negative Impact of this Solution in This Project Domain				
<p>The goal of this work is to build a TTS system which can generate in a data efficient manner natural speech for a wide variety of speakers, not necessarily seen during the training phase.</p>		<p>Lack of human-level naturalness.</p> <p>Inability to reproduce speaker prosody.</p> <p>Data limitations</p>				
Analyse This Work By Critical Thinking		The Tools That Assessed this Work	What is the Structure of this Paper			

The authors have provided detailed information about the methodology, including the use of deep neural networks and attention mechanisms	UMAP Mean Opinion Score (MOS)	<ul style="list-style-type: none"> ➤ Introduction ➤ Model Architecture ➤ Experiments and Results ➤ Conclusions ➤ References
--	----------------------------------	--

Diagram/Flowchart

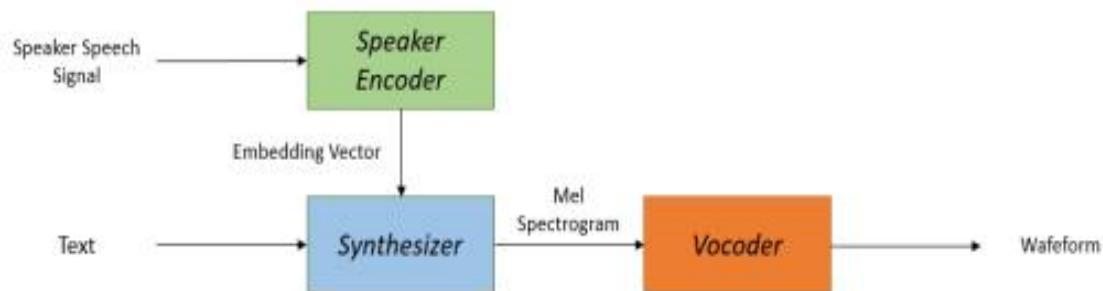


Figure 13. Voice cloning: A Multi-Speaker Text to Speech Synthesis approach based on transfer learning.

--End of Paper 17--

18		
Reference in APA format	Text to Speech Conversion with Emotion Detection	
URL of the Reference	Authors Names and Emails	Keywords in this Reference
https://www.ripublication.com/ijaer18/ijaerv13n14_23.pdf	Anita , Srinivasan.	Overall, the techniques used involve analyzing sentence patterns, assigning emotion constants to words, and matching words with audio files in the multimedia database based on their assigned emotions.

The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)		The Goal (Objective) of this Solution & What is the problem that need to be solved	What are the components of it?
data collection, pattern recognition, output audio generation and validation.		To create a text-to-speech conversion, analyze the emotions present in textual data.	Emotion Detection, Grammar Identification, Text-to-Speech Conversion, Pattern Recognition.
The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process			
	Process Steps	Advantage	Disadvantage (Limitation)
1	Emotion Detection, Grammar Identification, Text-to-Speech Conversion, Audio Arrangement	Helps in understanding the emotional context of the textual data. Ensures proper sentence structure and improves the quality of the speech output.	May not accurately detect complex emotions or subtle nuances. May not handle all grammar rules or variations in sentence structure.
Major Impact Factors in this Work			
Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
The emotion associated with a word or sentence.	patterns of emotion, the strength of the words in the sentence, the position of the words.	The process of emotion detection from text and converting it into speech.	None
Relationship Among The Above 4 Variables in This article			
The article discusses the relationship among the variables of emotion categories, sentence pattern, emotion strength, text analysis, machine learning, and the accuracy and performance of predictive models.			
Input and Output		Feature of This Solution	Contribution & The Value of This Work

<table><tr><th>Input</th><th>Output</th></tr><tr><td>Takes input in the form of a sentence with an emotional word</td><td>output is a synthesized speech that conveys the emotional content of the input sentence</td></tr></table>		Input	Output	Takes input in the form of a sentence with an emotional word	output is a synthesized speech that conveys the emotional content of the input sentence	Accurate and Reliable Data, User-Friendly Interface, Handling Complex Sentences and Emotions.	This work efficiently handles complex sentences, saving time in text-to-speech conversion, providing accurate information with emotional detection. It's valuable for the information age's accuracy expectations.
Input	Output						
Takes input in the form of a sentence with an emotional word	output is a synthesized speech that conveys the emotional content of the input sentence						
Positive Impact of this Solution in This Project Domain			Negative Impact of this Solution in This Project Domain				
Improved data understanding, user interaction, robotics applications, accuracy, time efficiency, and future enhancements.			Limits in emotional range, database dependency, context understanding, and training				
Analyse This Work By Critical Thinking		The Tools That Assessed this Work	What is the Structure of this Paper				
This paper explores emotion detection in text, converting it to speech for human-like machine interaction. Utilizing complex algorithms, the system achieves 100% accuracy with ample training data, offering potential enhancements for emotional voice outputs in robotics, promising more engaging machines.		Vector Space Model (VSM), Naïve Bayes classifier, Support Vector Machine (SVM), unsupervised machine learning approach.	Abstract <ol style="list-style-type: none">1. Introduction2. Literature Review3. Methodology4. Results and Discussion5. Conclusion6. References				
Diagram/Flowchart							

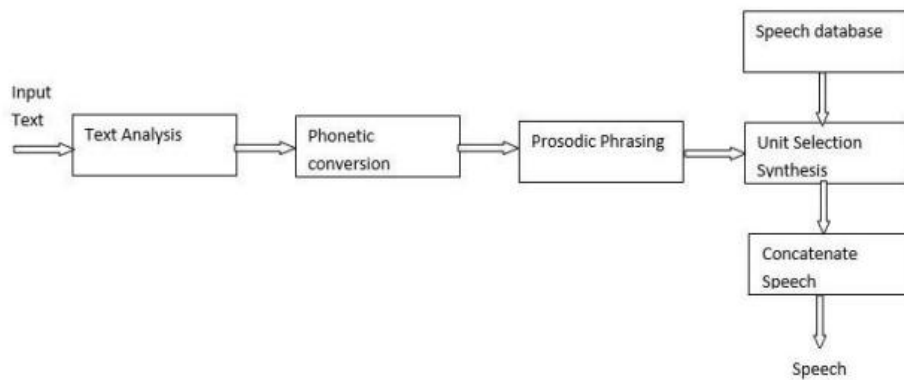


Figure 14. Text to Speech Conversion with Emotion Detection

--End of Paper 18--

19		
Reference in APA format		
URL of the Reference	Authors Names and Emails	Keywords in this Reference
https://www.researchgate.net/publication/322509087_Developing_an_End-to-End_Secure_Chat_Application	Noor Sabah Jamal M. Kadhim Ban N. Dhannoon	Developing an End-to-End Secure Chat Application, Two-step verification, MongoDB ..etc
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	The Goal (Objective) of this Solution & What is the problem that need to be solved	What are the components of it?

Chat application that aims to preserve the security and privacy of the chat communication	The goal is to solve the problem of security and privacy concerns, To provide end-to-end security for users to safely exchange private information without worrying about data leakage.	Client-side Server-side
---	---	----------------------------

The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process

- Registration
- Login
- Message Encryption

	Process Steps	Advantage	Disadvantage (Limitation)
1	Users need to register an account by providing their name, email, and password.	Provides a secure container to store the local storage key, making it difficult for unauthorized access.	The encryption algorithm used for password encryption may have vulnerabilities that could be exploited by attackers.
2	Users authenticate themselves by providing their email and password. The password is encrypted and sent to the server for validation.	Ensures user authentication and generates a JWT for secure communication.	If the JWT is compromised, an attacker could impersonate the user and gain unauthorized access.
3	Each message has its own separate key and nonce for better security.	Ensures the confidentiality and integrity of messages by encrypting them and computing a MAC.	The encryption algorithm used may have vulnerabilities that could be exploited by attackers.
Major Impact Factors in this Work			

- Security and Privacy
- Encryption Algorithms
- Keystore and Local Storage
- Client-Server Architecture

Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
<ul style="list-style-type: none"> ➤ Security and Privacy of the Chat Application ➤ Message Encryption ➤ End-to-end Security 	<ul style="list-style-type: none"> ➤ Registration Information ➤ Encryption Algorithm ➤ Authentication Information 	<ul style="list-style-type: none"> ➤ End-to-End Security ➤ Secure Storage 	<ul style="list-style-type: none"> ➤ MongoDB ➤ Node.js

Relationship Among The Above 4 Variables in This article

In this variable there is a strong relationship among security and privacy measures, storage protection, speed and performance, and user interface and user experience.

Input and Output

Feature of This Solution

Contribution & The Value of This Work

Input	Output
Text messages, images, or files that they want to send to another user.	encrypts the input messages to ensure security and privacy.

Secure chat: End-to-End Encryption, secure authentication, local storage, TLS, no server storage, two-factor authentication.

The contribution of this work is the development of an end-to-end secure chat application, ensures the security and privacy of user communications by implementing various encryption algorithms and secure storage mechanisms.

Positive Impact of this Solution in This Project Domain

Negative Impact of this Solution in This Project Domain

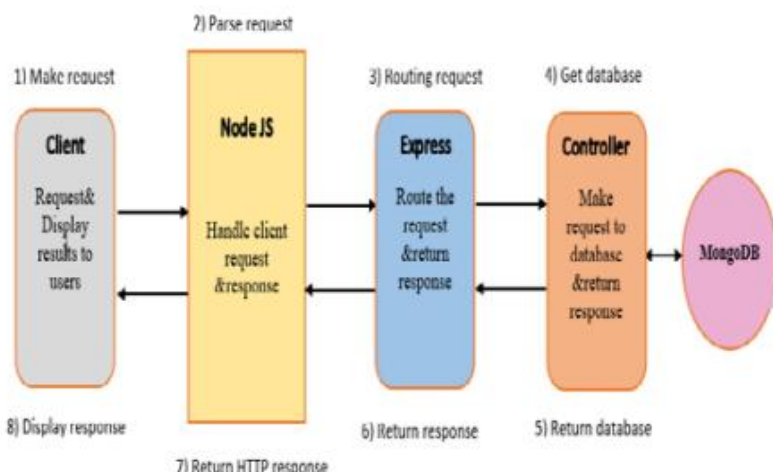
It ensures that messages exchanged between users are encrypted and can only be read by the sender and receiver, without the involvement of any third party.	Compatibility issues, open-source evaluation absence, limited features, user adoption challenges, performance issues, security and privacy concerns.	
Analyse This Work By Critical Thinking	The Tools That Assessed this Work	What is the Structure of this Paper
The importance of security and privacy in chat applications that addresses these concerns by implementing modern methods and lightweight algorithms and providing confidence to clients that their messages are protected even if their mobile phones are compromised.	None	Abstract I. Introduction II. Mobile Chat Applications III. Proposed architecture IV. Analysis the Proposed Chat V. Conclusion
Diagram/Flowchart		
 <pre>graph LR Client[Client] -- "1) Make request" --> NodeJS[Node JS] NodeJS -- "2) Parse request" --> Express[Express] Express -- "3) Routing request" --> Controller[Controller] Controller -- "4) Get database" --> MongoDB[(MongoDB)] MongoDB -- "5) Return database" --> Controller Controller -- "6) Return response" --> Express Express -- "7) Return HTTP response" --> NodeJS NodeJS -- "8) Display response" --> Client</pre> <p>The flowchart illustrates the architecture of the chat application. It starts with a Client (grey box) making a request (1) to Node JS (yellow box). Node JS parses the request (2) and sends it to Express (blue box). Express routes the request (3) to the Controller (orange box). The Controller gets the database (4) from MongoDB (pink oval) and returns the database (5) to the Controller. The Controller then returns a response (6) to Express, which returns an HTTP response (7) to Node JS. Finally, Node JS displays the response (8) to the Client.</p>		

Figure 15. providing security using end to end encryption.

Figure 15. providing security using end to end encryption.

20		
Reference in APA format	Voice Generation Using Deep Learning	
URL of the Reference	Authors Names and Emails	Keywords in this Reference
https://www.academia.edu/91460398/Voice_generation_using_deep_learning	Gonzalo Gómez Sánchez	Text-to-Speech Systems, Accessibility Tools, Multimedia Production, Language Learning and Education.
The Name of the Current Solution (Technique/ Method/ Scheme/ Algorithm/ Model/ Tool/ Framework/ ... etc)	The Goal (Objective) of this Solution & What is the to problem that need be solved	What are the components of it?
Deep Neural Networks	The solution is to develop a system for voice generation using deep learning	Deep neural networks (DNNs), LSTM.
The Process (Mechanism) of this Work; Means How the Problem has Solved & Advantage & Disadvantage of Each Step in This Process		
<ul style="list-style-type: none"> ○ Acoustic Feature Extraction ○ DNN Training ○ Audio Waveform Generation 		

	Process Steps	Advantage	Disadvantage (Limitation)
1	Acoustic features, extracted from speech waveform and text transcriptions, serve as DNN input capturing speech characteristics.	DNN system enhances voice with natural, intelligible quality surpassing tradition	Training and optimizing deep neural networks can be computationally expensive, requiring significant computational resources.
2	DNN training optimizes network to learn mapping between acoustic features and audio waveform	DNNs optimize synthesis system, eliminating separate steps like feature extraction for efficiency	System produces intelligible audio, but voice quality may lag behind state-of-the-art
3	If the DNN system is trained, it can be used to generate the audio waveform directly from text input.	Success could yield a text-to-voice Deep Learning system, bypassing pre-recorded data.	Research needed to enhance proposed system's performance and quality.
Major Impact Factors in this Work			
Dependent Variable	Independent Variable	Moderating variable	Mediating (Intervening) variable
Audio Waveforms	Acoustic Features	None	None
Intelligibility of Audio	Datasets	None	None
Text-to-Speech System	Pseudo-Quadrature Mirror Filters	None	None
Relationship Among The Above 4 Variables in This article			

Input and Output		Feature of This Solution	Contribution & The Value of This Work
Input	Output	Use of Deep Neural Networks (DNN), Investigation of different architectures, Generation of audio waveform	The thesis contributes to speech synthesis through deep learning, proposing models for intelligible audio, introducing Pseudo-Quadrature Mirror Filter banks for efficiency, and emphasizing computational cost and audio quality considerations for future advancements.
Text generates audio waveform parameters.	System outputs audio waveform via parameter-based vocoder.		
Positive Impact of this Solution in This Project Domain			Negative Impact of this Solution in This Project Domain

The audio signal in speech synthesis can lead to improved quality, simplified system architecture, parallelization, and the potential for more advanced text-to-speech systems.		Proposed solution faces high computational cost, audio distortion, limiting scalability.
Analyse This Work By Critical Thinking	The Tools That Assessed this Work	What is the Structure of this Paper
It explores deep learning for voice generation, proposing architectures with limitations. Parallelization reduces computational costs, suggesting avenues for future improvements in audio quality and alternative architectures.	PQMF architecture, Deep Convolutional Neural Network (CNN)	Abstract Introduction State of the art Background Data Preparation Proposed models for Voice Generation Conclusions
Diagram/Flowchart		

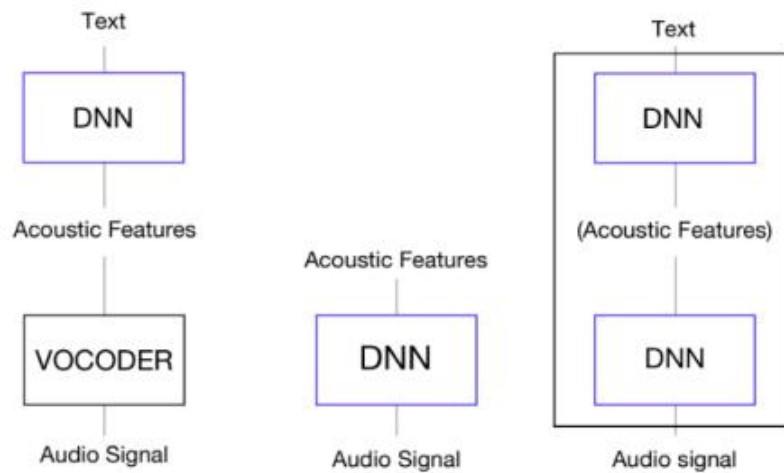


Figure 16.Voice Generation Using Deep Learning

--End of Paper 20--

2.2 COMPARISION TABLE

Author	Year	Approach	Description
Nal Kalchbrenne, Erich Elsen.	2018	Sparse WaveRNN	Reducing the computing demands for audio synthesis without sacrificing sound quality is the aim of the Sparse WaveRNN method. It attempts to address the issues of enabling audio synthesis on low-power mobile CPUs and achieving real-time or faster audio synthesis on GPUs.
Yuxuan Wang, Daisy Stanton.	2017	Tacotron: Towards End-to-End Audio Synthesis	The Tacotron approach aims to provide a generative text-to-speech (TTS) model that can convert text to audio in an end-to-end fashion. The intricacy and tediousness of conventional TTS pipelines, which usually involve several steps and demand deep domain

			knowledge, is the issue it seeks to address. With the use of rich conditioning on several attributes and a single model trained on pairs, Tacotron seeks to streamline the TTS process. This eliminates the need for feature engineering.
Aaron van den Oord, Heiga Zen.	2016	WaveNet	Using a deep neural network, WaveNet aims to produce unprocessed audio waveforms. It attempts to address the issue of producing audio that sounds natural and of excellent quality, including music and conversation. As a result of WaveNet's fully probabilistic and autoregressive architecture, every audio sample's prediction is dependent on every other sample. Because of this, WaveNet is able to produce realistic audio with natural intonation while also capturing the traits of many speakers. WaveNet can also be utilized for phoneme identification and text-to-speech jobs.
Gregory Diamos, Andrew Gibiansky.	2018	Deep Voice 3	Creating a TTS system that is both effective and of good quality is the aim of Deep Voice 3. It seeks to find a solution to the challenge of generating coherent, natural voice from textual input. By employing an attention-based sequence-to-sequence model that prevents frequent attention problems and enables a more compact architecture, Deep Voice 3 overcomes the drawbacks of earlier TTS systems. In

			order to make TTS feasible for production systems, it also focuses on enhancing inference efficiency and training speed.
T. Raman Singh, Ark Nandan Singh Chauhan, Hitesh Tewari	2022	blockchain-based end-to-end encryption (E2EE) framework	To give instant messaging applications a true end-to-end encrypted messaging service is the aim of the proposed blockchain-based end-to-end encryption (E2EE) system.
Paarth Neekhara, Shehzeen Hussain, Shlomo Dubnov.	2021	Expressive Neural Voice Cloning	It aims to address the limitation of current voice cloning methods that lack the ability to control the expressiveness of synthesized audio.
Jitong Chen, Kainan Peng.	2018	Sequence-to-sequence audio synthesis	The goal of the voice replication system is to generate a individual's voice from just a few samples. The system aims to solve the problem of voice cloning, which involves learning the voice of a speaker from a limited amount of data and generating speech that sounds like it is pronounced by the target speaker.
Sercan O. Arik, Adam Coates.	2017	Deep voice: Real-time Neural TTS	To develop a TTS system that solves the problem of generating high-quality and natural-sounding speech from text in real-time. It focuses on optimizing the different components of a TTS system.
Akshata D Vhandale, Saundarya A Karhale.	2022	Real-time Chat application	The goal of the current solution is to provide users with seamless conversations. The chat application provides a platform for

			users to connect and exchange messages in real-time, regardless of their location.
Daria Diatlova - Vitaly Shutov -	2023	EmoSpeech	It aims to solve the problem of synthesizing speech with desired emotions. The EmoSpeech model extends the FastSpeech2 architecture with modifications that enable conditioning on a given emotion while maintaining fast inference speed.
Hruthik B Gowda,Karun Datta Ramakumar,Sheethal.V,Sushma M,Dr. Madhusudhan G K	2023	Voice cloning, text-to-speech synthesis, and speaker encoding	The aim is to develop a system that can produce speech in any given speaker's voice from given text input data.
Fahima Khanam, Nadia Afrin Ritu.	2018	TTS Synthesizer for Speech Generation	The goal is to convert written text into natural and intelligible speech by using TTS.
Dr.T NAnitha, Amilio Dsouza, Ashutosh , Akshay Gole	2022	Three-level pipeline	The goal of the "Real Time Voice Cloning" is to develop a three-stage deep learning system that can clone a person's voice in real-time.
Tomáš Nekvinda, Ondřej Dušek	2020	TTS model.	The goal of the multilingual text-to-speech model discussed in the document is to enable natural-sounding speech synthesis in multiple languages using less training data.
Dr. Abhay Kasetwar, Ritik Gajbhiye.	2022	Java script,Internet,Application Registration	The goal of the chat application is to provide a reliable and flexible chat system that allows users to communicate in real-time.

2.3 WORKING OF EVALUATION TABLE:

Author Name and Year	Work Goal	System's Components	System's Mechanism	Features /Characteristics	Advantages	Limitations /Disadvantages	Platform	Results
1.Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aˆaron van den Oord, Sander Dieleman, and Koray Kavukcuoglu .	The aim is to improve the efficiency of audio synthesis without sacrificing the quality of the generated audio .	Wave Recurrent Neural Networks Sparse waveRNN Subscale dependency scheme Batched sampling	The Sparse WaveRNN	Real-time audiosynthesis High-quality synthesis Batched sampling	Reduced computation time Real-time synthesis is on mobile CPUs High-quality output	Low audio quality or synthesis performance compared to other models .	It can be run on both mobile devices and desktop computers.	the Sparse Wave RNN solution is efficient and effective for audio synthesis.
2.Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgianakis, Rob Clark, and Rif A.	to develop an end-to-end text-to-speech synthesis system using deep learning techn	Text analysis frontend Acoustic model Audio synthesis module Sequence-to-sequence framework Waveform synthesis	Tacotron	Tacotron offers simpler and more efficient approach Faster generation speed	End to end training Faster generation robustness	Large amount of data required for training Alignment learning Limited experimental	Tensor Flow,	The Tacotron model achieved a mean opinion score (MOS) of 3.82, outperforming a production param

Saurous. The paper was published in 2017.	iques .					results		etric system in terms of naturalness.
3.aaron van den oord, sander dieleman,heiga zen,karen simonyan,Orjol vinyals, alex graves, nal kalchbrenner, andrew senior,and koray kavukcuoglu.	To generate raw audio waveform using a deep neural network.	Dilated casual convolution s,autoregressive,conditioning modeling,global conditioning ,local conditioning .	Wavenet	Feature extraction from FFT spectral envelopes . Generative image modeling using spatial LSTMs.	Opeartes directly on raw audioform. Generating realistic audio, Control over generated audio.	Designing the filters for wavenet can be challenging, Computationally expensive, Time consuming.	TTS	categorical distribution over the next sample.
Wei Ping* , Kainan Peng* , Andrew Gibiansky* , Sercan " O. Arik * Ajay Kannan, Sharan Narang	To solve the problem of synthesizing natural and intelligible speech from text input .	<ul style="list-style-type: none"> ● Encoder ● Decoder ● converter 	Deep voice 3	Optimize the overall objective function , Preprocessing steps to normalize the input text for better performance.	Ability to process text quickly and efficiently, To generate high quality spectrograms To generate the final waveform parameters	May not capture long range dependencies as effectively as recurrent models . Mispronunciation.	TTS	Supporting multispeaker synthesis and delivering superior audio quality.
Raman Singh and Hitesh Tewari.	To provide a real	Mobile user, Mobile network	E2EE	This framework ensures	Users have full control	If the trusted thirdparty is	Cryptography keys	Encrypted message,

	end to end encrypted messaging service for instant messaging applications.	operator Blockchain Instant messenger Encryption/decryption Backup and restoration.		true end to end encryption.	over their encryption keys, enhancing the security and privacy	compromised it could lead to the compromise of the user's digital certificate.		Decrypted message
Author Name and Year Hruthik B Gowda, Karun Datta Ramakumar, Sheethal.V, Sushma M, Dr. Madhusudhan G K	The aim is to develop a system that can produce speech in any given speaker's voice from given text input data.	Author used TTS Synthesis, WaveNet Vocoder, Neural Network Training to create a voice cloning system that can replicate speech in any given speaker's voice from supplied text input	Speaker Encoding, TTS Synthesis, Vocoder:	The key features of this solution include its ability to produce high-quality text-to-speech (TTS) output	The positive impact of this solution is there is a better scope in advancement of technology on applications like voice assistants.	The major negative aspect of this solution is this model may give unsatisfactory results if the given audio has noise it.	Google colab	Speaker's voice generated from the supplied text input.
Author Name and Year Fahima Khanam, Farha Akhter Munmun,	The goal is to convert written	NLP-The NLP includes text analysis, phonetic conversion, and prosodic	Natural Language Processing (NLP), Digital Signal Processing	One of the features of this solution is the use of	One positive impact of this solution in	One negative impact of this solution in the	Google colab	Synthesized speech waveform that

Nadia Afrin Ritu, Alope Kumar Saha	text into natural and intelligible speech by using TTS	phrasing. And DSP(Digital signal processing)	g (DSP)	unit selection speech synthesis, which selects an optimum set of sounding units from a speech database	the project domain is that the output speech is natural and intelligible, making it easier for blind people to access information through speech. Additionally, the system can be used in various applications such as teaching aids, text reading, and talking books/toys.	project domain is that the output speech for words may have discontinuities between transitions of phonemes. This can affect the overall smoothness and naturalness of the speech output.		corresponds to the input text.
Author Name and Year Dr.T NAnitha, Amilio	The goal of the "Real Time	Three-level pipeline, Deep learning models, vocoder, reference speech, training	Feature Extraction, Acoustic Model, Vocoder	One of the features of this solution is Personal	One positive impact of solution in	One negative impact of solution in	Google colab	Cloned voice of reference

Dsouza, Ashutosh , Akshay Gole	Voic e Clon ing" is to devel op a three - stage deep learn ing syste m that can clone a perso n's voice in real-time.	ng data		ised voice interface	this project domai n is that it allows for the replica tion of unseen voices from just a few second s of referen ce speech . This means that with minim al input, the system is able to genera te highly natural soundi ng cloned voices. This can be benefi cial in variou s applica tions such as person alized voice interfa ces.	this project domain is that the cloned voice may lack natural ness and accent. This means that the cloned voice may not sound as natural or have the same accent as the origina l human voice.		audio
Author	The	The	Model	One of	This	The	Google	Natur

Name and Year Tomáš Nekvinda, Ondřej Dušek	goal of the multilingual text-to-speech (TTS) model discussed in the document is to enable natural-sounding speech synthesis in multiple languages using less training data.	multilingual text-to-speech (TTS) model discussed in the document consists of the following components : Input Text Encoder, Decoder, Convolutional Encoder, Parameter Generator Network	Architecture, Convolutional Encoders, Encoder Parameter Generation.	the features of this solution is the use of multilingual training batches to fully utilize the potential of the architecture.	solution enables better knowledge sharing , expands voice cloning capabilities, and facilitates code switching, leading to improved performance and versatility in the project domain .	major negative aspect in this solution is this model is prone to error generation of speech while working with complex languages like Chinese	colab	al sounding Multilingual Speech
Author Name and Year Dr. Abhay Kasetwar, Ritik Gajbhiye, Gopal Papewar, Rohan	The goal of the chat application is to provide a	JavaScript, Internet, Application Registration Page, Message editing field with keyboard	Define the Purpose and Goals Plan the features and functionalities Design	This is a Two way communication system which includes basic app function	The positive impact of this chat application in the project domain	Development of Chatting application is Complex and Time taking	Google colab	Display of messages, and the transmission of text messages

Nikhare, Priya Warade	reliable and flexible chat system that allows users to communicate in real-time.		the User Interface: Design the user interface (UI) of the app.	activities like Notifications and statistics for unread messages, Saving past messages etc.	n is that it focuses on developers and aims to increase their productivity. By providing a real-time messaging platform, it allows for easier and faster communication between developers	and another limitation is Internet is required in order to use the application		ges to other users
--------------------------	--	--	---	---	---	--	--	--------------------

Author Name and Year	The goal is to Create an advanced real-time voice cloning framework, enhancing accuracy and realism through deep learning	Front-end and back-end of a text-to-speech system, Text normalization, preprocessing, and tokenization, NLP module and DSP module	Text Input, Conversation	Text-to-speech functionality, User-friendly interface, Input options, Saving audio files.	It provides a useful tool for people with visual impairments, allowing them to convert text into speech and listen to it. The application has a	The quality and naturalness of the synthesized speech may vary depending on the TTS API used. Some TTS systems may	Google Collaboration	text is converted into synthesized speech that can be heard by the user.
Sakith Nalluri, A.Rohan Sai, M.Saraswati								

		in the synthesis process			simple and user- friendly interface, making it easy for users to input text and convert it into speech.	produce speech that sounds less natural or robotic		
--	--	--------------------------------	--	--	--	---	--	--

Author Name and Year Giuseppe Ruggiero, Enrico Zovato, Luigi Di Caro, Vincent Pollet	The goal of the "Advanced Gru Network" solution is to build a Text-to-Speech (TTS) system that can generate natural speech for a wide variety of speakers .	1 Conv1D layer , 3 GRU layers Linear projection layers	Advanced Gru Network" model is a sequence of mel spectrogram frames	It consists of 1 Conv1D layer and 3 GRU layers , each followed by a linear projection layer.	Compared to the GRU network, the "Advanced Gru Network" is faster during training, which can save computational time.	The main disadvantage of the Conv1D layer is that it may not capture long-range dependencies effectively, as it operates on local regions of the input.	Google Collab	fixed-dimensional embedding vector that represents the speaker characteristics in the transformed space
Author Name and Year Anita , Srinivasan	To create a text-to-speech conversion, analyze the emotions present in textual data.	Emotion Detection, Grammar Identification, Text-to-Speech Conversion, Pattern Recognition.	Emotion Takes input in the form of a sentence with an emotional word	Accurate and Reliable Data, User-Friendly Interface, Handling Complex Sentences and Emotions	Helps in understanding the emotional context of the textual data.	May not accurately detect complex emotions or subtle nuances .	Google Collab	synthesized speech that conveys the emotional content of the input sentence

Author Name and Year							
Noor Sabah Jamal M. Kadhim Ban N. Dhannoon	The goal is to solve the problem of security and privacy concerns, To provide end-to-end security for users to safely exchange private information without worrying about data leakage.	Client-side Server-side	Text messages, images, or files that they want to send to another user.	Provides a secure container to store the local storage key, making it difficult for unauthorized access.	The encryption algorithm used for password encryption may have vulnerabilities that could be exploited by attackers.	Google Collaboration	encrypts the input messages to ensure security and privacy.

2.4 DISADVANTAGES OF EXISTING SYSTEM:

Concisely summarizing the disadvantages of the above implementations:

- Lack of Accessibility:

The current system lacks accessibility features for visually impaired users and elders who have age-related vision impairments or cognitive decline, making it challenging for them to read text messages. This limitation excludes a significant portion of users who rely on auditory assistance to consume information effectively.

- Limited Multitasking Capability:

Users are unable to read messages while engaged in other activities, such as driving or exercising. This restriction forces users to stop what they are doing to read each message individually, potentially leading to distractions and safety concerns in certain situations.

- Inconvenience in noisy environments:

Users find it difficult to send audio messages in environments like buses or markets which hinders their chatting experience.

- Inconvenience in reading lengthy messages:

Users often feel reading lengthy messages is tiresome, whereas the proposed application can help read out the message while the user continues their work.

- Reduced User Engagement:

Plain text communication may not be as engaging as direct communication

CHAPTER 3

PROPOSED SYSTEM

3.1 PROPOSED SYSTEM

The invention is a mobile application that integrates cutting-edge voice cloning technology with messaging functionality. It enables users to convert text-based chats into audio messages delivered in the sender's own voice. This innovation addresses the limitations of traditional text messaging by infusing conversations with a personalized and emotionally rich dimension. By seamlessly replicating voices and providing customization options, the app enhances user engagement and emotional expression. It promotes accessibility for visually impaired individuals and aligns with the evolving landscape of digital communication. The project leverages advancements in AI, [25] voice cloning, and [29] mobile app development to create a revolutionary tool that transforms how people connect and converse in the digital age.

3.2 ADVANTAGES OF PROPOSED SYSTEM

The proposed system has the following advantages:

- Accessible to Visually impaired users.
- Enhanced multitasking.
- Increased convenience in reading lengthy messages.
- User friendliness.
- Higher user engagement.

3.3 SYSTEM REQUIREMENTS

The system requirements for the development and deployment of the project as an application are specified in this section. These requirements are not to be confused with the end-user system requirements. There are no specific, end-user requirements as the intended application is cross-platform and is supposed to work on devices of all form factors and configurations.

3.3.1 SOFTWARE REQUIREMENTS

Below are the software requirements for application development:

1. Editor for HTML, CSS and JavaScript- VS Code
2. Editor for TorToiSe : VS Code
3. Compatible with Google Chrome, Firefox, Microsoft Edge or Brave Browser.
4. Internet connection of 5 Mbps or higher.

3.3.2 HARDWARE REQUIREMENTS

Hardware requirements for application development are as follows:

1. CPU- intel i3 or higher
2. RAM – 4 GB or higher
3. Operating System – Windows 7 or higher

3.3.3 IMPLEMENTATION TECHNOLOGIES

TTS:

Text-to-speech (TTS) is a technology that converts written text into spoken words. It enables computers, devices, and [19] applications to produce natural-sounding speech output. TTS systems analyse the input text, apply linguistic rules, and use synthesized voices to [7] generate speech. These systems have numerous applications, including accessibility features for individuals with visual impairments, language learning tools, navigation systems, and automated customer service. Advances in TTS technology have led to more natural-sounding voices and improved intelligibility, making them increasingly prevalent in everyday devices and services.

TorToiSe:

TorToiSe stands out in the world of text-to-speech (TTS) for its ability to create speech with a range of voices and realistic intonation. Unlike some TTS models that offer a single voice option, TorToiSe shines in its multi-voice capabilities. It achieves this by leveraging short audio samples of a target speaker. By analyzing these samples, TorToiSe captures the unique characteristics of the voice, including pitch, tone, and speaking style. This allows the model to then generate speech that sounds remarkably similar to the reference speaker, making it a powerful tool for applications like voice acting, where different character voices are needed.

Technically, TorToiSe breaks down the text-to-speech process into multiple stages. First, it analyzes the written text and converts it into a sequence of codes representing the sounds. Next, it utilizes a diffusion model to predict how these sounds would be spoken, factoring in the flow and rhythm of natural speech (prosody). Finally, a vocoder translates this

information into actual audio that we can hear. This multi-stage approach contributes to the realistic quality of the generated speech.

Another advantage of TorToiSe is its relatively fast inference speed compared to some other TTS models. This means it can generate speech audio at a quicker pace, which can be beneficial for real-time applications where immediate audio creation is crucial.

Overall, TorToiSe offers a powerful and versatile approach to text-to-speech. Its ability to create high-quality, natural-sounding speech with diverse voices, coupled with its faster inference speed, makes it a valuable tool for various applications, including accessibility tools for reading text aloud and potentially for developers seeking to customize speech generation for specific needs. However, it's important to remember that achieving voice cloning requires reference voice samples, and the quality of the cloned voice hinges on the quality and quantity of those samples. Additionally, while faster than some models, TorToiSe still requires a decent graphics card for smooth operation.

HTML:

HTML, standing for HyperText Markup Language, is the cornerstone of web pages. Imagine it as the blueprint that instructs a web browser on how to build and display the content you see on any website.

HTML utilizes tags to define the various components of a web page, such as headings, paragraphs, images, and links. These tags act like instructions for the browser, specifying how to format and present the information. It's important to note that HTML focuses on the structure and content of a web page, not necessarily how it appears visually. Colors, fonts,

and layouts are determined by Cascading Style Sheets (CSS), which work hand-in-hand with HTML.

A typical HTML document consists of two main elements: the head and the body. The head section contains metadata about the webpage, while the body section holds the visible content that users see.

CSS:

Cascading Style Sheets, or CSS, is the secret sauce behind a website's visual appeal. While HTML lays the foundation with content and structure, CSS determines how that content is displayed. It allows you to define fonts, colors, layouts, and more to create a cohesive and user-friendly experience. Imagine it as the fashion designer for your website, taking a basic structure and transforming it into something stylish and functional.

Furthermore, CSS offers a separation of concerns by isolating presentation aspects from content. This means you can make global style changes by editing a single CSS file, rather than having to modify every HTML page individually. This not only saves time but also ensures consistency across your entire website. In short, CSS is an essential tool for web developers to bring their creative visions to life.

Vanilla JS:

Vanilla JavaScript, or vanilla JS for short, is essentially JavaScript in its raw form. It avoids the use of external libraries or frameworks, meaning you code everything from scratch. This lean approach boasts two key benefits: speed and flexibility. Websites built with vanilla JS tend to load faster due to the lack of additional libraries. Additionally, you have complete control over the code, allowing for deep customization to fit your specific needs.

CHAPTER 4

SYSTEM DESIGN

4.1 PROPOSED SYSTEM ARCHITECTURE

The Proposed system is a chat application that integrates cutting-edge voice cloning technology with messaging functionality using deep learning. It enables users to convert text-based chats into audio messages. This innovation addresses the limitations of traditional text messaging by infusing conversations with audio messages. It promotes accessibility for visually impaired individuals and aligns with the evolving landscape of digital communication.

4.2 UML DIAGRAMS

The application on an overall involves four UML Representations including class, use-case, activity and sequence representations.

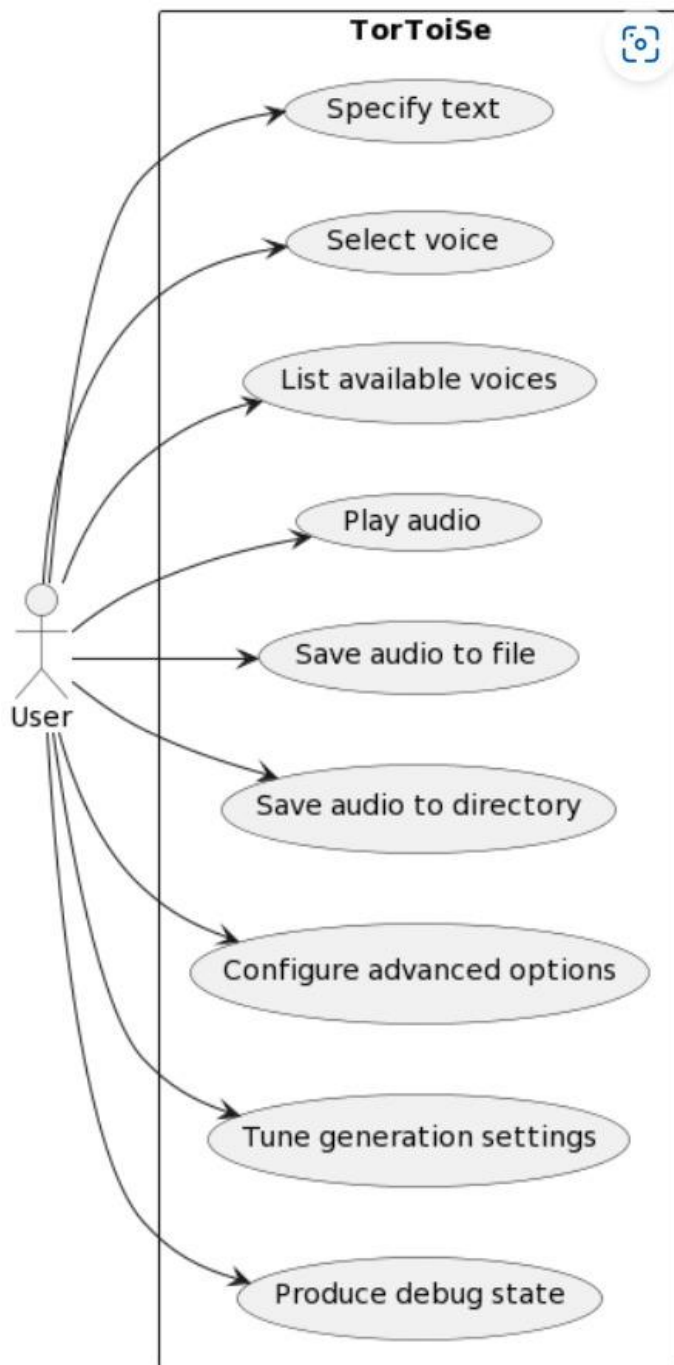


Figure 17. process of Tortoise

4.2.1 Use case diagram:

This use case diagram represents the functionality of the "TorToiSe" text-to-speech program.

Here's a description of each element:

1. User (U): Represents the user interacting with the TorToiSe application.
2. Specify text: This use case involves the user specifying the text that they want to convert to speech. It's a fundamental step in using the application.
3. Select voice: In this use case, the user selects the voice that will be used for synthesizing the speech. The application likely provides a list of available voices for the user to choose from.
4. List available voices: This use case allows the user to view a list of all available voices supported by the application. It provides information to the user about the voices they can use for speech synthesis.
5. Play audio: Once the text has been synthesized into speech, the user can choose to play the audio output directly within the application.
6. Save audio to file: This use case enables the user to save the synthesized audio to a file on their local system. It provides an option for users who want to keep a copy of the speech output for later use.
7. Save audio to directory: Similar to saving audio to a file, this use case allows the user to save the audio output to a specific directory on their system. It may be useful for users who need to organize multiple audio files.

8. Configure advanced options: Some users may want to configure advanced settings for speech synthesis, such as setting the quality preset or adjusting specific parameters. This use case provides functionality for such customization.

9. Tune generation settings: This use case involves fine-tuning the settings related to speech generation, such as adjusting the number of samples or temperature. It provides users with more control over the synthesis process.

10. Produce debug state: For debugging purposes or advanced users, this use case allows the application to produce debug states, which can aid in troubleshooting or reproducing issues.

Overall, this use case diagram illustrates the various actions and functionalities available to the user within the TorToiSe text-to-speech application.

4.2.2 Sequence diagram:

This sequence diagram illustrates the interaction between the user, the TorToiSe application (TTS), the TextToSpeech module (TTSP), and the pydub library during the process of text-to-speech synthesis. Here's a description of each step:

1. User: The user initiates the TorToiSe application by running it with specific arguments.
2. TorToiSe (TTS): Upon activation, the TorToiSe application initializes and begins processing the user's input.

3. TextToSpeech (TTSP): TTS interacts with the TextToSpeech module to initialize an instance, which involves loading voices and settings necessary for speech synthesis.
4. Parse command line arguments: TTS parses the command line arguments provided by the user to determine the desired functionality, such as specifying the text to synthesize and selecting voices.
5. Split and recombine text: If necessary, TTS splits the input text into smaller chunks and recombines them to optimize the synthesis process.
6. Determine output directory: TTS determines the output directory where the synthesized audio will be saved, if applicable.
7. Set up generation settings: TTS configures the generation settings based on the user's preferences and any default settings.
8. Loop for each voice and text combination: TTS iterates through each combination of voice and text to generate the corresponding speech audio.
9. Load voice samples and conditioning latents: For each combination, TTS loads the voice samples and conditioning latents required for synthesis.
10. Generate speech audio: TTSP generates the speech audio based on the loaded voice samples and conditioning latents.

11. Combine generated audio parts: TTS combines the generated audio parts if multiple chunks are synthesized for the same voice and text combination.

12. Save or play audio: TTS saves the synthesized audio to the specified output directory or plays it directly, depending on the user's preferences.

13. Produce debug states if enabled: If enabled, TTS produces debug states, which can aid in troubleshooting or analysis.

14. Output audio or debug states: Finally, TTS provides the output to the user, which includes the synthesized audio or debug states, depending on the application's configuration.

This sequence diagram provides a comprehensive overview of the workflow involved in text-to-speech synthesis using the TorToiSe application, highlighting the interactions between different components during the process.

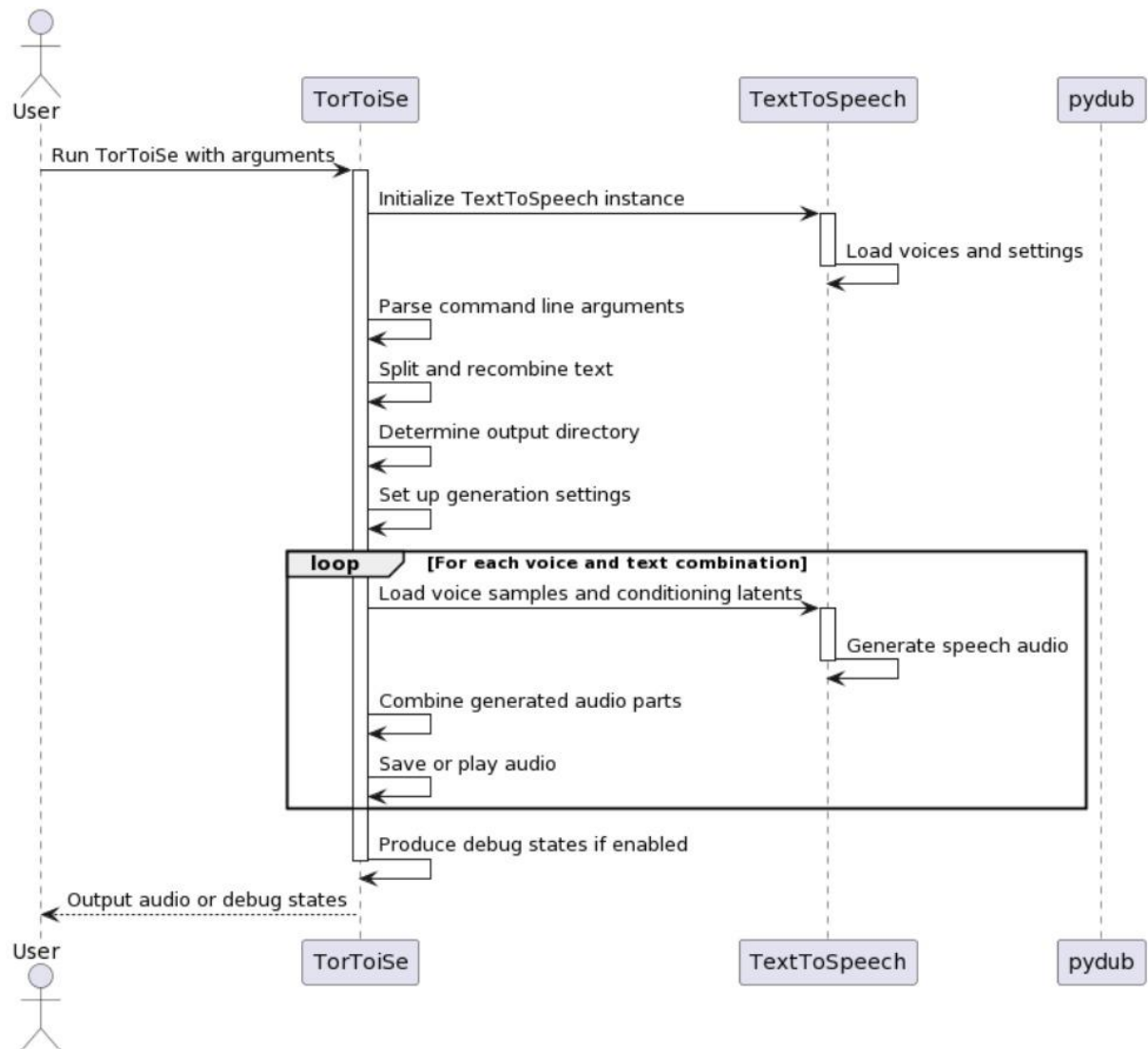


Figure 18. process of text to speech synthesis.

4.2.3 Activity diagram:

The activity diagram outlines the process flow of the TorTolSe text-to-speech program, depicting various actions and decisions taken during its execution.

1. User Interaction:

- The process starts with user interaction, indicating the initiation of the TorTolSe program.

2. List Voices:

- If the user chooses to list available voices, the program proceeds to list the voices and stops afterward.

3. Parse Command Line Arguments:

- If the user doesn't opt to list voices, the program parses the command line arguments provided.

4. Text Processing:

- Depending on whether text is provided as an argument or read from stdin, the program either splits and recombines the text or reads it from the standard input.

5. Output Handling:

- The program checks if an output directory is specified.
- If yes, it creates the output directory if it doesn't exist.
- If no, it checks if multiple voices or multiple candidates are specified. If so, the process stops, as these options require an output directory.

6. Rendering Text:

- After text processing and output handling, the program proceeds to render the text.
- If the play option is selected, the audio is played.
- If the output directory is specified, the audio clips are saved to the directory.
- If neither option is selected, the audio is saved to a file.

7. Debug State Production:

- Finally, if the option to produce debug states is enabled, the program generates debug states before stopping.

This activity diagram provides a clear overview of the sequential steps and decision points involved in the TorToiSe program's execution, from user interaction to text rendering and optional debug state production.

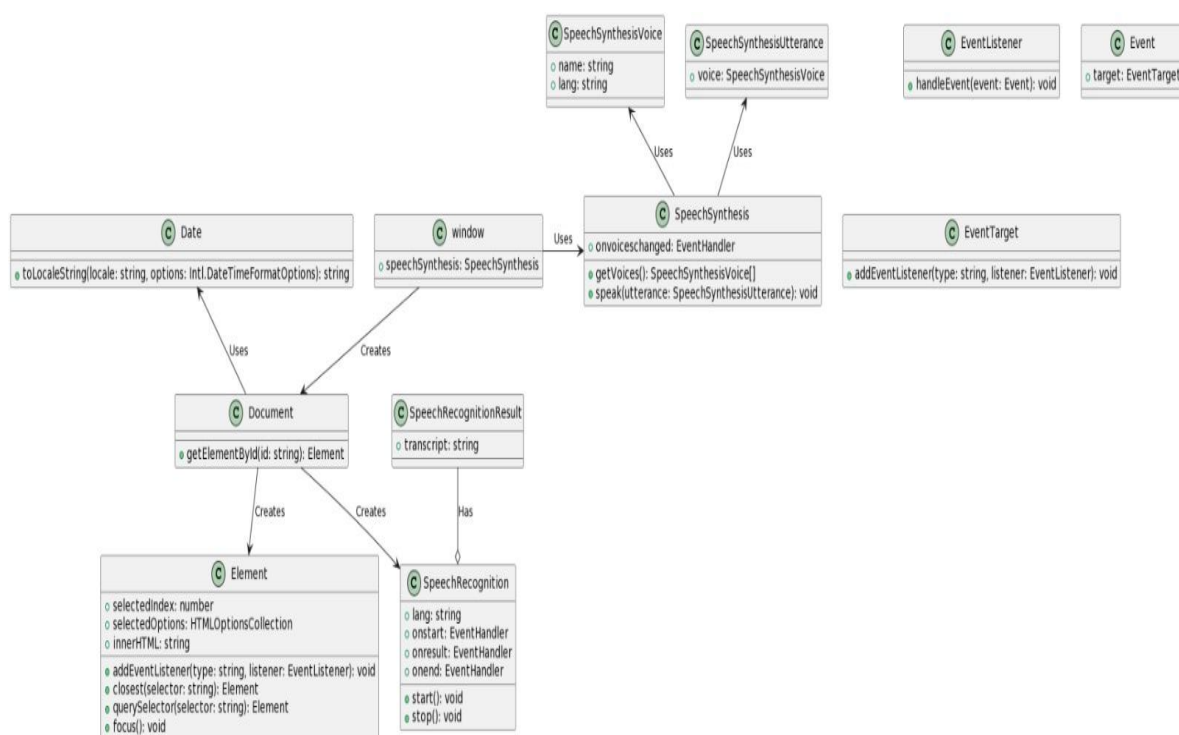


Figure 19. function of application.

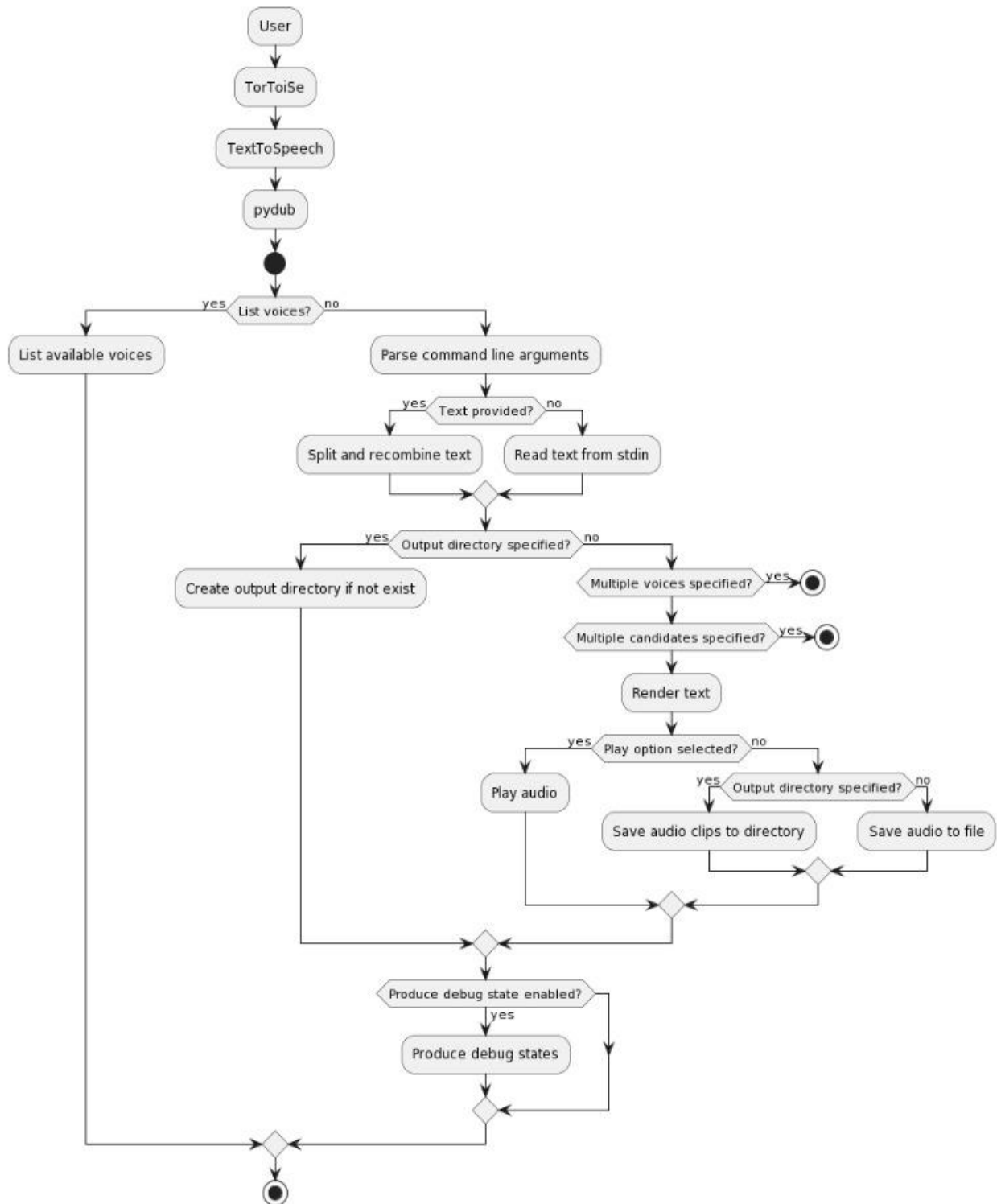


Figure 20. Detailed description of conversion of text to speech.

4.2.3 Use case diagram:

The use case diagram illustrates the relationships and interactions between various classes and objects in a JavaScript environment related to speech synthesis and recognition.

1. window Class:

- Represents the global window object in a web browser environment.
- Contains a reference to the `'speechSynthesis'` object for speech synthesis functionality.

2. SpeechSynthesis Class:

- Represents the speech synthesis interface provided by the browser.
- Provides methods such as `'getVoices()'` to retrieve available voices and `'speak()'` to initiate speech synthesis.

3. SpeechSynthesisUtterance Class:

- Represents an utterance that can be spoken using speech synthesis.
- Contains properties such as `'voice'` to specify the voice to be used for synthesis.

4. SpeechSynthesisVoice Class:

- Represents a voice available for speech synthesis.
- Contains properties like `'name'` and `'lang'` to specify the name and language of the voice.

5. Document Class:

- Represents the DOM document object.
- Provides methods like `'getElementById()'` to retrieve DOM elements.

6. Element Class:

- Represents a generic DOM element.
- Provides various methods for DOM manipulation, event handling, and element selection.

7. SpeechRecognition Class:

- Represents the speech recognition interface provided by the browser.
- Allows speech recognition functionality with methods like ``start()`` and ``stop()``.

8. Date Class:

- Represents the Date object in JavaScript.
- Provides methods like ``toLocaleString()`` to format date and time according to the specified locale and options.

9. SpeechRecognitionResult Class:

- Represents the result of speech recognition.
- Contains the ``transcript`` property, which holds the recognized speech text.

The diagram shows how these classes are related and how they interact within the JavaScript environment. It demonstrates how the global window object creates instances of the Document and SpeechRecognition classes, which in turn utilize the SpeechSynthesis and Date classes for speech synthesis and recognition functionalities, respectively.

CHAPTER 5

IMPLEMENTATION

5.1 IMPLEMENTATION WITH HYPOTHETICAL SCENARIOS

When a user receives a text message within the app and selects the chat, they can choose to utilize the text-to-speech feature.

1. Recording User's Voice:

This step is the base for [16] voice cloning process. Users can register their voice to the application via recording or uploading an audio file.

2. Analysing voice patterns:

The process begins with the [23] app analysing existing voice recordings of the sender to capture their distinct vocal patterns, intonations, and nuances.

3. Voice cloning:

These acoustic features are then fed into sophisticated AI algorithms, which generate a synthetic voice that closely emulates the sender's authentic voice.

4. Integration of synthesized audio with text:

This synthesized voice is seamlessly applied to the text message, resulting in an audio message that retains the [5] emotional depth and personal touch of the original speaker. This subsection evaluates the use of application in various scenarios involving different hypothetical situations.

5.1.1 Working of TorToiSe

The working of a text-to-speech (TTS) system implemented through a Python script with a command-line interface. A brief explanation of its operation is mentioned below:

1. Argument Parsing:

By using a predefined parser, the script first parses the command-line parameters.

The input text, voice selection, output choices, and other settings for [8] TTS synthesis are all specified by these arguments.

2. Voice Management:

It loads the voices that are available and selects a voice by using the arguments that are supplied. When [14] more than one voice is chosen, it gets ready to use each voice to synthesize speech.

3. TTS Configuration:

The script configures the parameters needed for TTS synthesis, including preset quality, random seed, and synthesis process adjustment options.

4. Synthesis Loop:

For each selected voice and each text chunk, the script generates audio based on the text and voice using the specified settings. It synthesises speech for all combinations of voices and text chunks.

5. Output Handling:

Depending on the output options specified, the script either saves the synthesized audio to files, plays it back, or both. If multiple voices or text chunks are involved, it organises the output accordingly.

6. Exit:

Once the synthesis process is complete, the script exits, concluding its operation.

5.2 SOURCE CODE

index.html

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Chat App</title>

    <link rel="stylesheet" href="style.css" />

  </head>

  <body>

    <!-- Person selector: this contains buttons for user to select whether to chat as yogi
    or vicky -->

    <div class="person-selector">

      <button class="button person-selector-button active-person" id="yogi-
selector">yogi</button>

      <button class="button person-selector-button" id="vicky-selector">vicky</button>

    </div>

    <div class="chat-container">

      <h2 class="chat-header">yogi chatting...</h2>

      <div class="chat-messages">

        <div class="message blue-bg">

          <div class="message-sender">yogi</div>

          <div class="message-text">Hey vicky, what's up?</div>

          <div class="message-timestamp"><p id="datetime"></p></div>

        </div>

        <div class="message gray-bg">

          <div class="message-sender">vicky</div>

          <div class="message-text">Not much, just living the dream. How about you?</div>

          <div class="message-timestamp"><p id="datetime2"></p></div>

        </div>

      </div>

    </div>

  </body>

</html>
```



```

    </div>

</div>

<div class="box">

    <select id="voiceList">

        <option>Option 1</option>

        <option>Option 2</option>

        <option>Option 3</option>

        <option>Option 4</option>

        <option>Option 5</option>

    </select>

</div>


<style>

    /* Adjust the size of the voice message button */

    .voice-message-button {

        width: 40px;

        height: 40px;

    }

    /* Adjust the size of the voice icon */

    .voice-message-button img {

        width: 25px; /* Adjust the width as needed */

        height: 25px; /* Adjust the height as needed */

    }

</style>


<form class="chat-input-form">

    <input id="input" type="text" class="chat-input" required placeholder="Type here,
yogi..." />

    <button id="buttonOne" type="submit" class="button send-button">Send</button>

    <!-- Voice message button with voice icon -->

    <button id="voiceMessageButton" class="button voice-message-button">

```

```

    </button>

</form>

    <button class="button clear-chat-button">Clear Chat</button>

</div>

<script>

    const timestamp = new Date().toLocaleString('en-US', { hour: 'numeric', minute:
'numeric', hour12: true })

    document.getElementById("datetime").innerHTML = timestamp;

    document.getElementById("datetime2").innerHTML = timestamp;

</script>

<script src="app.js"></script>

</body>

</html>

```

style.css

```

* {
    margin: 0;
    padding: 0;
}

body {

    background-image: linear-gradient(

        23deg,

        hsl(49deg 100% 69%) 0%,

        hsl(16deg 80% 61%) 2%,

        hsl(330deg 81% 34%) 12%,

        hsl(259deg 100% 15%) 50%,

        hsl(212deg 100% 25%) 88%,

        hsl(197deg 100% 30%) 98%,

        hsl(183deg 79% 36%) 100%

    );

    height: 100vh;

```

```
}
```

```
.button {  
  border: none;  
  padding: 0.625em;  
  border-radius: 0.5em;  
  cursor: pointer;  
}
```

```
.button:hover {  
  filter: brightness(0.9);  
}
```

```
.button:active {  
  transform: translateY(2px);  
}
```

```
.person-selector {  
  display: flex;  
  justify-content: center;  
  gap: 1em;  
  margin: 3em auto 1em;  
  max-width: 40em;  
}
```

```
.person-selector-button {  
  width: 100%;  
  background-color: #15202b;  
  color: #fff;  
  font-size: 1.1em;  
}
```

```
.active-person {  
  background: #08529d;  
  box-shadow: 0 0 0.5em 0.1em #c3c3c333;  
}  
  
.chat-container {  
  background: #15202b;  
  font-family: 'Roboto', sans-serif;  
  border-radius: 0.5em;  
  padding: 0.5em 1.25em;  
  margin: auto;  
  max-width: 37.5em;  
  height: 37.5em;  
  box-shadow: 0 0 1.25em 0.5em #c3c3c333;  
}  
  
.chat-header {  
  margin-bottom: 1em;  
  color: #fff;  
}  
  
.chat-header h2 {  
  font-size: 1.25em;  
  font-weight: bold;  
}  
  
.chat-messages {  
  height: 23em;  
  overflow-y: scroll;  
}
```

```
.chat-messages::-webkit-scrollbar {  
    display: none;  
}  
  
#voiceList {  
    display: none; /* Hide the voice options dropdown by default */  
}  
  
.message {  
    padding: 0.625em;  
    border-radius: 1em;  
    margin-bottom: 0.625em;  
    display: flex;  
    flex-direction: column;  
    color: #fff;  
}  
  
.message-sender {  
    font-weight: bold;  
    margin-bottom: 0.31em;  
}  
  
.message-text {  
    font-size: 1em;  
    margin-bottom: 0.31em;  
    word-wrap: break-word;  
}  
  
.message-timestamp {  
    font-size: 0.75em;
```

```
    text-align: right;
}

.blue-bg {
    background-color: #1c9bef;
}

.gray-bg {
    background-color: #3d5365;
}

.chat-input-form {
    display: flex;
    align-items: center;
    margin-top: 2em;
    gap: 0.625em;
}

.chat-input {
    padding: 0.625em;
    border: none;
    border-radius: 0.5em;
    background-color: #f5f5f5;
    color: #333;
    font-size: 1em;
    flex-grow: 1;
}

.send-button {
    background-color: #1c9bef;
    color: #fff;
```

```

    font-size: 1em;

    font-weight: bold;
}

.clear-chat-button {

    display: block;

    margin: 2.5em auto;
}

```

app.js

```

function hideLoading() {
    document.getElementById("loading").style.display = "none";
}
var txtInput = document.getElementById("input");
var voiceList = document.getElementById("voiceList");
var btnSpeak = document.getElementById("buttonOne");
var synth = window.speechSynthesis;
var voices = [];

PopulateVoices();

if (speechSynthesis !== undefined) {
    speechSynthesis.onvoiceschanged = PopulateVoices;
}

function PopulateVoices() {
    voices = synth.getVoices();
    var selectedIndex = voiceList.selectedIndex < 0 ? 0 : voiceList.selectedIndex;
    voiceList.innerHTML = "";
    let selectedCounter = 0;

    voices.forEach((voice) => {
        var listItem = document.createElement("option");
        listItem.textContent = voice.name;
        listItem.setAttribute("data-lang", voice.lang);
        listItem.setAttribute("data-name", voice.name);
        voiceList.appendChild(listItem);
        // Set the first option as default
        if (selectedCounter === 45) {
            setTimeout(() => {
                listItem.selected = true;
                voiceList.selectedIndex = index;
            }, 0);
        }
        selectedCounter++;
    });

    voiceList.selectedIndex = selectedIndex;
    // Hide the dropdown
    voiceList.style.display = "none";
}

```

```

// @ts-nocheck

const yogiSelectorBtn = document.querySelector('#yogi-selector')
const vickySelectorBtn = document.querySelector('#vicky-selector')
const chatHeader = document.querySelector('.chat-header')
const chatMessages = document.querySelector('.chat-messages')
const chatInputForm = document.querySelector('.chat-input-form')
const chatInput = document.querySelector('.chat-input')
const clearChatBtn = document.querySelector('.clear-chat-button')

const messages = JSON.parse(localStorage.getItem('messages')) || []

const readMessage = (event) => {
  const clickedMessage = event.target.closest('.message');

  if (clickedMessage) {
    const messageText = clickedMessage.querySelector('.message-text').textContent;

    if (messageText) {
      const toSpeak = new SpeechSynthesisUtterance(messageText);
      var selectedVoiceName = voiceList.selectedOptions[0].getAttribute(
        "data-name"
      );
      voices.forEach((voice) => {
        if (voice.name === selectedVoiceName) {
          toSpeak.voice = voice;
        }
      });
      synth.speak(toSpeak);
    }
  }
};

chatMessages.addEventListener('click', readMessage);

const createChatMessageElement = (message) => `
  <div class="message ${message.sender === 'yogi' ? 'blue-bg' : 'gray-bg'}">
    <div class="message-sender">${message.sender}</div>
    <div class="message-text">${message.text}</div>
    <div class="message-timestamp">${message.timestamp}</div>
  </div>
`

window.onload = () => {
  messages.forEach((message) => {
    chatMessages.innerHTML += createChatMessageElement(message)
  })
}

let messageSender = 'yogi'

const updateMessageSender = (name) => {
  messageSender = name
  chatHeader.innerText = `${messageSender} chatting...`
  chatInput.placeholder = `Type here, ${messageSender}...`

  if (name === 'yogi') {
    yogiSelectorBtn.classList.add('active-person')
    vickySelectorBtn.classList.remove('active-person')
  }
}

```



```

    }
    if (name === 'vicky') {
      vickySelectorBtn.classList.add('active-person')
      yogiSelectorBtn.classList.remove('active-person')
    }

    /* auto-focus the input field */
    chatInput.focus()
  }

  yogiSelectorBtn.onclick = () => updateMessageSender('yogi')
  vickySelectorBtn.onclick = () => updateMessageSender('vicky')

  const sendMessage = (e) => {
    e.preventDefault()

    const timestamp = new Date().toLocaleString('en-US', { hour: 'numeric', minute: 'numeric',
    hour12: true })
    const message = {
      sender: messageSender,
      text: chatInput.value,
      timestamp,
    }

    /* Save message to local storage */
    messages.push(message)
    localStorage.setItem('messages', JSON.stringify(messages))

    /* Add message to DOM */
    chatMessages.innerHTML += createChatMessageElement(message)

    /* Clear input field */
    chatInputForm.reset()

    /* Scroll to bottom of chat messages */
    chatMessages.scrollTop = chatMessages.scrollHeight
  }

  chatInputForm.addEventListener('submit', sendMessage)

  clearChatBtn.addEventListener('click', () => {
    localStorage.clear()
    chatMessages.innerHTML = ''
  })

  const startButton = document.getElementById('voiceMessageButton');

  const recognition = new (window.SpeechRecognition || window.webkitSpeechRecognition ||
  window.mozSpeechRecognition || window.msSpeechRecognition)();
  recognition.lang = 'en-US';

  recognition.onstart = () => {
    txtInput.placeholder = 'Listening...'; // Update the placeholder to indicate listening
  };

  const timeoutId = setTimeout(() => {
    recognition.stop();
  }, 5000);

```

```

recognition.onresult = (event) => {
  const transcript = event.results[0][0].transcript;
  txtInput.value = transcript; // Display the transcript in the text input box
};
// Clear the timeout if recognition is successful
clearTimeout(timeoutId);

recognition.onend = () => {
  txtInput.placeholder = 'Type here, yogi...'; // Reset the placeholder after recognition
ends
};

startButton.addEventListener('click', () => {
  recognition.start(); // Start the recognition when the button is clicked
});

```

tts.py

```

<!DOCTYPE html>

#!/usr/bin/env python3

import argparse

import os

import sys

import tempfile

import time

import torch

import torchaudio

from tortoise.api import MODELS_DIR, TextToSpeech

from tortoise.utils.audio import get_voices, load_voices, load_audio

from tortoise.utils.text import split_and_recombine_text

parser = argparse.ArgumentParser(

    description='TorToiSe is a text-to-speech program that is capable of synthesizing
speech '

        'in multiple voices with realistic prosody and intonation.')

parser.add_argument(

```

```

    'text', type=str, nargs='*',

    help='Text to speak. If omitted, text is read from stdin.')

parser.add_argument(

    '-v, --voice', type=str, default='random', metavar='VOICE', dest='voice',

    help='Selects the voice to use for generation. Use the & character to join two voices together. '

        'Use a comma to perform inference on multiple voices. Set to "all" to use all available voices. '

        'Note that multiple voices require the --output-dir option to be set.')

parser.add_argument(

    '-V, --voices-dir', metavar='VOICES_DIR', type=str, dest='voices_dir',

    help='Path to directory containing extra voices to be loaded. Use a comma to specify multiple directories.')

parser.add_argument(

    '-p, --preset', type=str, default='fast', choices=['ultra_fast', 'fast', 'standard', 'high_quality'], dest='preset',

    help='Which voice quality preset to use.')

parser.add_argument(

    '-q, --quiet', default=False, action='store_true', dest='quiet',

    help='Suppress all output.')

output_group = parser.add_mutually_exclusive_group(required=True)

output_group.add_argument(

    '-l, --list-voices', default=False, action='store_true', dest='list_voices',

    help='List available voices and exit.')

output_group.add_argument(

    '-P, --play', action='store_true', dest='play',

    help='Play the audio (requires pydub).')

output_group.add_argument(

    '-o, --output', type=str, metavar='OUTPUT', dest='output',

    help='Save the audio to a file.')

output_group.add_argument(

    '-O, --output-dir', type=str, metavar='OUTPUT_DIR', dest='output_dir',

```

```

    help='Save the audio to a directory as individual segments.')

multi_output_group = parser.add_argument_group('multi-output options (requires --output-dir)')

multi_output_group.add_argument(
    '--candidates', type=int, default=1,
    help='How many output candidates to produce per-voice. Note that only the first candidate is used in the combined output.')

multi_output_group.add_argument(
    '--regenerate', type=str, default=None,
    help='Comma-separated list of clip numbers to re-generate.')

multi_output_group.add_argument(
    '--skip-existing', action='store_true',
    help='Set to skip re-generating existing clips.')

advanced_group = parser.add_argument_group('advanced options')

advanced_group.add_argument(
    '--produce-debug-state', default=False, action='store_true',
    help='Whether or not to produce debug_states in current directory, which can aid in reproducing problems.')

advanced_group.add_argument(
    '--seed', type=int, default=None,
    help='Random seed which can be used to reproduce results.')

advanced_group.add_argument(
    '--models-dir', type=str, default=MODELS_DIR,
    help='Where to find pretrained model checkpoints. Tortoise automatically downloads these to '

    '~/.cache/tortoise/.models, so this should only be specified if you have custom checkpoints.')

advanced_group.add_argument(
    '--text-split', type=str, default=None,
    help='How big chunks to split the text into, in the format <desired_length>,<max_length>.')

advanced_group.add_argument(

```

```

        '--disable-redaction', default=False, action='store_true',

        help='Normally text enclosed in brackets are automatically redacted from the spoken
output '

        '(but are still rendered by the model), this can be used for prompt engineering. '

        'Set this to disable this behavior.')
advanced_group.add_argument(

    '--device', type=str, default=None,

    help='Device to use for inference.')
advanced_group.add_argument(

    '--batch-size', type=int, default=None,

    help='Batch size to use for inference. If omitted, the batch size is set based on
available GPU memory.')

tuning_group = parser.add_argument_group('tuning options (overrides preset settings)')
tuning_group.add_argument(

    '--num-autoregressive-samples', type=int, default=None,

    help='Number of samples taken from the autoregressive model, all of which are filtered
using CLVP. '

    'As TorToiSe is a probabilistic model, more samples means a higher probability of
creating something "great".')
tuning_group.add_argument(

    '--temperature', type=float, default=None,

    help='The softmax temperature of the autoregressive model.')
tuning_group.add_argument(

    '--length-penalty', type=float, default=None,

    help='A length penalty applied to the autoregressive decoder. Higher settings causes
the model to produce more terse outputs.')
tuning_group.add_argument(

    '--repetition-penalty', type=float, default=None,

    help='A penalty that prevents the autoregressive decoder from repeating itself during
decoding. '

    'Can be used to reduce the incidence of long silences or "uhhhhhhs", etc.')
tuning_group.add_argument(

    '--top-p', type=float, default=None,

```

```

    help='P value used in nucleus sampling. 0 to 1. Lower values mean the decoder produces
more "likely" (aka boring) outputs.')
```

tuning_group.add_argument(

```

    '--max-mel-tokens', type=int, default=None,

    help='Restricts the output length. 1 to 600. Each unit is 1/20 of a second.')
```

tuning_group.add_argument(

```

    '--cvvp-amount', type=float, default=None,

    help='How much the CVVP model should influence the output.'

    'Increasing this can in some cases reduce the likelihood of multiple speakers.')
```

tuning_group.add_argument(

```

    '--diffusion-iterations', type=int, default=None,

    help='Number of diffusion steps to perform. More steps means the network has more
chances to iteratively'

    'refine the output, which should theoretically mean a higher quality output. '

    'Generally a value above 250 is not noticeably better, however.')
```

tuning_group.add_argument(

```

    '--cond-free', type=bool, default=None,

    help='Whether or not to perform conditioning-free diffusion. Conditioning-free
diffusion performs two forward passes for '

    'each diffusion step: one with the outputs of the autoregressive model and one
with no conditioning priors. The output '

    'of the two is blended according to the cond_free_k value below. Conditioning-free
diffusion is the real deal, and '

    'dramatically improves realism.')
```

tuning_group.add_argument(

```

    '--cond-free-k', type=float, default=None,

    help='Knob that determines how to balance the conditioning free signal with the
conditioning-present signal. [0,inf]. '

    'As cond_free_k increases, the output becomes dominated by the conditioning-free
signal. '

    'Formula is: output=cond_present_output*(cond_free_k+1)-
cond_absenct_output*cond_free_k')
```

tuning_group.add_argument(

```

    '--diffusion-temperature', type=float, default=None,

    help='Controls the variance of the noise fed into the diffusion model. [0,1]. Values at
0 '
```

```
        'are the "mean" prediction of the diffusion network and will sound bland and smeared. ')
```

```
usage_examples = f'''
```

Examples:

Read text using random voice and place it in a file:

```
{parser.prog} -o hello.wav "Hello, how are you?"
```

Read text from stdin and play it using the tom voice:

```
echo "Say it like you mean it!" | {parser.prog} -P -v tom
```

Read a text file using multiple voices and save the audio clips to a directory:

```
{parser.prog} -O /tmp/tts-results -v tom,emma <textfile.txt
```

```
'''
```

try:

```
    args = parser.parse_args()
```

except SystemExit as e:

```
    if e.code == 0:
```

```
        print(usage_examples)
```

```
    sys.exit(e.code)
```

```
extra_voice_dirs = args.voices_dir.split(',') if args.voices_dir else []
```

```
all_voices = sorted(get_voices(extra_voice_dirs))
```

```
if args.list_voices:
```

```
    for v in all_voices:
```

```

        print(v)

    sys.exit(0)

selected_voices = all_voices if args.voice == 'all' else args.voice.split(',')
selected_voices = [v.split('&') if '&' in v else [v] for v in selected_voices]
for voices in selected_voices:
    for v in voices:
        if v != 'random' and v not in all_voices:
            parser.error(f'voice {v} not available, use --list-voices to see available
voices.')

if len(args.text) == 0:
    text = ''

    for line in sys.stdin:
        text += line
else:
    text = ' '.join(args.text)

text = text.strip()

if args.text_split:
    desired_length, max_length = [int(x) for x in args.text_split.split(',')]

    if desired_length > max_length:
        parser.error(f'--text-split: desired_length ({desired_length}) must be <=
max_length ({max_length})')

    texts = split_and_recombine_text(text, desired_length, max_length)
else:
    texts = split_and_recombine_text(text)

if len(texts) == 0:
    parser.error('no text provided')

if args.output_dir:
    os.makedirs(args.output_dir, exist_ok=True)
else:

```



```

if len(selected_voices) > 1:
    parser.error('cannot have multiple voices without --output-dir')

if args.candidates > 1:
    parser.error('cannot have multiple candidates without --output-dir')

# error out early if pydub isn't installed
if args.play:
    try:
        import pydub
        import pydub.playback

    except ImportError:
        parser.error('--play requires pydub to be installed, which can be done with "pip install pydub"')

seed = int(time.time()) if args.seed is None else args.seed

if not args.quiet:
    print('Loading tts...')

tts = TextToSpeech(models_dir=args.models_dir, enable_redaction=not args.disable_redaction,
                   device=args.device, autoregressive_batch_size=args.batch_size)

gen_settings = {
    'use_deterministic_seed': seed,
    'verbose': not args.quiet,
    'k': args.candidates,
    'preset': args.preset,
}

tuning_options = [
    'num_autoregressive_samples', 'temperature', 'length_penalty', 'repetition_penalty',
    'top_p',
    'max_mel_tokens', 'cvvp_amount', 'diffusion_iterations', 'cond_free', 'cond_free_k',
    'diffusion_temperature']

for option in tuning_options:
    if getattr(args, option) is not None:

```

```

        gen_settings[option] = getattr(args, option)

total_clips = len(texts) * len(selected_voices)

regenerate_clips = [int(x) for x in args.regenerate.split(',')] if args.regenerate else
None

for voice_idx, voice in enumerate(selected_voices):

    audio_parts = []

    voice_samples, conditioning_latents = load_voices(voice, extra_voice_dirs)

    for text_idx, text in enumerate(texts):

        clip_name = f'{"-".join(voice)}_{text_idx:02d}'

        if args.output_dir:

            first_clip = os.path.join(args.output_dir, f'{clip_name}_00.wav')

            if (args.skip_existing or (regenerate_clips and text_idx not in
regenerate_clips)) and os.path.exists(first_clip):

                audio_parts.append(load_audio(first_clip, 24000))

                if not args.quiet:

                    print(f'Skipping {clip_name}')

                continue

            if not args.quiet:

                print(f'Rendering {clip_name} ({(voice_idx * len(texts) + text_idx + 1)} of
{total_clips})...')

                print(' ' + text)

            gen = tts.tts_with_preset(

                text, voice_samples=voice_samples, conditioning_latents=conditioning_latents,
**gen_settings)

            gen = gen if args.candidates > 1 else [gen]

            for candidate_idx, audio in enumerate(gen):

                audio = audio.squeeze(0).cpu()

                if candidate_idx == 0:

                    audio_parts.append(audio)

            if args.output_dir:

                filename = f'{clip_name}_{candidate_idx:02d}.wav'

                torchaudio.save(os.path.join(args.output_dir, filename), audio, 24000)

```

```

audio = torch.cat(audio_parts, dim=-1)

if args.output_dir:
    filename = f'{"-".join(voice)}_combined.wav'
    torchaudio.save(os.path.join(args.output_dir, filename), audio, 24000)
elif args.output:
    filename = args.output if args.output else os.tmp
    torchaudio.save(args.output, audio, 24000)
elif args.play:
    f = tempfile.NamedTemporaryFile(suffix='.wav', delete=True)
    torchaudio.save(f.name, audio, 24000)
    pydub.playback.play(pydub.AudioSegment.from_wav(f.name))

if args.produce_debug_state:
    os.makedirs('debug_states', exist_ok=True)
    dbg_state = (seed, texts, voice_samples, conditioning_latents, args)
    torch.save(dbg_state, os.path.join('debug_states', f'debug_{"-".join(voice)}.pth'))

```

CHAPTER 6

RESULTS

Our presents the live implementation of the proposed solution, VoiceChat. In the above figure, we can see two users, yogi and vicky chatting with each other in the interface. The chats act as buttons and when clicked provide accurate text-to-speech synthesis. These chats are integrated with live time features to resemble traditional chat applications.

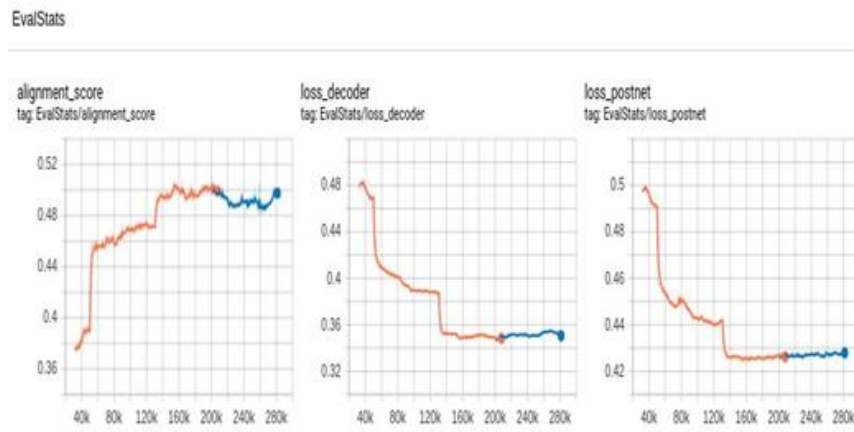


Figure 21. Performance metrics of Text to Speech Algorithm

The obtained results after the development of the VoiceChat application are as follows.

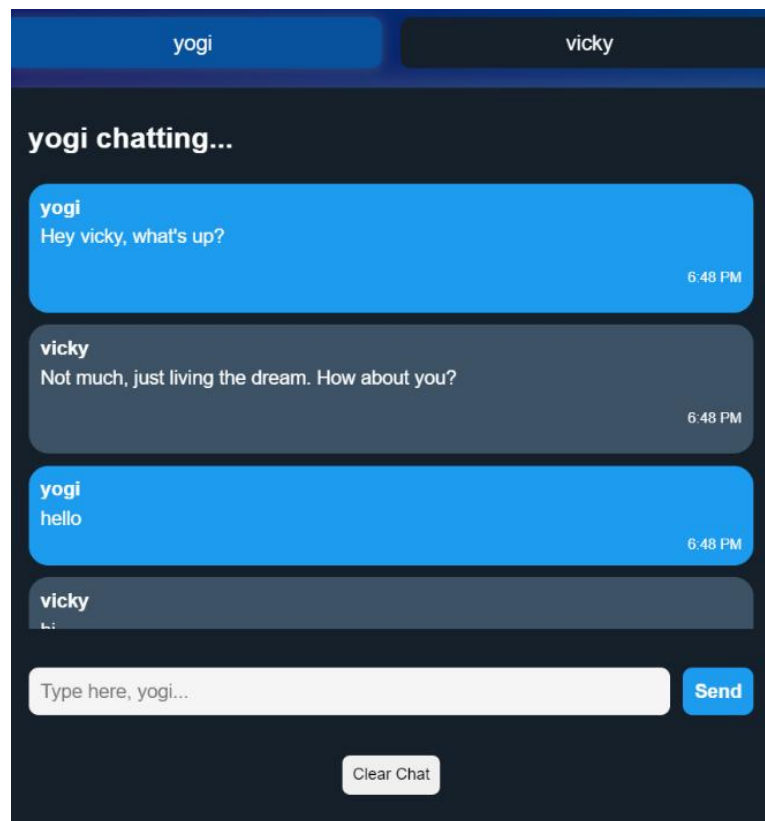


Figure 22. Live implementation of VoiceChat application

CHAPTER 7

CONCLUSION

Text-to-speech (TTS) is a technology that converts written text into spoken words. It enables computers, devices, and applications to produce natural-sounding speech output. TTS systems analyze the input text, apply linguistic rules, and use synthesized voices to generate speech.

These systems have numerous applications, including accessibility features for individuals with visual impairments, language learning tools, navigation systems, and automated customer service. Advances in TTS technology have led to more natural-sounding voices and improved intelligibility, making them increasingly prevalent in everyday devices and services. The integration of chat applications and text-to-speech functionality provides a promising approach towards the development of modern application development and deep learning.

FUTURE ENHANCEMENTS AND DISCUSSIONS

This chat application can be developed further in the future which involves the following features:

- Advanced Deep Learning Algorithms:

Better algorithms might evolve which can result in:

- Accurate voice cloning
- Higher inference speed
- Lower response time.

- End-to-end security:

Security can be embedded into the chat application using encryption algorithms such as E2EE (end-to-end encryption).

- Added emotions:

Emotions can be added to the messages to make the messaging functionality emotionally rich in nature.