From Impersonation to Authentication: Techniques for Identifying Deep Fake Voices

by

B.M Saqlain Rabbi
19101587
Erfanul Haque
19201009
Humayera Shayera Huri
21341026
Nuzhut Tabassum
23241091

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
October 2024

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Humayera Shayera Huri	B.M Saqlain Rabbi
21341026	19101587
Nuzhut Tabassum	Erfanul Haque
23241091	19201009

Approval

The thesis/project titled "From Impersonation to Authentication: Techniques for Identifying Deep Fake Voices" submitted by

1. Humayera Shayera Huri (21341026)
2. B.M Saqlain Rabbi (19101587)
3. Nuzhut Tabassum (23241091)
4. Erfanul Haque (19201009)

Of Summer, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October, 2024.

Examining Committee:

Supervisor:
(Member)

MD Tanzim Reza

Lecturer
Department of Computer Science and Engineering
BRAC University

Co-supervisor:
(Member)

Rafeed Rahman

Lecturer
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

The research is carried out in accordance with the BRAC University's laws and procedures on ethical norms. In our thesis, we use information from original sources. We're double-checking references and in-text citations for accuracy. We, the ve co-authors, take full responsibility for any violations of the thesis code. To address problems, we used a variety of literature and internet resources. We also recruited the help of several of our university's professors. Finally, we express our gratitude to everyone who helped us. To nish the thesis, we did not employ any misleading techniques. Our research complies with BRAC University's ethics norms.

Abstract

The proliferation of fake voices has become a concerning issue, making it increasingly challenging to distinguish between authentic and fabricated audio recordings. Notable examples include the replication of the voices of prominent U.S. President through the use of AI technology.

These manipulated audio clips have been disseminated across various YouTube channels, serving both benign and malicious purposes. While fake audio can be entertaining in content creation, it also carries a darker potential, including threats to political leaders and diplomatic relations between nations, potentially leading to conflict. In Bangladesh, the surge in fake audio content has left its populace in a state of skepticism, casting doubt on the authenticity of online videos and news reports. The citizens of Bangladesh have become susceptible to accepting counterfeit content as real, amplifying the need for a solution to this issue.

Dealing with this problem needs the use of machine learning, AI, and various algorithmic techniques to mitigate the spread of fake audio within Bangladesh. To tackle this problem, we propose utilizing diverse audio datasets and trained models, running them through specific algorithms to enhance accuracy in discerning genuine from manipulated audio. The main goal of our thesis is to implement three features that will help to detect the deep-fake audio spread across various social media affecting the lives of people and prevent scamming across various online banking transactions. In order to complete our desired project we had to work on various machine-learning models – namely Recurrent Neural Network (RNN), Long Short Term Memory (LSTM), bi LSTM and LSTM based RNN to detect fake audios across social media. Also using these models along with feature extraction like Mel-Frequency Cepstral Coefficients (MFCC) and Short-Term Fourier Transform (STFT), our study aims to create a strong system to differentiate between real and deepfake-audio. In spite of facing multiple challenges and achieving a better accuracy, we were successfully able to reach our desired goal.We accomplished our accuracy at 99% using Bi-LSTM for deepfake audio detection.

Keywords: Deepfake , Manipulated Audio , Machine Learning, RNN , LSTM, bi LSTM , MFCC , STFT, CQT.

Acknowledgement

First and foremost, we are immensely grateful to the Almighty Allah, who granted us the knowledge, strength, and patience to complete this research without any major setbacks. We would like to express our gratitude to our supervisor, Md. Tanzim Reza, whose guidance, support, and encouragement has been invaluable throughout the duration of this research. His continued encouragement and feedback helped us to refine our work and enhance its quality. We would also like to extend our gratitude to BRAC University for providing the facilities and resources required for this research. Finally, honorable mention goes to our parents and family, without whose unwavering support we wouldn't be able to achieve our goals. We are currently on the verge of graduating as a result of their kind assistance and prayers.

Table of Contents

List of Figures

List of Tables

Nomenclature

This list describes the symbols and abbreviations that have been used within the body of the document

DNN Deep Neural Network
RNN Recurrent Neural Network
LSTM Long Shot Term Memory
DF DeepFake
EER Equal Error Rate
XAI Explainable Ai
SVM Support Vector Machine

MFCC Mel Frequence Cepstral Coefficients

CQT Constant Q Transform
STFT Short Time Fourier Transform

TN True Negative
TP True Positive
FN False Negative
FP False Positive

LA Logical Address
PA Physical Address
ML Machine Learning

Chapter 1

Introduction

1.1 Introduction

Deepfake audio refers to highly convincing and manipulated audio content created by making use of machine learning and AI, that simulates voice of the real person. It can be misused by people in various ways including spreading misinformation, fake speech or statements from public figures, identity theft, fake emergency calls, social engineering attacks, extortion and blackmail. Nowadays scammers use text-to-speech to record phone conversations and use it against us. In Bangladesh, it is mostly seen while dealing with monetary transactions from banks and online. Criminals use manipulated audio to create fake recordings, which are then used for extorting money or blackmailing. In case of news media, certain reporters seeking popularity for their television channels and online platforms stream content that includes fabricated audio evidence concerning specific individuals, from undisclosed sources. In the prominent spheres of Deepfake technology, individuals can detect fake videos through diligent research and the application of Artificial Intelligence. However, distinguishing between authentic and deepfake manipulated voices remain challenging as technology advances, making it easier for the third-parties to pose threats to unsuspecting individuals.

1.2 Research Objective

The primary objective for this research is to create an advance detection system capable of detecting deepfake audio with high accuracy. By leveraging machine learning methods, including Support Vector Machine (SVM) and Long Term Short Memory (LSTM), this study aims to improve the precision of differentiating between genuine and manipulated audio. Additionally. The research seeks to strengthen the resilience of detection models by using a variety of dataset that covers various deepfake scenarios. The ultimate goal is to create a reliable tool that can mitigate- the risks posed by deep fake audio in areas such as media integrity, voice-activated systems. And system applications.

- Developing an advanced deep fake audio detection system by leveraging existing models and new dataset.
- Creating a system to compare original and deep fake audio to identify manipulated voices accurately
- Collecting and utilizing audio dataset covering various deep fake scenarios.
- Exploring and implementing RNN and LSTM algorithms to enhance detection precision.
- Providing a reliable solution for distinguishing between real human speech and AI generated fake audio.

1.3 Problem statement

The prevalence of deep fake audio technology has introduced new challenges in the realm of digital fraud and misinformation. The use of voice impersonation poses significant threats to individuals and organizations alike. Highly convincing fake audio recordings are created using advanced techniques such as voice synthesis and conversion, mimicking the voices of prominent individuals or the average citizens. Impersonators use these manipulated voices to commit crimes like financial fraud, data theft etc. The advancement of AI technology is making it increasingly difficult to distinguish between real and fabricated speech. Even sophisticated systems often fail to detect manipulated audios. In Bangladesh, fake audio recordings have been exploited to manipulate online narratives, perpetuate fraud, and to spread misinformation.

There are mainly 2 types of synthesized voices accessible. Non-DNN based synthesizers generate robotic voices, which can be easily identified. But more advanced DNN-based models create highly realistic fake voices, making detection increasingly difficult. As a result, traditional voice authentication systems are vulnerable to attacks. Though several models have been proposed, they often fail in real-world situations where impersonators exploit the subtle differences between human voice and synthetic voices. Addressing this gap requires robust methods for identifying impersonation attacks in audio-based systems, which remains a largely unexplored challenge.

Current methods rely on specific datasets that is not enough to cover new types of fake audio as it is highlighted in the ADD 2023 challenge paper [18] . Small sample sizes and lack of diverse data hinder the ability to create solutions that work well in various scenarios . Evaluations based on internal datasets may be biased impacting the reliability of results . The absence of consistent metrics across datasets and models makes it challenging to compare different solutions effectively. Solutions often perform well in ideal situations but lack robustness against adversarial attacks is not adequately explored. Improvement is needed in understanding and explaining how deep learning models work. Small research teams face limitations impacting the diversity of solutions.

Despite the numerous studies conducted on voice identification using deep learning, many of these approaches still struggle to accurately detect manipulated voices. There remains significant room for improvement in differentiating between authentic and synthetic voices. While some research has been performed in this field, detecting deepfake voices is currently an innovative challenge. Current models lack the ability to consistently distinguish real voices from fake ones due to shortcomings in feature extraction and prediction methodologies. Many existing voice detection techniques suffer from accuracy issues, often due to data loss during feature extraction or the lack of using proper methods. To address these limitations, our research introduces a model designed to effectively identify and differentiate between real and synthetic voices. [1].

1.4 Thesis Outline

This research focuses on the difficulties to differentiate between deepfake and real voice data using advanced models designed to outperform current methods. The arrangement for this research paper is outlined below:.

For the introduction part, Chapter 1 provides a summary related with research topics and motivation behind it. It discusses the key issues that led the authors to address this problem, along with an introduction to the methodology and objectives of the research.

In the literature review section, (Chapter 2) reviews and evaluates some relevant and noteworthy works done previously on voice impersonation detection. A critical assessment of the limitations of the prior studies and identification of flaws is also achieved here. This research aims to fulfill those gaps by coming up with a better model. A detailed discussion on the challenges in voice data processing is also included, which sets the foundation for the model developed in this study.

In Chapter 3, methodology is described where research starts to get into its original form. This part describes the structure and implementation of proposed models, which includes an LSTM-based Recurrent Neural Network (RNN). This methodology covers the process of gathering dataset, data processing, feature extraction, and model training.

The fourth section, Chapter 4, dives into the model's performance. This chapter presents in detail the results from the implemented LSTM, RNN and biLSTM models. Finally, a comparison is displayed between these results and present methods, and discusses efficacy of the models based on some evaluation metrics.

Chapter 2

Literature review

For our thesis, we studied a paper of authors Hira Dhamyal, Ayesha Ali, Ihsan Ayyub Qazi and Agha Ali Raza Where a deep neural model called Senet-50 , which employs the squeeze and excitation and residual layers with a Self-Attention Mechanism has been used in alongside with deep learning based audio audio synthesizers to generate a person's voice The experiment showing the lowest EER on both train and dev subsets - 7.54% on train and 11.9% on the dev data occurs when the top-16 most attended phonemes are used . The performance of vowels comes next , with an EER of 9.36% on the train set and 12.15% on the development set. The limitations that were discussed in this experiment were the consistency across weights which gives similar results,creating differences between actual phonemes used in the pronunciation and samples, due to lack of data and only limited to english language. The authors expressed that they want to explore more light weight ML models to enhance the ability and research in other languages[1].

In other study of Li Kai, Xugang Lu, Masato Akagi and Masashi Unoki where identifying generated fraudulent speech through advanced speech-synthesis techniques like advanced VC and TTS technologies were used for ASV systems protection.
The jitter and shimmer features , which show clear differences in the amount of jitter and shimmer between fake and real speech , are specifically examined in this paper to analyze the prosody information between genuine and fake speech . Of particular interest, the shimmer feature exhibits a significant dynamic variation for fake speech.
Accurate estimation of F0 for comparitively better jitter and shimmer feature representations for ensuring, they proposed by using 2 additional F0 for estimation method, For feature extraction , YIN and SWIPE are used instead of IRAPT method Additionally , by explicitly adding the shimmer and Mel-spectrogram characteristics , they created a DNN-FAD system . The Audio Synthesis Detection (ADD) 2022 and 2023 challenges datasets are used to assess the efficacy of the suggested approach for FAD .The experimental results show that both static and dynamic continuous shimmer features , traditional spectrum-based FAD systems can benefit from additional information particularly from those recovered by YIN and SWIPE algorithms .

During the test that was taken online of ADD2022 , EER decreased from 33.47 % to 31.50 % when VAD was absent, an improvement of 5.89% was visible. A combined weight of 3:2 produced a notable improvement on the online test of ADD2023 so the EER improved by 13.37 % falling from 41.29 % to 35.77%. [2].

The thesis paper that we read about was called "Deepfake Audio Detection via MFCC Features Using Machine Learning"[3]. The audio spoof and detection research has depended on ASVSpoof or AVSpoof data as well as several machine learning, deep neural network algorithms.

This study uses machine and deep learning methods in detecting deepfake audio. Using the MFCCs technique, we get the required features from audio. Deep learning framework was used to detect audio-deep fake detection; also using the LSTM based network, a sub-sample signal was predicted to improve the model.
Deep convolutional networks were currently used as a detection approach for the full replay attack method. This method has EER = 0 percent for ASVspoof2017 training and test dataset. Speech synthesis is the imitation of a digitized human speech usually through computer software or hardware. 1,000 sentences per second were synthesized by Lyrebird through deep learning models.

The present study showed how low accuracy the values were in identifying deep fake audios. To make sure the outputs are improved provided by machine learning models, feature-based methods were applied. The models used for this research paper were Random Forest, Support Vector Machine, Multi-Layer Perception and Extreme Gradient Boosting. After applying these models and MFCCs techniques, it was observed that the accuracy obtained was around 93 percent[3].

The thesis paper that we studied next was called "ASVspoof 2021: Towards Spoofed and Deepfake Speech Detection in the Wild" [4]. In this paper the authors discuss about the challenges of ASVspoof 2021 challenge, led by 54 international teams around the world.

The purpose of ASVspoof was to create an opponent for biometrics system which led to threaten several spoofing attacks. This paper provides with the insights and meta-data for the three-task database for ASVspoof and the challenges faced during research ideas and trends.

The tasks described below were LA, PA and DF. The task for ASVspoof 2021 Logical Access is to minimize differences between optimal laboratory conditions and spoofed speech across real telephony system, including VoIP and PSTN. Spoofed data were provided from ASVspoof 2019 LA evaluation database which was obtained from the VCTK database. The requirement of ASVspoof 2021 LA task was designing spoofing CMs that was simplified to unknown channels.

ASVspoof 2021 PA task aimed to promote development in generalization of Computer-Aided Systems (CM) trained and refined utilizing simulated attacks. Data effectively identifies presentation attacks happening in original physical environments. The Deepfake task is a collection of fake speech spoken processed with different lossy codecs for media storage.

The data is encoded and decoded to recover uncompressed audio, introducing distortions depending on the codec and its configuration. Main objective is to improve solutions for detecting deepfakes in condensed audio used in television, media hosted on news websites, and social media platforms.

The DF task does not require any involvement in the usage of the ASV system., and the initial matric for DF task is CM Equal Error Rate (EER). Study ranks baseline systems and submissions based on performance for the evaluation subset and primary metric. The evaluation subset is the EER for new DF tasks, with EER of 0.82 % , for LA and PA tasks, main matric is min t-DCF. Since 2021 additional anti-spoofing and also deepfake detection challenges and datasets were involved. First audio deep synthesis detection (ADD) challenge assessed with risks involved with low-quality and partially-fake audio, revealing the instability of synthesis detection models against lower quality and unseen fake audio. The spoofing-aware speaker verification (SASV) challenge provides protocols for the design of not just CM solutions but also ASV systems and alternative architectures for their combination. The findings of SASV point to the importance of enhancing the dependability of the CM and ASV subsystems, including cooperative optimization techniques.[4].

This document “ADD 2023: The Second Audio Deepfake Detection Challenge” gives an overall idea for providing technological advancements countering manipulated speech. . This challenge aims to advance audio deep fake detection recognizing the urgent need for strong solutions in an era where misleading audio can spread misinformation. In recent decades the progress of artificial intelligence has resulted in substantial improvements in the fields of speech synthesis and voice conversion technologies.Firstly, Spotting manipulated bits in audio files requires a good grasp of speech changes.
It’s a big step in getting better at telling real audio from fake. This foundational step is essentialfor differentiating between authentic and fake audio. The misuse of technology poses a threat to society. Detecting fake audio, known as deepfakes, is gaining attention. Recent efforts and challenges like ASVspoof 2021 and ADD 2022 are driving research in this area. Secondly, Finding where the audio has been messed with, not just knowing it happened . This mirrors real life situations and makes detection skills sharper .

It enhances detection skills by not only acknowledging manipulation but pinpointing the specific areas within the audio file that have been tampered with.

Such precision is vital for practical applications in real world scenarios.We make use of AISHELL 3 corpus a comprehensive collection of chinese speech that comprises more than 88000 utterances and 85 hours of speech . The evaluation dataset contains samples from and unidentified deepfake algorithm. Deepfake algorithm recognition (AR) aims to identify the algorithm of the deep deepfake speech.

Track 1.2: The real audio of the test set for two rounds consists of sources including AISHELL-1, Thchs30, etc.utilize the macro-average F1-score in open set recognition. LFCC LCNN system (baseline S02) operates on LFCC features with a light convolutional neural network (LCNN). Thirdly, figuring out the tech behind making a fake audio staying ahead of new methods. Understanding these technologies makes our defenses stronger.Understanding these techniques is crucial for staying ahead of evolving methods, ultimately strengthening defense mechanisms against emerging threats.

This proactive approach ensures that detection solutions remain effective against new and sophisticated manipulation techniques.The wav2vec2 pretrained model variant "wav2vec XLSR" is used as a pretrained feature extractor, which is trained on 56k hours of audio samples in 53 languages using additional linear transformations and a larger context network.

Baseline SO5, based on Resnet 18, employsas softmax theshold technique for distinguishing known and unknowns deepfake algorithms in speech signals. Baseline SO6 utilizes the open set recognition method from OpenM.
The document underlines the significance of datasets as practice fields for algorithms, encompassing various fake audio scenarios. Careful selection of datasets ensures that challenges presented align with real world situations, enhancing the practicality of the solutions developed during the challenge. In a nutshell, ADD 2023 guides researchers in fighting fake audio. It checks current skills and pushes the field to handle future challenges. In our evolving tech world, challenges like ADD 2023 help us defend against tricky audio fakes.

ADD 2023, the second audio deepfake detection challenge, confronts pivotal issues in audio deepfake detection. it goes beyond binary classification, focusing on nuanced challenges like localizing manipulated intervals, pinpointing the source of fake audio, and addressing real world threats. With subchallenges such as audio fake game and Manipulation Region Location diverse datasets, and tailored evaluation metrics , ADD 2023 aims to stimulate innovation in detecting AI generated audio , fostering advancements with practical implications.

ADD 2023 uses three sub challenges audio fake game (FG) , manipulation region location (RL), and deepfake algorithm recognition. FG involves two tasks audio fake game generation (FGG) and audio fake game detection (FGD) RL focuses on locating and manipulated regions in partially fake audio. AR aims to recognize the algorithms of deepfake utterances. Detailed information on training, dev sets, and

test sets for each subchallenge. Six baseline system are provided for the detection tasks, including GMM based and CNN based models.

The limitations of ADD 2023 include dependence on predefined datasets , potentially overlooking emerging audio deepfake variations. The challenges complexity may discourage smaller teams, limiting solutions diversity, generalizing proposed solutions to real world scenarios and assessing their practical applicability raises concern[5].

The paper "Wavefake: A dataset to facilitate Audio Deepfake detection" introduces a novel dataset for audio deepfake detection and provides an in depth analysis of multiple state of the generative models. Thisataset comprises samples form six different network architectures across two languages allowing for comparisons of audio clips between the architectures. The authors Joel Frank and Lea Schönherr analyze the subtle differences between generators, focusing on samples that resemble the training distributions. They also investigate the performance of baseline models, including Gaussian Mixture Models (GMM) and neural network-based solutions, across different dataset and settings .

The finding suggest that while neural networks perform better overall, GMM classifiers demonstrate greater robustness.The paper also discusses the potential implications of releasing research into detecting deepfakes and emphasizes the importance of not withholding such research. Additionally, the authors provide a popular attribution method for practitioners to inspect their predictions when building on the results.

This paper serves as an invaluable contribution to the field of audio deep fake detection, targeting both researchers and professionals. The paper titled "Wavefake: A Dataset to Facilitate Audi Deepfake Detection" identifies a significant gap of research in deep fake detection, particularly the lack of focus on audio manipulations. The authors of this paper Joel Frank and Lea Schonherr tried to address this gap by presenting a specialized dataset designed for detection of deep fake audio and by performing a comprehensive evaluation of various advanced generative models. The author acknowledges a limitation in the dataset's scope which may become a problem. Lastly, they highlight the importance of testing classifier robustness, given the presence of numerous adversarial attacks on automatic speech recognition systems and the potential impact of common audio perturbations.[6].

We also studied the paper "Detecting Synthetic Speech Manipulation in Real Audio Recordings" which talks about the problem of spotting fake speech in real audio recordings . It mentions improvements made by RawNet2 and RW- ResNet and successfu training methods like one-class (OC) loss and large margin cosine loss (LMCL). The authors said that while one system may not be great at everything , combining systems has shown good results overall, even though it requires more computing power. To test the ability to detect fake speech, the researchers used two sets of data. The first set included ASVsproof2019 LA and FoR datasets, and the second set had partially fake speech from the partialSpoof database, Semantic Forensics (SemaFor) internal program evaluation 1, SemaFor hackathon2 data and

some datasets were created in house.

Additionally, the paper talks about new developments in fake speech and audio technologies like FastsSpeech 2 , Flowtron, Natural TTS synthesis, Fastpitch and RAD-TTS. It brings up the ASVPoof 2015 challenge and other studies. By comparing how well various systems perform on completely generated and partially fake datasets , it also mentions other studies on preventing deception and detecting fake speech. Antispoofing Gathering datasets that have partially fake speech , including ASVspoof2019 LA , FoR , PartialSpoof, Creating a detection system using the x-ResNet architecture and speech Activity Detector(SAD) for spotting fake speech , then teaching the models using deep learning methods like one-class loss and large margin cosine loss , then checking how well they do on the collected datasets. Doing tests to see how effective the system is at finding parts of fake speech in real audio recordings. Looking at the experimental outcomes to understand how well the system works and comparing it with existing methods. Figuring out where improvements can be made and suggesting areas for future research like dealing with fake audio in real recordings[7].

Next we studied the paper titled "Audio Deepfake Detection by Using Machine and Deep Learning", addresses the critical issues of fake voices, a growing concern in cybersecurity, forensics, and social media. The paper discusses the rise of audio deepfake technology and its potential for causing harm, citing an example of a 35 million dollar transfer authorized by a bank manager due to a cloned voice. Additionally, the paper discusses the wide spectrum of content, from memes to hate speech generated using AI voice cloning technology. The increasing use of artificial intelligence, exemplified by products like Google Audio LM, enables the creation of highly realistic synthetic audio content with potential applications ranging from speech disorders to fraudulent activities. The study aims to detect imitated sound sequences produced using Google Audio LM through feature extraction, ML, and DL methods. Proposed approach underlines extraction for features through Mel-Frequency Cepstral Coefficient, followed by classification using both ML and DL models. Detection model provides potential applications across domains ranging from social media platforms, cybersecurity, to mobile devices.

The framework for detecting fake audio is structured around a few key steps: extracting audio features, training models with machine and deep learning techniques, and calculating prediction possibilities. Moreover, the authors reviewed existing research in audio deep fake detection. Their review covered research works focusing on areas like; bi-spectral analysis for synthetic speech detection, evaluation of acoustic features, analysis of silent segments in generated speech, semantic-based techniques, and folding methods for identifying manipulated audio.The paper provides an overview of the current state and challenges in synthetic speech detection, highlighting the need for a more comprehensive approach that considers different languages, accents, and audio characteristics.

The proposed audio deepfake detection model in this research utilizes Mel frequency features to represent sound perception and employs machine learning and deep learning models for classification, The model uses the Librosa library to convert audio files into audio objects which are then divided into smaller sizes for analysis.

MFCC method is applied to extract audio object attributes, and various windowing techniques are utilized to reshape audio streams into frames. Deep learning algorithms such as CNN and MLP, as well as machine learning algorithms including SVC, Decision Tree, and others, are employed for fake or real audio detection, with final decisions based on majority voting. The results from the detection module are then conveyed to different platforms through the Audio Deepfake Notification Module.

They used 28 distinct speech data sets as a dataset. Determining accuracy for eight ML models in the supervised learning , the study ran them through 10 iterations . Among the models were KNeighorsClassifier, AdaBoostClassifier, DecisionTreeClassifier, Support Vector Classifier (SVC), and MLPClassifier. With the accuracy from 0.76 to 0.81 the DecisionTreeClassifier displayed variations whereas the SVC consistently displayed an accuracy of 0.79.
The moderate accuracy levels that the AdaBoostClassifier generated ranged from 0.75 to 0.78 . Accuracy values for ExtraTreesClassifier and GradientBoostingClassifier were 0.85 to 0.86 and 0.84 to 0.85 respectively indicating strong and consistent performances. The best performing model was the MLPClassifier , which demonstrated competence in identifying complex data patterns with an accuracy of 0.86 to 0.88. All models were trained with default parameters using Pythons sci-kit learn package . The accuracy rates of the researchers MFCC-based detection ranged from 75 to 88% when tested with real and fake speeches using eight different classification techniques . The focus of future research will be on improving algorithm parameters and examining cybersecurity. Furthermore, the researchers want to improve and test the model under different distortions , such as background noises, clipping, echos and speaker environment consequences . According to the paper , robust audio deepfake detection systems are necessary to prevent the bad use of AI speech generators and maintain the benefits of this society.[8]

The emergence of deepfake technology poses a significant threat in various domains, including voice-based authentication, fraud prevention, and misinformation detection. This paper, authored by Suk-Young Lim, Dong-Kyu Chae, and Sang-Chul Lee, titled "Detecting Deepfake Voice Using Explainable Deep Learning Techniques", explores the limitations of artificial intelligence methods for deepfake voice detection and addresses the challenge of detecting deepfake voices through the application of explainable deep learning techniques. the authors underscore the need for advanced detection methods and introduce their unique contribution- a deep learning approach that not only distinguishes genuine from manipulated voices but also offers transparency through explainable techniques.

The focus is on providing interpretability, for non-experts in linguistic and artificial intelligence. Firstly, the authors provide a comprehensive review of existing methods for voice synthesis, highlighting the need for sophisticated detection mechanisms to

combat the evolving landscape of deepfake voice technology. The core of this paper presents the authors' novel model architecture for detecting deepfake voices.

Leveraging explainable deep learning techniques, the model not only classifies audio as genuine or manipulated but also provides insight into the specific features contributing to the decision. This transparency enhances the trustworthiness of the detection process and enables better understanding and validation by experts.

The study uses simple model structures like convolutional neural networks and LSTM for interpretation delivery. It conducts experiments on ASVspoof 2019 Logical Access and LJSpeech datasets, each exclusively representing a near real-world dataset and constrained dataset. The trained model is analyzed using popular XAI methods such as Deep Taylor, integrated gradients, and layer-wise relevance propagation. The paper discusses how the addition of LSTM changes the attribution score distributions. The study demonstrates that XAI methods applied to deepfake voice detection enable understanding level of human's perception. It also compares proposed model with existing deepfake voice detection methods. This paper also provides insights into the differences between model and human perception in detecting deepfake media. It also discusses the potential for future studies to alleviate public anxiety towards generative models creating fake media. The paper concludes by discussing potential applications of the proposed model in real-world scenarios, such as fraud prevention, voice-based authentication, and content moderation. The authors also consider the ethical implications of deepfake technology and emphasize the importance of responsible AI development. Grants from the Korean government and the DGIST R and D program supported the paper. In anticipation of the evolving landscape of deepfake technology, the authors outline potential future directions for research. They discuss avenues for refining the proposed model, expanding the dataset, and adapting to emerging threats in the field of deepfake voice manipulation. The authors emphasize the significance of their explainable deep learning approach in addressing the challenges posed by deepfake voice technology, providing a robust and transparent solution for detection.[9]

Researchers have increasingly focused on developing advanced techniques to detect deepfake audio, and one such approach is explored in the paper "Unmasking the Truth: A Deep Learning Approach to Detecting Deepfake Audio through MFCC Features" by Islam Altalahin, Shadi AlZu'bi, Assal Alqudah, and Ala Mughaid. The paper "Unmasking the truth: A deep learning approach to detecting deepfake audio through mfcc features" gives an overall idea about the newest developments, providing importance on the use of MFCCs. After a thorough evaluation, the writers showed how capable their way of approach is, also giving importance to achieve the features of MFCC to detect deepfake audio.
Going through this paper, there are many techniques used to identify deepfake voices, which includes spectrogram analysis, statistical approach, synthesis speech detection and speaker verification. Spectrogram is more reliable to differentiate deepfake audio through deep learning methods. Speaker verification makes use of sounding and verbal characteristics to accomplish greater accuracy to differentiate between real and deepfake audio. Statistical methods are used to detect deepfake audio, such as Gaussain mixture models.

Moreover, to differentiate between authentic and generated speech, synthetic speech detection analyzes phonemic and linguistic elements.
Using Convolutional Neural Networks (CNNs) with Long short term memory network (LSTMs) the author detects the deepfake audios . For handling sequential data both speech and audio this design works effectively . Mel-Frequency Cepstral Coeffiecient (MFCCs) feature extraction was used . This feature extraction captures special characteristics of audio signals. Different sound generation techniques leave unique sound effects which MFCCs with preserved temporal properties of sound detects perfectly .

The challenge 2019 dataset is divided into two groups : scenarios involving physical access (PA) and logical access (LA). Both followed the VCTK dataset ruleset. The data undergoes two preprocessing steps before training: OneHotEncoding and Standard Scaling. These steps ensure the audio is properly formatted for machine learning models. The hybrid CNN-LSTM model is used to process and learn patterns from sequential data. The CNN layers extract local patterns from the data, while the LSTM layers learn the temporal dependencies in the extracted features. This combination allows the model to recognize both localized patterns and long-term relationships, making it well suited for tasks like speech and audio processing.

Performance metrics essential for evaluating the model include accuracy, confusion matrix, precision, recall, and F1-score. This research differs from previous ones by combining three feature selection techniques with three classification systems. Here authors used CNN-LSTM classification to classify sounds as real or fake based on the features of the sound extracted by the MFCC feature. The results showed that the methodology used was effective in detecting audio deepfakes using CNN-LST¡ with an accuracy of 88 percent outperforming other machine learning models. This result shows the effectiveness of the CNN-LSTM model as a promising choice to address the challenges faced by deep fake voices.[10]

Next we learn about synthetic speech to the generator used synthesize it from "SYN ATT : Synthetic speech Attribution via Semi Supervised Unknown Multiclass ensemble of CNNs". The detector transforms audio into logmel spectrograms extracts features using CNN, and classified between known and unknown algorithms.And utilizing it to semi supervision ensebling methods to improve robustness and generalizability significantly. Author Md Awsafur Rahman, Bishmoy Paul, Najibul Haque Sarker, Zaber Ibn Abdul Hakim,Shaikh Anowarul Fatta and Mohammad Saquib validates the proposed method on two evaluation datasets outperforming other top teams in accuracy by 12 to 13 percent on one dataset and 1 to 2 percent on another.
The approach semi supervises training and ensemble methodologies to enhance model robustness and generalizability demonstrating remarkable effectiveness in synthetic speech attribution . These finding provide evidence of the effectiveness of the proposed approach in addressing forensic concerns linked to malicious synthetic speech use.So they gathered different types of synthetic speech samples including known and unknown algorithms . then converted the audio data in to a format that can be used for analysis .

By creating a model using a type of AI called convolutional neural network to classify the synthetic speech samples. Used a method called semi supervised learning to improve the models performance .
As for conclusion and future work summarizing the finding and suggesting areas for future research and improvement highlighting the proposed approach in addressing the forensic concerns.[11]

The EmoFake project talks about the problem of detecting emotion fake audio which can give threat to peoples lives if it is misused . The dataset includes fake audio and real emotional speech and a Graph attention networks using Deep Emotion Embedding (GADE) method and it demonstrates good performance in detecting emotion fake audio. The authors emphasized the importance of detecting emotion fake audio and highlighting the limitations of existing datasets in considering the specific type of fake audio.
Theya also reference different kinds of voice conversion techniques commonly used in emotion voice conversion tasks , While there have been some datasets developed for detecting fake audio as the ASVproof and ADD challenges none of htem consider the specific case of emotion fake audio. The project aims to fill the gap introducing the EmoFake dataset which includes emotion fake audio generated by emotion voice conversion models.
The method Graph Attention Networks using Deep Emotion embedding (GADE) uses deep emotion embeddings to capture the emotional content of the audio and graph attention networks to model the relationships between the embeddings.EmoFake dataset and compared to existing fake audio detection models such as LCNN , RawNet2 , SENet, and ReNet34. The results show that their model outperforms these models in detecting emotion fake audio with and EER of 0.76 percent for neutral to happy conversion 0.55 percent for neutral to surprise conversion 0.52 percent for neutral to angry conversion and 1.43 percent for neutral to sad conversion.Overall, this project talks about the importance of detecting emotion fake audio and proposes a new dataset and method for addressing this problem , The model GADE method demonstrates promising results and could be further developed to improve the detection of emotional fake audio in real world scenarios. [12]

The next thesis paper we studied was called “Deepfake audio as a data augmentation technique for training automatic speech to text transcription models” [14]. In this paper, the authors used various datasets to train the transcription model. While training these datasets they faced a lot of challenges like not getting enough data for their research and using up a lot of time to transcript data manually.
Also, while collecting audio data for different languages became more difficult to obtain data due to the availability for that particular language. Another problem they faced were the background noise for that sound where was difficult to cancel out unwanted sound. To solve these issues, the authors suggested to apply deepfake audio as data expansion technique to improvise the robustness of the transcription model.

It uses a voice cloner to generate audios for the same speaker with different speech contents, increasing and decreasing pitchiness and record different pitched audio for the same person and preserving the original voice of that person. The expansion dataset of the trained data is used in transcription models to improve consistency of the outcomes and also improvising the training model to acquire wide range of audio patterns. Deepfake audio also the cost and time required to obtain different datasets. But it lefts with a question about the use of these technologies in the wrong hands. [13]

This paper conducted by Ariel Cohen , Inbai Rimon , Eran Afiaio and Haim Permuter addresses the challenges that faces voice anti spoofing systems particularly in the challenges of the ASVpoof 2021 challenge . The authors identify audio based systems to channel variability compression and transmission effects which significantly degrade system performance . This problem revolves around the need to enhance the robustness of anti spoofing systems to these challenges including different audio compressions bandwidth differences and unseen spoofing attacks . The workflow of the project involves data centric approach , the authors focuses on amking changes and improves in the dataset to maximize results while keeping the models fixed .

This study introduces two forms of data augmentation compression and channel augmentation for the logical access category .Additionally a new type of online data augmentation. SpecAverage is introduced to improve generalization. The authors also propose a Log spectogram feature design that enhances system performance .These methods the study aims to addres the challenges faced by anti spoofing systems and improve their performance in detecting synthetic or spoofed audio.

In summary , this research explores different ways to make voice anti spoofing system better especially for the ASVspoof 2021 challenge and it deals with problems like changes in audio quality different types of audio attacks and variations in how audio is transmitted . The study deals with problems like changes in audio quality different types of audio attacks and variations in how audio is transmitted. The authors showed that by improving the dataset rather than changing the models , they could make the systems much stronger. These findings are exciting because they can lead to even better systems for recognizing speakers and preventing spoofing attacks in various situations .[14]

Another thesis paper we studied was called "MIS-AVoiDD: Modality Invariant and Specific Representation for Audio-Visual Deepfake Detection" [15]. In this research paper, the authors used a new approach called MIS-AVoiDD, which is used to detect audio deepfakes to overcome distributional modality gaps and also improve performance. MIS-AVoiDD learns audio data through modality-invariant and modality-specific subspaces, to retain the common patterns and combine various audio data for deepfake manipulation detection.

MIS-AVoiDD uses multimodal deepfake detection approach to increase accuracy of audio streams. This detection approach learns audio representations through modality-invariant and modality specific subspaces, apprehending both common and unique patterns. Fusion of audio streams uses multimodal CNNs and voting schemes to improvise deepfake detection performance. The MIS-AVoiDD approach can cause harm to society like political and financial manipulation. It could be used to detect and prevent spreading false information, fake news and fraud through various social platforms, news organizations and through financial institutions.[15]

The paper titled "Deepfake Audio Detection with Neural Networks using Audio Features" by Abu Qais, Akshar Rastogi, Akash Saxena, Arpit Rana, and Deependra Sinha, proposes a speech spoofing detection system using convolutional neural networks (CNN) and audio features to differentiate between human and deepfake audio. The authors aim to detect deepfake audio with lower computational requirements by transforming audio signals into images of audio features like spectrogram, MFCC, FFT, STFT, and feeding them into a CNN model. The ASVspoof 2017 dataset is used by splitting it 80/20 into train/test. Previous works have developed automated speech recognition and anti-spoofing detection using deep neural networks with CNN to learn from spectral audio features. Raw input audio suffers from high dimensionality in deep models. MFCC has provided useful representations capturing timbre and formants and proved effective previously in spoofing scenarios. Fusing various features like spectrogram, FFT, MFCC etc. has shown promise for enhanced detection. Analysing the results from this research we see, that the MFCC features outperformed others with 75.97 percent accuracy in detecting spoofed audio using a CNN with ReLU activations.

Concatenating all feature arrays gave the best performance with 80.47 percent accuracy. MFCC had more vibrant images providing more patterns for the CNN to learn from. The limitations noted are the dated dataset and the potential for improved performance using recurrent or graph neural networks. Key contributions of this paper are demonstrating MFCC's detection efficacy, the concatenated feature approach, and computational efficiency gains from transposing audio inputs as images. This research aligns with the trends in optimizing CNN feature fusion architectures. Innovation lies in the image-formatted audio feature input approach to ease the computational load. Future work is suggested around robustness under noise, model interpretability, and exploring recurrent/ graph networks. Overall the paper highlights promising directions to work out generalizable and efficient deepfake audio detection solutions via multi-facated neural frameworks.[16]

This paper authored by Basant Kumar and Shatha Rashad Alraisi, titled "Deepfakes

Audio Detection Techniques Using Deep Convolutional Neural Network", discusses using Convolutional neural networks (CNNs) for detecting deepfake audio. Deepfakes refer to synthetic media that aims to impersonate real people, often using generative adversarial networks (GANs). While much attention has been focused on deepfake images and videos, audio deepfakes remain neglected comparatively. Detecting them is an important challenge. The paper summarizes previous work done on the detection of audio deepfakes. RNNs have been applied for audio analysis but face issues with long-term dependencies and hardware compatibility. CNN models designed for images have been adapted for audio spectrograms, using directional kernels to capture temporal patterns.

Models using raw audio waveforms as input have also been developed. The paper aims to evaluate different CNN architectures for classifying real vs fake audio effectively. Previous references cover audio deepfake generation methods like GANs and VAEs, prior deepfake audio detection research, and CNN architectures like XCeption net. The DFDC dataset provides training data for face swap videos.

Features like Mel-frequency cepstral coefficients (MFCCs) help represent audio for classification. The paper takes note of the lack of standard benchmarks for comparing audio deepfake detection techniques as an ongoing challenge. The methodology involves testing different CNN models on audio datasets to classify real vs fake samples, and measuring accuracy including CNN with MLP using UrbanSound8k, CNN with XCeption net, and ResNext using the DFDC dataset and, performance metrics and validation schemes. Novel aspects of this paper include applying XCeption net for audio for the first time. The paper also examines using an ensemble of ResNext and XCeption together. The XCeption net architecture achieved 90 percent accuracy, outperforming MLP and ResNext models. The wave patterns between real and fake audio are discernible. The model training time was under 20 minutes, enabling real-time detection. This paper highlights the growing threat of audio deepfakes for fraud, copyright infringement, and misinformation campaigns. Developing fast, accurate detection tools is crucial. The findings of this research demonstrate that CNNs can reliably differentiate fake audio. In conclusion, the paper makes a novel contribution in demonstrating how CNN architectures can enable real-time audio deepfake detection to address an important and evolving threat.[17]

This paper, Deepfake audio detection via MFCC features using Machine Learning [3] discussed about the Automatic speaker verification (ASV) systems getting vulnerable by malicious attacks where attacker provides fake information that were biometric to trick the ASV systems through machine helped by speech generation such as voice conversion, Outputs from text to speech synthesis systems that have been altered and customized. Therefore, it was crucial to differentiate between speech that was human generated and speech that was fake. The authors therefore chose to examine impersonation attacks on the both black and white box of ASV systems . They discovered that in the majority of attacks P/SP based black boxes and white boxes exhibit a significantly higher ASV EER percent than STFT or MFCC based methods.

When it comes to SP-based boxes , this is even more evident. It is possible to hypothesize that both AP and SP are characteristics that correlate to identify independent attributes rather than content dependent ones. Even the identities of the speakers are largely separated from SP. The results are expected because the AP and SP signals are chosen to capture the details of the differences between original speech and fake speech, while ASV systems require features that are more accurate in distinguishing the speaker voice characteristics. Nonetheless, the paper's fascinating conclusion is that AP/SP features, specifically AP seems to be helpful supplementary information that lowers the EERpercent for attacks in which STFT and MFCC lack experience.

Our theory that they could identify signature information to differentiate between machine generated and human generated speech is supported by this . Chenglong Wang, Jiangyan Yi, Cunhang Fan and Jun Xue's paper.Dan Zhang and Zhao Lv discussed self distillation with Zhenqi Wen . Approach to fake speech detection FSD which can greatly enhance the effectiveness of FSD without making the model more complicated . Consequently , the authors decided to add classifiers behind the shallow network to build engagement in both feature and prediction with the deepest network dimensions, which improves shallow networks capacity to gather specific discriminative data.
Reducing the disparity between deep and shallow features is the goal of feature distillation, and by maximizing the information in each network layer to optimize the network is the main goal of prediction dimension distillation here.
The EER of the baseline "SENet34" is 1.14 percent , which is the worst performance in the overall baseline. However,the "SENet34(SD)" system could obtain an EER of 0.65percent , making it the best-performing system. With the use of different architectures of ECANet and SENet, the LA and PA dataset results validate the effectiveness and generality of the approach, significantly improving the performance of the baseline.

In the paper of Hira Dhamyal, Ayesha Ali, Ihsan Ayyub Qazi and Agha Ali Raza, though there were some promising results, some limitations including the consistency across weights leading to similar results, potential discrepancies between sample and actual phonemes in pronunciation due to insufficient data, and the study being confined to the English language. The problem revolves around the need of alternative model architectures to increase the robustness and generalization of deep learning-based audio synthesizers, especially when applied to languages beyond English. The problem addressed in Li Kai, Xugang Lu, Masato Akagi and Masashi Unoki's paper is about the identification of fraudulent speech generated through advanced speech synthesis techniques where the study introduces a deep neural network-based fraudulent audio detection (DNN-FAD) system by explicitly combining shimmer and Mel-spectrogram features. However, the problem of the model is the reliability of the ASV systems in FAD systems that includes the robustness of the feature representation method as well as the effective DNN-based design system to reduce the risks of fraudulent speech in ASV applications. The 3rd paper showed more robust detection of spoofed speech, and result in rendering ASV systems more robust to attacks generated using unseen methods. The model will not work on more low-parameter and highly robust FSD systems for now. [18]

The paper "Voice Impersonation Detection using LSTM-based RNN and Explainable AI" by Kawshik Barua, Abdur Rahim, Prantozit Saha Parizat, Md. Asad Uzzaman Noor, and Miftahul Jannah address the important problem of detecting synthetic or impersonated voices generated by advanced technologies. This research introduces RNN model based on LSTM to differentiate between genuine and fake audio. The authors argued that previous research has not tackled the vanishing gradient issue associated with audio inputs adequately. According to them this can lead to information loss and hinder the performance of detection models.

From this paper "Voice impersonation detection using lstm based rnn and explainable ai", the authors suggested using LSTM to improve accuracy to identify deepfake voice. The authors produced 8 key features from the provided audio files to train the models like MFCCs, Zero-Crossing Rate (ZCR), Spectral Roll-off, Chroma vectors, Mean Square Energy and Spectral Centroid.

The authors extracted their features, processed and used them to train the LSTM based RNN model. Also, they carried out SVM classifiers as a means for comparison. For both the LSTM based RNN models and SVM, the authors applied Local Interpretable Model-agnostic Explanations as an Explainable AI technique.

The models trained using a dataset available publicly, FoR (Fake or Real). This dataset contains Ai-generated fake audios and authentic human voices. Outperforming the SVM classifier evaluating the same dataset The LSTM based RNN achieved greater accuracy of 98.33%.

LIME was used to explain the model predictions to clarify how the different features influence the results of classification. It makes the model more transparent and deserving . All-inclusive this research presents a novel method for detecting voice impersonation by combining the strengths of an LSTM based model with Explainable AI. It provides enhanced interpretability and performance .
[19]

Chapter 3

Methodology

3.1 Proposed Approach

For this research, we focused on developing a model using Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) and Bidirectional LSTM (Bi-LSTM) to compare between real and deepfake audio. We used a Bengali dataset from "Mozilla Common Voice", which solely focuses on the people whose ages are between 20 to 29. We also extracted our audio using the key features such as Mel Frequency Coefficients (MFCC), Constant Q Transform (CQT) and Short-Term Fourier Transform (STFT) to grab temporary patterns and harmonic patterns. We used label encoding and one-hot encoding for data preprocessing, afterwards we splitted our dataset (80% training, 20% testing) and padded them to give a smooth sequence length. Our features were also scaled using StandardScalar to increase our model consistency. The LSTM model captures temporal dependencies in sequences, while BiLSTM processes data in both forward and backward directions, offering better context awareness. RNN introduces recurrent connections to retain information across time steps, though LSTM and BiLSTM are better at handling long-term dependencies. All three models were trained on the dataset and to get the measure such as accuracy, precision, recall, and F1-score. A confusion matrix was used to analyze correct and incorrect predictions. The use of BiLSTM provides an advantage in learning more complex temporal patterns, ensuring improved performance. This research offers a novel approach to deepfake audio detection for Bangla speech, contributing to the growing need for reliable detection methods across diverse languages.

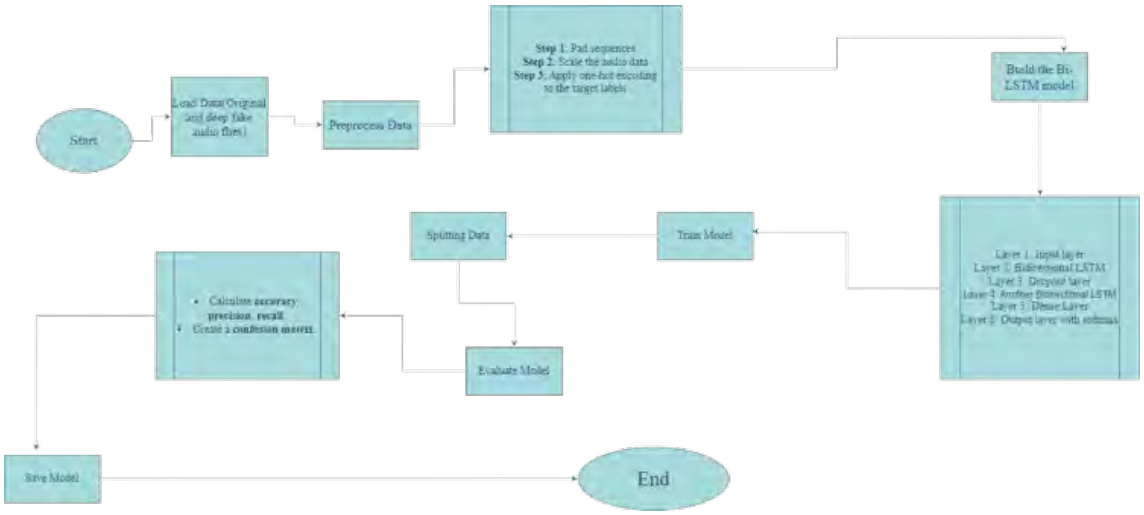

Figure 3.1: BiLSTM Workflow

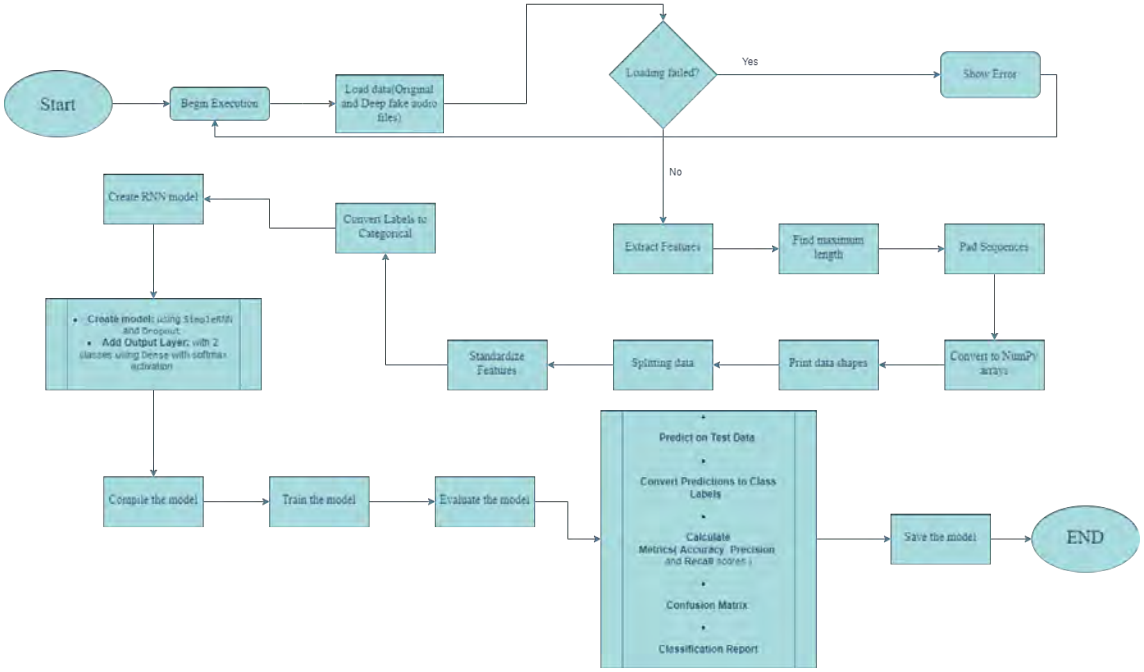

Figure 3.2: RNN Workflow

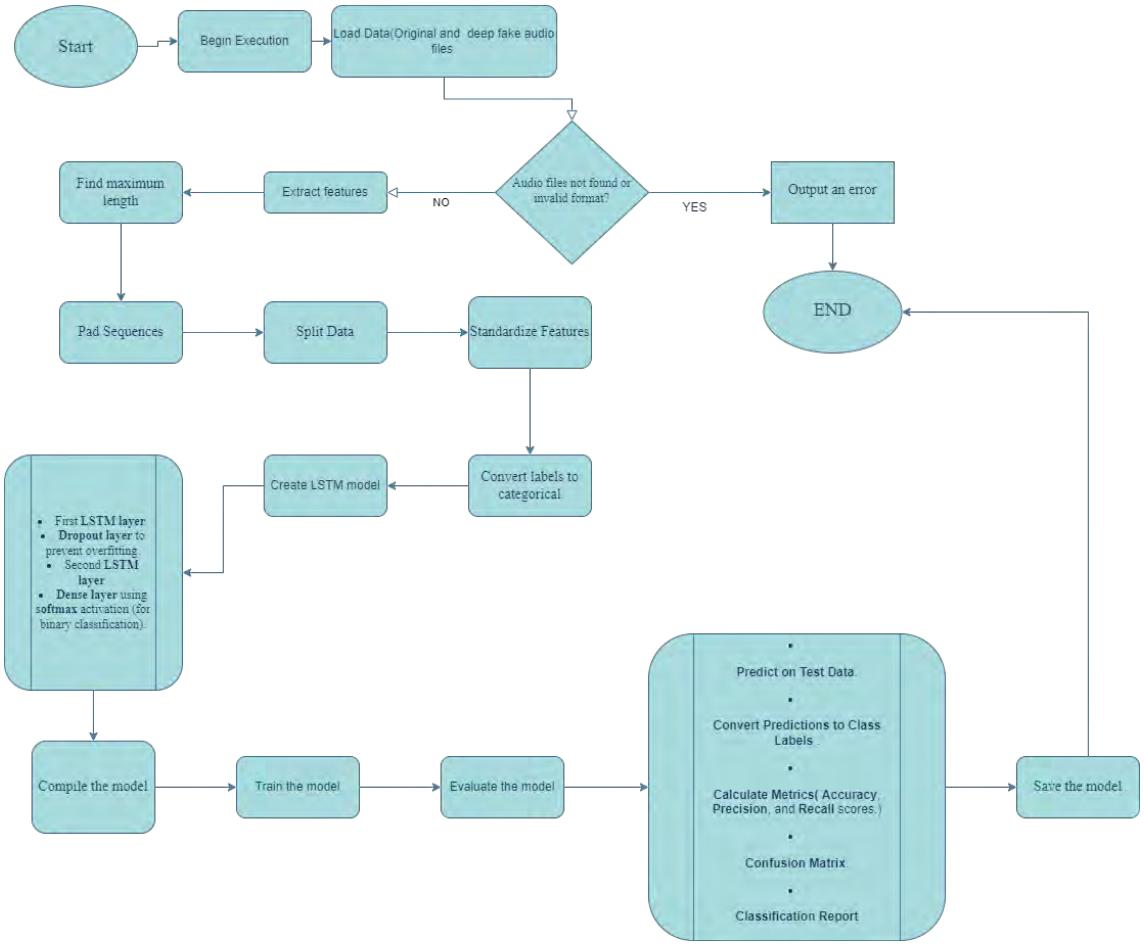

Figure 3.3: LSTM Workflow

3.2 Dataset Description

We used a raw and existing dataset for our model. The dataset consists of over 1800 audio clips gathered from random people. For the existing dataset of 7200 data, we collected it from Mozilla Common Voice. [20] All our datasets are in Bangla. Each one of the audio clips in the dataset is a unique target. The dataset is the vast significant Bengali dataset available that is about the number for audio clips. The dataset provides real monologues and a few 2-way conversations. Our dataset was made from scratch primarily for machine learning purposes. Moreover, we generated deep fake audios from both dataset. This was achieved with the help of a voice cloning pipeline that combines audio processing, machine learning, and file manipulation to create a system for voice cloning and deep fake audio generation. The use of libraries like pyworld and pydub allows for sophisticated audio analysis and manipulation, while TensorFlow/Keras provides the framework for the deep learning aspects of voice conversion.

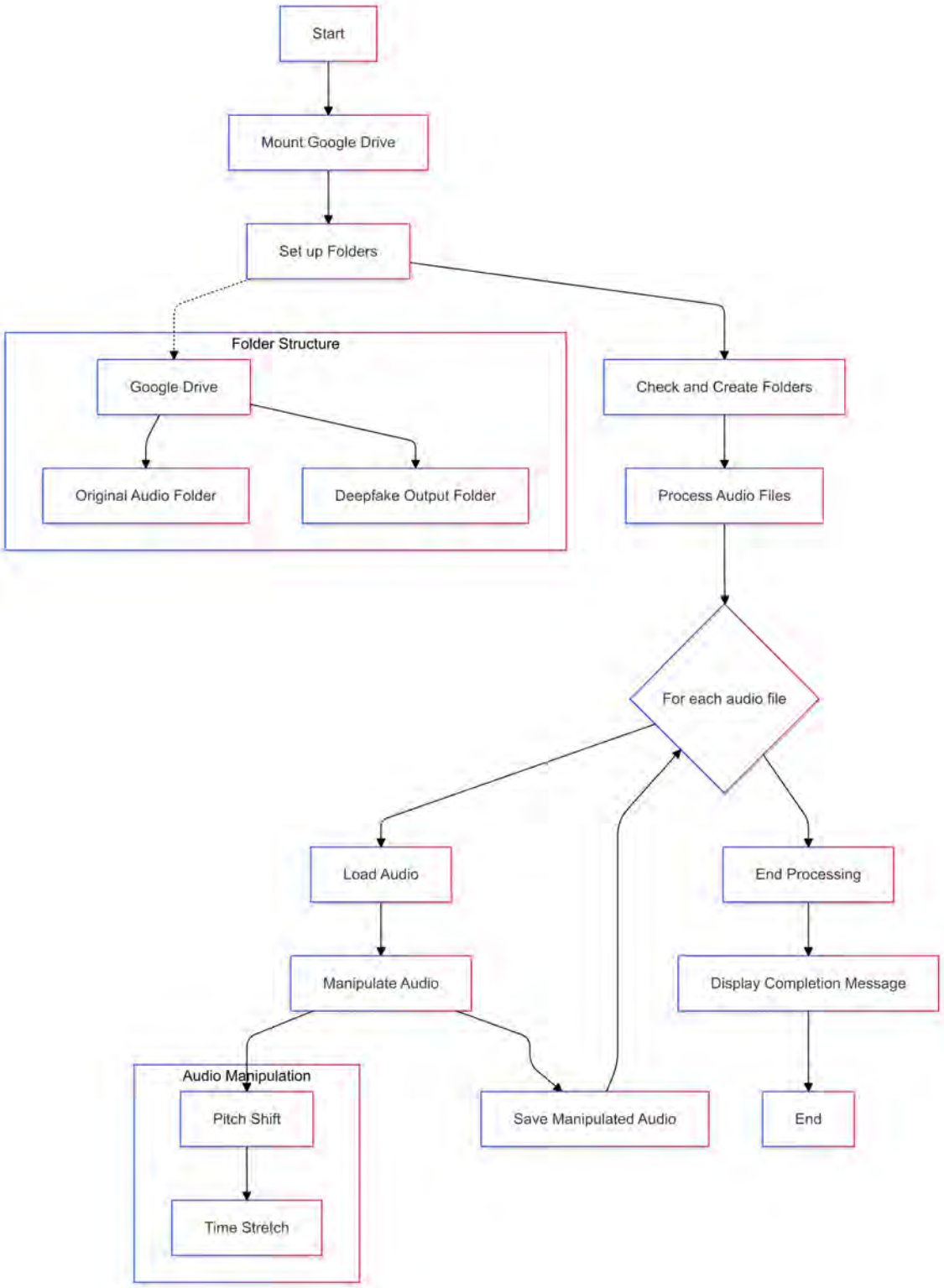

Figure 3.4: deepfake generator

3.3 Feature Extraction

Our dataset consists of audio files in different formats. These data are present in the type of audio files, which deep learning models cannot understand. These audio files are required to be converted into features, interpretable by the proposed deep learning models.One of the most necessary steps in processing audio data for deep learning models are the Feature extractions. It successfully inherits the intrinsic characteristics of an audio signal including frequency content , time variations and harmonics structures. Emphasizing relevant parts of the audio signals also reduces irrelevant noises. For deep learning models the feature extraction stage is essential for the conversion method transforming the unstructured audio data into a more organized format.

In order to support our suggested model for detecting deepfake voices or to differentiate between real and synthetic audio samples we extracted three unique properties from audio files for our study . A python based library for music and audio analysis called`Librosa` was employed. It offers the fundamental components required to develop audio information retrieval systems. We used `librosa` for loading audio files and extracting audio features such as MFCC, STFT, and CQT. We also used `numpy` for the numerical operations and combining the extracted features. Lastly, we used `Matplotlib` for visualizing the extracted features. Here, we dive into the details of these features and the methods for their extraction.

Mel-Frequency Cepstral Coefficients (MFCC)
The use of MFCCs in speech and audio processing and recognition is really common. On a nonlinear mel scale of frequency they show the short term power spectrum of a sound by taking the linear cosine transform of a log power spectrum . MFCCs are designed to model the way the human auditory system perceives sound frequencies. This makes them highly effective for capturing the timbral texture of sounds. MFCCs are effective for identifying unique characteristics in speech and other audio signals.
Extraction process:

- At the beginning, we loaded the audio file using `librosa.load()`.
- Then, we computed the MFCCs using `librosa.feature.mfcc()`, which involves taking the Fourier transform of the audio signal, mapping it onto the mel scale, and then applying the discrete cosine transform. The mel scale is also known as the intuitive scale of sounds where the sound is evenly spaced apart which is considered by the listeners. The most commonly used formula to convert frequency f in hertz to mels m is shown below:

$$\text{Mel}(h) = 2595 \log_{10}\left(1 + \frac{f}{700}\right)$$

- After that, the fixed-sized feature that we took was calculated using the average of MFCCs over time.

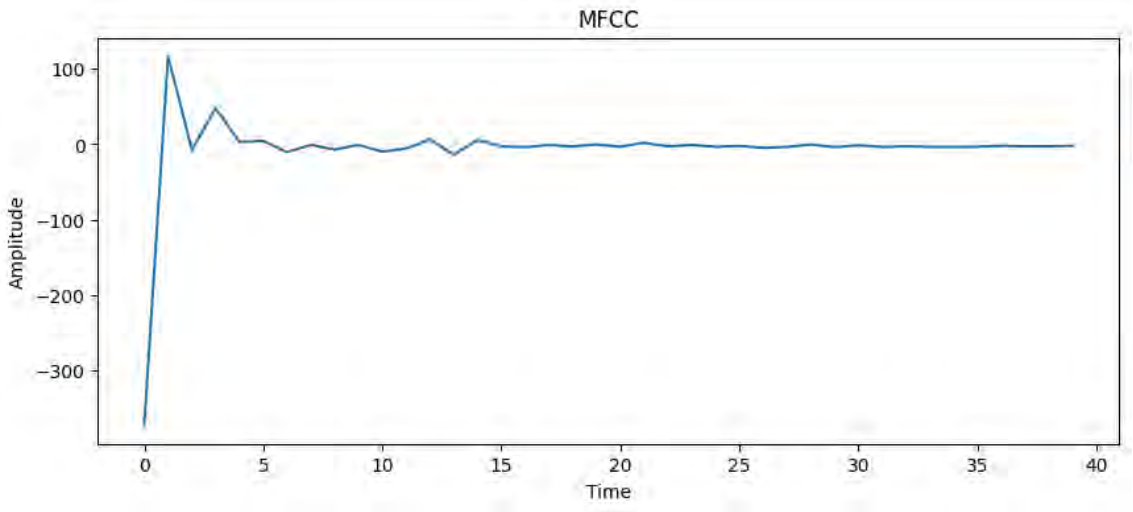

Figure 3.5: MFCC1

Short Time Fourier Transform (STFT):
STFT uses audio signals in the time-frequency domain , dividing the signals into overlapping segments and applying the Fourier transform to each one separately.This makes it easier to monitor the evolution of audio's spectral content over time,which is why it is beneficial for recording dynamic aspects.By analyzing both temporal dynamic and frequency components, it is a valuable tool in detecting subtle changes of manipulated audios.
Extraction Process:

- We loaded the audio file at the very beginning using `librosa.load()`.
- We computed STFT using `librosa.stft()`, involving audio signal segmentation and using the Fourier transform at each segment.
- The magnitude of STFT was taken and we calculated using this magnitude and averaging over time to achieve a fixed-size feature vector.

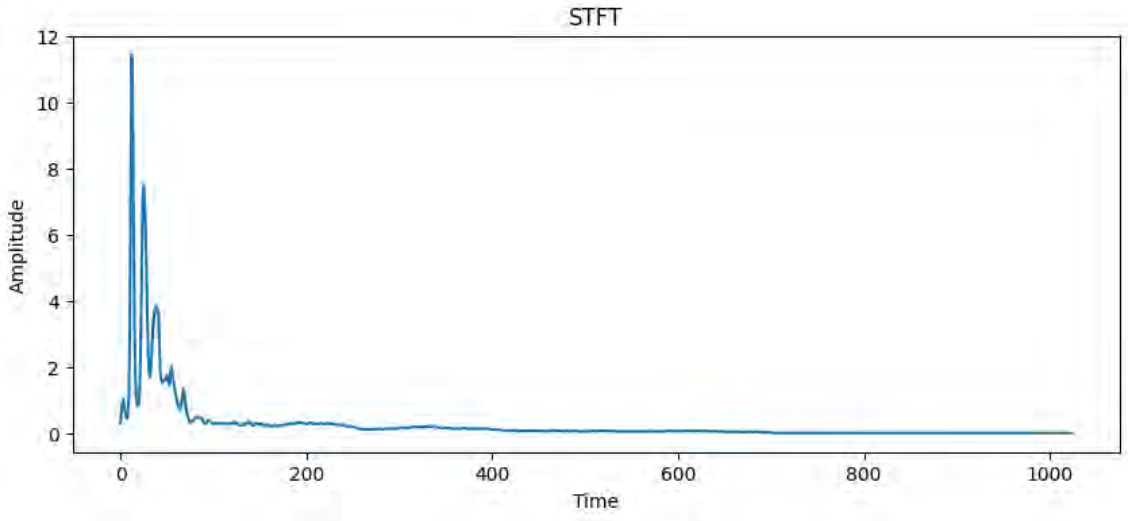

Figure 3.6: STFT1

Constant-Q Transform (CQT):
Similar to STFT ,CQT uses window lengths and a logarithmic frequency scale to create a more realistic depiction of audio information. Time-frequency representation provided to align with the human pitch perceptions. CQT is effective for analyzing complex audio signals by capturing harmonic structure and tonal components . Deep fake techniques in audio signals can be highlighted by CQT.
Extraction Process:

- We loaded the audio file at the very beginning using `librosa.load()`.
- Then, we computed the CQT using `librosa.cqt()`, which involves transforming the audio signal with a logarithmic frequency scale and variable window lengths.
- Later on, we have taken the magnitude of the CQT and average it over time to get a fixed-size feature vector.

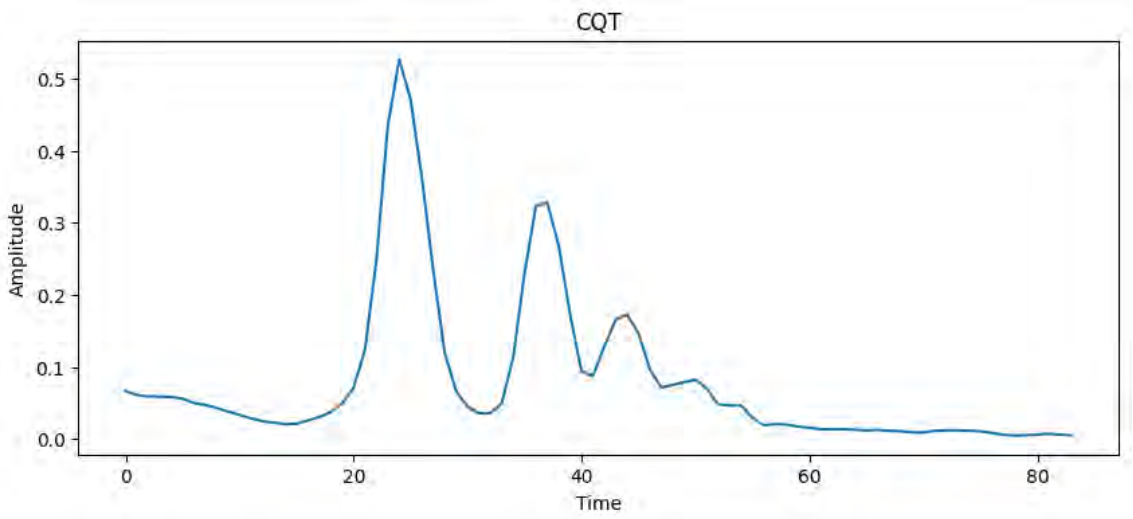

Figure 3.7: CQT1

Combining the three features:
After individually extracting the MFCC, STFT, and CQT features, they are merged into a single feature vector for each audio sample. MFCC captures the timbral texture, STFT represents time-frequency dynamics, and CQT highlights the harmonic structures. This unified feature vector offers a more comprehensive representation of the audio signal, that is more informative than any single feature alone. The features are concatenated horizontally to create one single vector. Combining these features strengthens the detection system, making it recognize various types of manipulations. While a deep fake algorithm might successfully replicate one aspect of the audio signal, it could fail to replicate all features accurately.

Each feature underlines distinct aspects of the audio signal. MFCC focuses on timbral texture and vocal properties, which are key for recognizing subtle traits of human speech that might be altered or missing in synthesized audio. STFT provides insights into how the frequency content shifts over time, helping to identify unnatural transitions that arise from deep fake generators. CQT, on the other hand, offers a deeper look into the harmonic and tonal components, revealing inconsistencies in pitch and tone often present in fake audio. Merging these features enables the model to gain a more complete view, improving its potential to distinguish between real and deepfake audio content. For example, MFCC may capture the overall sound quality, while STFT and CQT can capture frequency and harmonic irregularities.

Machine learning models, especially those using RNN or LSTMs, perform better with input features that are diverse and comprehensive. Integrating MFCC, STFT, and CQT provides the model with rich and varied information, enhancing its capacity to learn and generalize. Extracting and combining MFCC, STFT, and CQT creates a robust feature vector set, significantly improving the effectiveness of deep fake audio detection models. By leveraging strengths of each feature type, the system is better equipped to detect subtle and intricate manipulations, ensuring more accurate identification of fake audio.

3.4 Data Preprocessing

Any deep learning model must need data preparation and the performance and efficiency of LSTM and RNN models are directly affected by it. Effective preprocessing ensures that the data is in a format that is appropriate for effective learning . Generalizing unknown data and more precise predictions are the outcomes of this. In our research, we first set up the paths to the directories for fake and original audio files. Necessary libraries for feature extraction, label encoding, and data splitting were imported. The paths for fake and original audios are defined, and lists to hold extracted features and their corresponding labels were initialized.

The 'createDataset' function iterates over the audio files in a given directory, extracts features using the 'extract features' function, and appends the features and labels to their respective lists. The 'createDataset' function is called for both fake and original audio directories to extract features and prepare the dataset.

For computational efficiency, the feature and label lists are transformed into numpy arrays. Machine learning models need numerical inputs to label both fake and real audios which are done by using "LabelEncoder". Then the encoded labels are transformed into a one-hot encoding format that is helpful for categorical classification tasks in which the model produces probabilities for every class.

The dataset uses train and test then split into training and testing sets. To assess the model's performance,the model is trained on one subset of data and then tested on another. Model can be evaluated using data that hasn't been seen before , giving an indication how well it can generalize the data.

To prepare and organize dataset for training and testing, machine learning model is used for deepfake audio detection, several preprocessing steps are necessary.

3.5 Classifications

Long Short Term Memory (LSTM):
Long Short Term Memory (LSTM): LSTM networks are a advanced kind of recurrent neural network (RNN) that was designed to overcome the drawbacks of traditional RNNs, including the incapacity to efficiently learn then maintain the long term dependencies. RNN has a major flaw which is the vanishing gradient problem; it restricts the ability to spread meaningful information across lengthy sequences . To solve this issue Hochreiter and Schmidhuber presented LSTM in 1997 . The LSTM model is really helpful for activities that involve processing sequential data because of their architecture which enables them to record and use information over longer intervals.

LSTM uses multiple layers that participates in both testing and training dataset. These layers are optimized using various activation functions such as RelU, Tanh, and Softmax. LSTMs are intentionally created to overcome the issue for long-term dependency, inherently recalling information for continuous durations.

Although they are a component of RNN, LSTMs significantly overcome the limitations of traditional RNNs by maintaining information for longer durations.

To understand how LSTMs improve over RNNs, it is essential to have a basic knowledge of feed forward neural networks. In a feed-forward neural network, information travels in a single direction- from the input layer to the output layer through hidden layers. Data moves linearly across the network, never revisiting a node. These networks have limited memory and perform averagely in classification tasks because they only consider the current input without any sense of temporal order. They cannot remember past instructions, making them inadequate for tasks requiring context over time. Recurrent Neural Networks (RNNs) overcome the challenge of processing sequential data by allowing information to flow through loops, enabling the model to consider both current and past inputs when making predictions. This internal memory allows RNNs to retain essential information from previous inputs, enhancing their ability to make accurate predictions. Unlike feed-forward neural networks, RNNs transmit information across multiple time steps, offering a more comprehensive understanding of sequential data.

LSTM networks build on this capability by incorporating gating mechanisms that regulate the flow of information, these gates- the input gate, forget gate, and the output gate- work together to manage which information is stored, updated, or discarded, This selective memory mechanism ensures that LSTMs retain relevant information across long sequences, making them highly effective for tasks such as speech recognition, language modeling, and time series forecasting. The ability to remember or forget information as needed is essential for capturing context in sequential data, such as language or time series. This capability to maintain long-term dependencies makes LSTM particularly well-suited for applications where earlier elements in a sequence play a crucial role in predicting future outcomes. Due to their flexibility and effectiveness, LSTMs have become a foundational tool in sequential data processing, delivering reliable performance across a wide range of fields.

3.6 LSTM Architecture

A Long Short-Term Memory(LSTM) is composed of multiple interconnected LSTM cells, each specifically designed to store and manage information over extended sequences. The internal structure of these cells is sophisticated, featuring three essential gates: the input gate , the forget gate and the output gate. These gates work together to regulate the flow of information, enabling the network to retain crucial details while discarding useless data. The work of these three gates are described below:

1. **Input Gate:**The rate at which new data enters cell state is controlled by input gate .It selects the values from the current input and the previous hidden state that will be used to update the cell state.

   Additionally, it determines which new information belongs in Long-Term memory.

   $$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

To determine which components of the input should be added to cell state, it employs a sigmoid activation function(σ). Before being integrated into cell state, filtered input is subjected to additional processing through tanh activation function m scales the value to a range between -1 and 1.

2. **Forget Gate:** The Forget gate determines which data from the cell state should be erased . This is essential for getting rid of material that isn't pertinent to the situation anymore.

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f)$$

Components of the forget gate: The forget gate accepts the previous hidden state (ht-1) and the current input . It generates values between 0 and 1 , it represents which information in the cell state should be forgotten , using a sigmoid activation function . To remove the extra data these values are then multiplied element wise by the cell state.

3. **Cell State Update:** The LSTM cells memory serves is called cell state , Moving crucial ata between time steps like conveyor belt . The filtered cell state from the forget gate and the scaled input from the input gate are combined to update it.

$$c = \tanh(W_C \cdot [h_{t-1}, x_t] + b_c)$$
$$C_t = f_t * C_{t-1} + i_t * C_t$$

Updating Cell State: The Tanh function used to generate the candidate cell state (C). Input gate scales new candidate state, and forget gate filters the prior cell gate(C_t) to create a new cell state (C_{t-1}). The update makes sure that the relevant and fresh information is retained and new important information is added .

4. **Output Gate:** The output for LSTM cell is controlled by output gate, for the next time step and the network's output it will work as the hidden state.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

Components of the output gate: The output gate processes the prior hidden file state (h_{t-1}) and the current input (x_t). It uses a sigmoid activation function to decide which characteristics of the cell state should be output . The filtered cell state is then scaled by a tanh activation function to produce the new hidden state (h_t) , the LSTM cell's output.

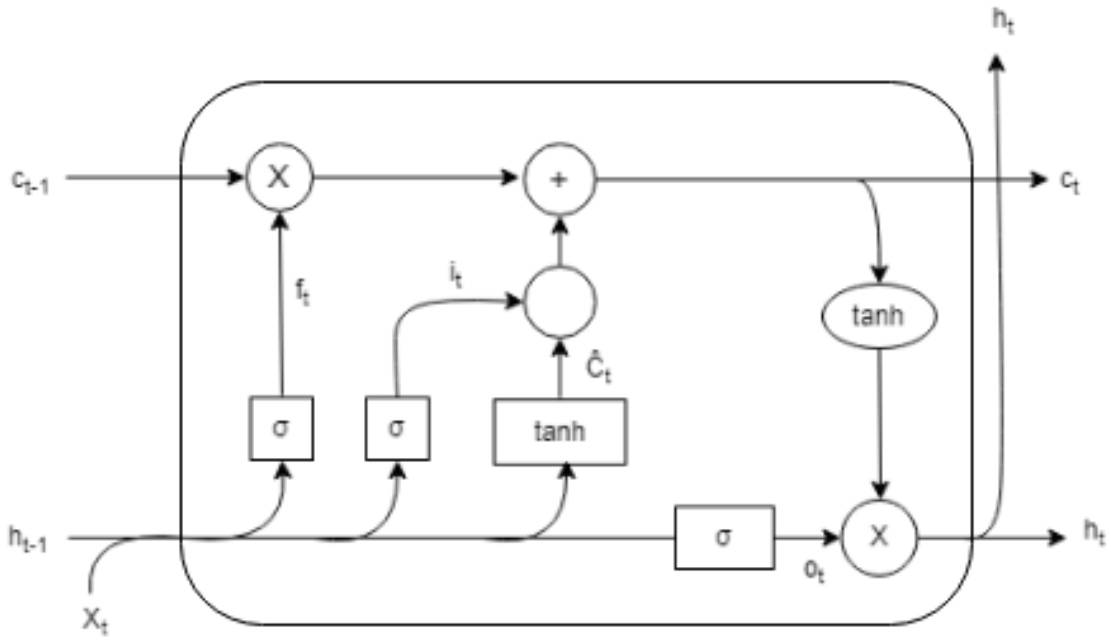

Figure 3.8: LSTM Architecture

3.7 RNN Architecture

Recurrent Neural Networks (RNN):
Recurrent Neural Networks (RNNs) were introduced to address the shortcomings of traditional Feedforward Neural Networks (FFNs). FFNs handle each input independently as it passes multiple hidden layers, without accounting for the order or relationships between inputs. As a result FFNs are not ideal for tasks that require understanding dependencies within sequences. In contrast, RNNs are better suited for applications like language modeling, speech recognition, machine translation, time series forecasting, and other tasks that demand an awareness of context.

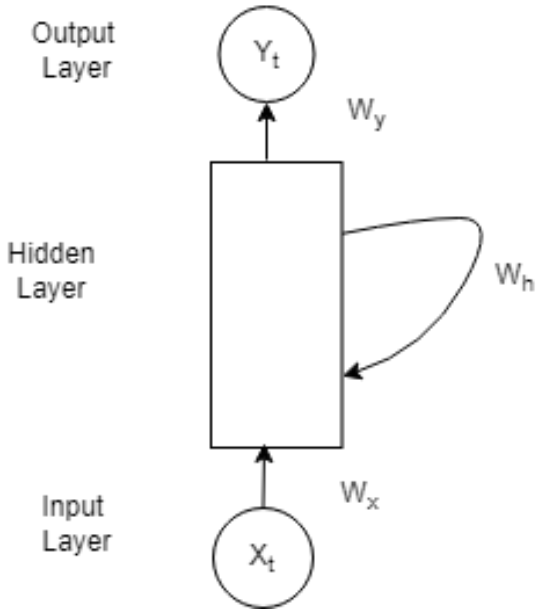

Figure 3.9: RNN Architecture

RNNs incorporate recurrent connections that enable information to flow from one time step to the next. This design allows the network to retain an internal memory, where the output from a given step feeds into the input of the following step. By utilizing past information, RNNs can identify temporal patterns and dependencies, making them highly effective for handling sequences of varying lengths.
Architecture Of RNN
For a more clear understanding of the concept of RNN, let's look at the unfolded RNN diagram.
An RNN processes an input vector **X** by scanning it sequentially from left to right, updating its hidden state and generating an output vector at each time step. The network uses the same set of parameters across all time steps to ensure consistency. These parameters include:

- **U**: Control weights between input layer **x** and the hidden layer **h**.
- **W**: Manage connections between consecutive hidden states.
- **V**: Define weights from hidden layer **h** to the output layer **y**.

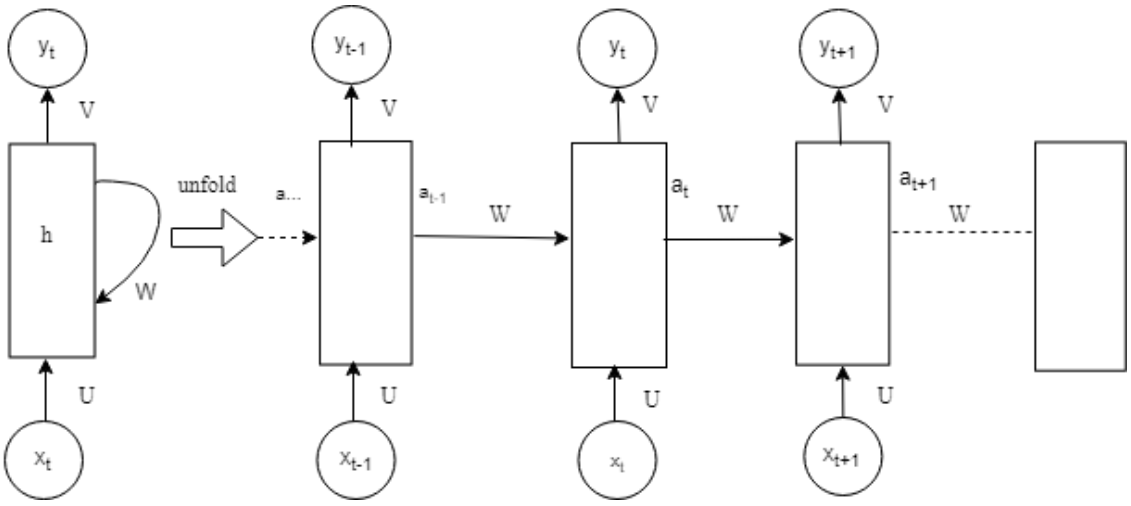

Figure 3.10: RNN Unfolded

This parameter sharing approach helps the RNN efficiently capture temporal dependencies, allowing it to retain information from previous inputs in its current hidden state. At each time step t, the hidden state $\mathbf{a}_t$ is calculated based on the current input $\mathbf{x}_t$, the previous hidden state $\mathbf{a}_{t-1}$, and the model parameters, following the formula:

$$\mathbf{a}_t = f(\mathbf{a}_{t-1}, \mathbf{x}_t; \Theta) \tag{3.1}$$

This equation can also be expressed as:

$$\mathbf{a}_t = f(\mathbf{U}\,\mathbf{x}_t + \mathbf{W}\,\mathbf{a}_{t-1} + \mathbf{b}) \tag{3.2}$$

Here:

- $\mathbf{a}_t$: Output from the hidden layer at time step t.
- $\mathbf{x}_t$: Input at time step t.
- Θ: A collection of trainable parameters, including weights and biases.
- $\boldsymbol{U}$: Weight matrix connecting the input layer to the hidden layer ($U \in \theta$).
- $\boldsymbol{W}$: Weight matrix for recurrent connections within the hidden layer ($W \in \theta$).
- $\boldsymbol{V}$: Weight matrix connecting the hidden layer to the output layer ($V \in \theta$).
- $\mathbf{a}_{t-1}$: Output of the hidden layer from the previous time step $(t-1)$.
- $\mathbf{b}$: The bias vector for the hidden layer.
- f: Activation function applied at each step.

3.8 BiLSTM

Bidirectional Long Short-Term Memory :
BiLSTM is an advanced RNN best for working with sequential data. For both directions forward and backward, it can process information . This enables to grasp information for both past and future in a sequence.

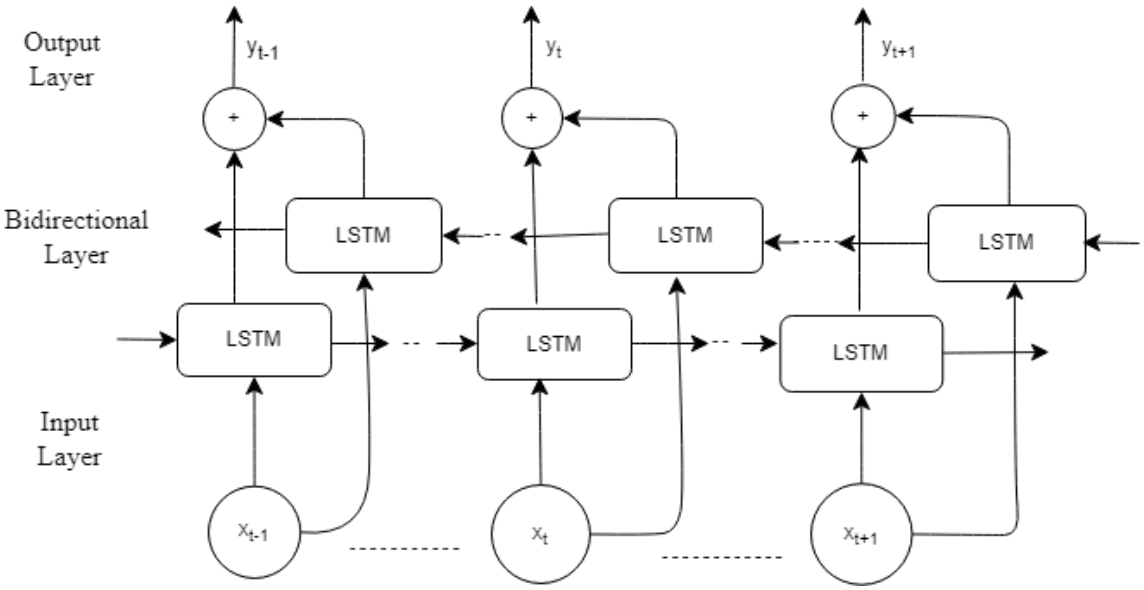

Figure 3.11: BiLSTM Architecture

It is crucial to examine BiLSTM structure and functionality in order to understand how it operates:

1. **LSTM (Long Short-Term Memory)**: LSTM is a form of RNN designed to address limitations of traditional RNN in dealing with long term dependencies. It regulates which information is to be kept or forgotten overtime through memory cells and gating mechanisms. This model can identify patterns and relationships over lengthy time steps because of its internal memory , which makes it appropriate for capturing long range dependencies.

2. **Bidirectional Processing**: BiLSTM functions in both ways in contrast to conventional RNNs which only analyze sequences in one direction. By using two LSTM layers , one which process the sequence from the beginning to the end and the other from end to beginning . The model can record patterns from both sides since each layer has its own memory cells and hidden states.

3. **Forward and Backward Passes**:Throughout forward pass, input sequence undergoes processing step-by-step by forward LSTM, updating hidden state and memory cell according to the existing input and previous output. In parallel, backward pass analyzes sequence in reverse, with backward LSTM updating its state. This dual pass approach enables the model to utilize past and future information effectively.

4. **Combining Outputs**: Once both the forward and backward passes are completed, the hidden states from each layer are merged at every time step, typically through concatenation or another transformation. This combination of forward and backward perspective enables the BiLSTM to capture both preceding and upcoming context, enhancing its ability to model and detect intricate dependencies within the sequence.

3.8.1 Architecture Overview of BiLSTM

Here is a breakdown of the key components of the BiLSTM architecture:
Embedding Layer: The input sequence is transformed into dense vector embeddings that capture the semantic meaning of each data point. The features provide a concise and informative representation that improves the learning process for the following layer.

Bi-LSTMLayer: The important part for this architecture is that it contains 2 LSTM structures; one that analyzes the sequence from start to end, while the other analyzes in the reverse order. Each LSTM layer comes with its own set of parameters, aiding to capture the unique patterns from both directions.

Output Layer: The last output is generated by adding the hidden states from the forward and backward LSTM layers at every time step. This merging can be done by transformation or concatenation methods.Depending on the specific task at hand.

The BiLSTM structure can be extended or modified to meet the needs of a particular task/ in an example. Moreover , Fully connected layers or attention mechanisms can be added on top of the BiLSTM layer to improve performance and expand the models capabilities .

Input Sequence: Input Sequence : The input is composed of a series of data points , such as words in a sentence or elements in data stream . The data points represented as a vector or embedded feature in a fixed-length format for further processing.

Chapter 4

Evaluation and Performence Analysis

4.1 Result Analysis

LSTM: The code that has been used for this research , has been used from the Keras library for building LSTM(Long Short-Term Memory) and for classification of tasks. Outlined below is a detailed breakdown of all the components of the code and their functions :

1. **Data Reshaping:** The input data (`X_train` and `X_test`) is reshaped to match the format used by **LSTM** model. **LSTM** expects input data to be structured in the form of (samples, timesteps,features),so`np.expand_dims` is used to add an extra dimension to the data.

2. **Model Definition:** A Sequential model is defined, which allows you to stack layers linearly.

3. **Adding Layers:**The model includes f two **LSTM** layers, each with 128 units. The first **LSTM** layer is configured to return sequences, meaning it outputs a sequence of 128 values per timestep for each sample in the batch. This setup is essential as the next **LSTM** layer also expects sequences as its input. Dropout layers are placed after each **LSTM** layer to avoid overfitting by randomly setting a fraction (20% in this case) of input units to zero at each update during training time.

4. **Output Layer:** The final layer is a Dense layer with the number of units equal to the number of classes in the target (`y_train`) It uses the 'softmax' activation function, making it ideal for multi-class classification. The softmax function output is a vector that represents the probability distributions of a list of possible outcomes.

5. **Model Compilation:** The model is compiled using the Adam optimizer, categorical cross-entropy loss (appropriate for multi-class classification), and accuracy as the evaluation metric.

6. **Model Training:** The model is trained for 50 epochs with a batch size of 40.The Validation data parameter ensures that its model's performance is evaluated on the test data at the end of each epoch.

The output of the code would be a trained **LSTM** model, along with the history of its performance (loss and accuracy) on both training and test data across the 50 epochs. The model's performance can be visualized by plotting the values in the history object. For example, `history.history['accuracy']` would give the training accuracy for each epoch.The image contains two line graphs, one showing the model's loss over time (epochs), and the other showing the model's accuracy over time.
Model Loss Graph: This graph illustrates how the model's loss changes during the training epochs. The 'Train' line shows a general downward trend, indicating an improvement in model performance on the training dataset as epochs increase. The 'Validation' line also trends downward but has a significant spike around epoch 5 where it peaks sharply above a loss value of 2.0 before decreasing again. This spike may suggest overfitting, where the model begins to memorize the training data and performs poorly on the validation data, or it could be due to an anomaly in the validation data at that point
Model Accuracy Graph: This graph illustrates how the model's accuracy varies over the same epochs. The 'Train' accuracy line fluctuates slightly but generally remains above the 'Validation' accuracy line, showing higher accuracy on the training set than the validation set throughout most epochs. This pattern may suggest that the model is overfitting to the training data, as it performs better on the training data than on the validation data.
These graphs are essential to detect learning problems, like overfitting and underfitting. In this instance, the model displays the signs of effective learning, where loss decreases gradually and accuracy increases for each time period. Although, the oscillations detected shows how inconsistent the learning process is. The uncertainty of some factors may lead to higher learning rate or unpredictability for improving the algorithm. To improve the steadiness for the training process, it could be helpful to repeat the process with numerous hyperparameters or changing the values for learning rate or stop the process early.

RNN: The code used for the research is a Recurrent Neural Network (**RNN**) model using the Keras library for a classification task. Here's a breakdown of what each part of the code does:

1. **Data Reshaping:** The input data (`X_train` and `X_test`) is reshaped to fit the **RNN** model. The **RNN** expects input data in the format of (samples, timesteps, features), so `np.expand_dims` is used to add an extra dimension to the data.

2. **Model Definition:** A Sequential model is defined, which allows you to stack layers linearly.

3. **Adding Layers:** The model consists of two SimpleRNN layers, each with 128 units. The first SimpleRNN layer returns sequences, meaning it outputs a sequence of 128 values per timestep for each sample in the batch. This is necessary because the next SimpleRNN layer also expects sequences as its input. Dropout layers are added after each SimpleRNN layer to prevent overfitting by randomly setting a fraction (20% in this case) of input units to 0 at each update during training time.

4. **Output Layer:** The final layer is a Dense layer with the number of units equal to the number of classes in the target (`y_ train`). It uses the 'softmax' activation function, making it ideal for multi-class classification. The softmax function output is a vector that represents the probability distributions of a list of possible outcomes.

5. **Model Compilation:** The model is compiled using the Adam optimizer, categorical cross-entropy loss (appropriate for multi-class classification), and accuracy as the evaluation metric.

6. **Model Training:** The model is trained for 50 epochs with a batch size of 40.The Validation data parameter ensures that its model's performance is evaluated on the test data at the end of each epoch.

The output of the code would be a trained **RNN** model, along with the history of its performance (loss and accuracy) on the training and test data over the 50 epochs. The model's performance can be visualized by plotting the values in the history object. For example, `history.history['accuracy']` would give the training accuracy for each epoch.
The image contains two line graphs, one showing the model's loss over time (epochs), and the other showing the model's accuracy over time.
Model Loss Graph: This graph shows how the model's loss changes as it learns during the training epochs. The 'Train' line fluctuates significantly, indicating that the model's performance on the training data varies from epoch to epoch. The "Validation" line , which also exhibits some volatility but generally displays a decreasing trend , suggests that the model is getting better over time at using validation data.
Model Accuracy Graph: This graph illustrates the variations in the model's accuracy on the training data is generally improving , though not consistently, as indicated by the upward trending but fluctuating "Train" line . The model's accuracy on the validation data is generally getting better , as indicated by the "validation" line , which displays an overall upward trend with some variations.
In order to detect any learning problems like overfitting or underfitting , these graphs are essential . Since the accuracy increases over time and loss gradually declines , the model in this instance exhibits learning . The oscillations that have been noticed, suggest that the learning process might not be stable . The optimization algorithm's intrinsic randomness or a high learning rate could be the cause of this instability . Using techniques like learning rate schedules or early halting , or experimenting with different hyperparameters ,may help improve the stability of the training process.

4.2 Evaluation Metrics for LSTM Model:

Confusion Matrix The confusion matrix presented below illustrates the overall prediction results for the test set. Confusion Matrix is constructed below:

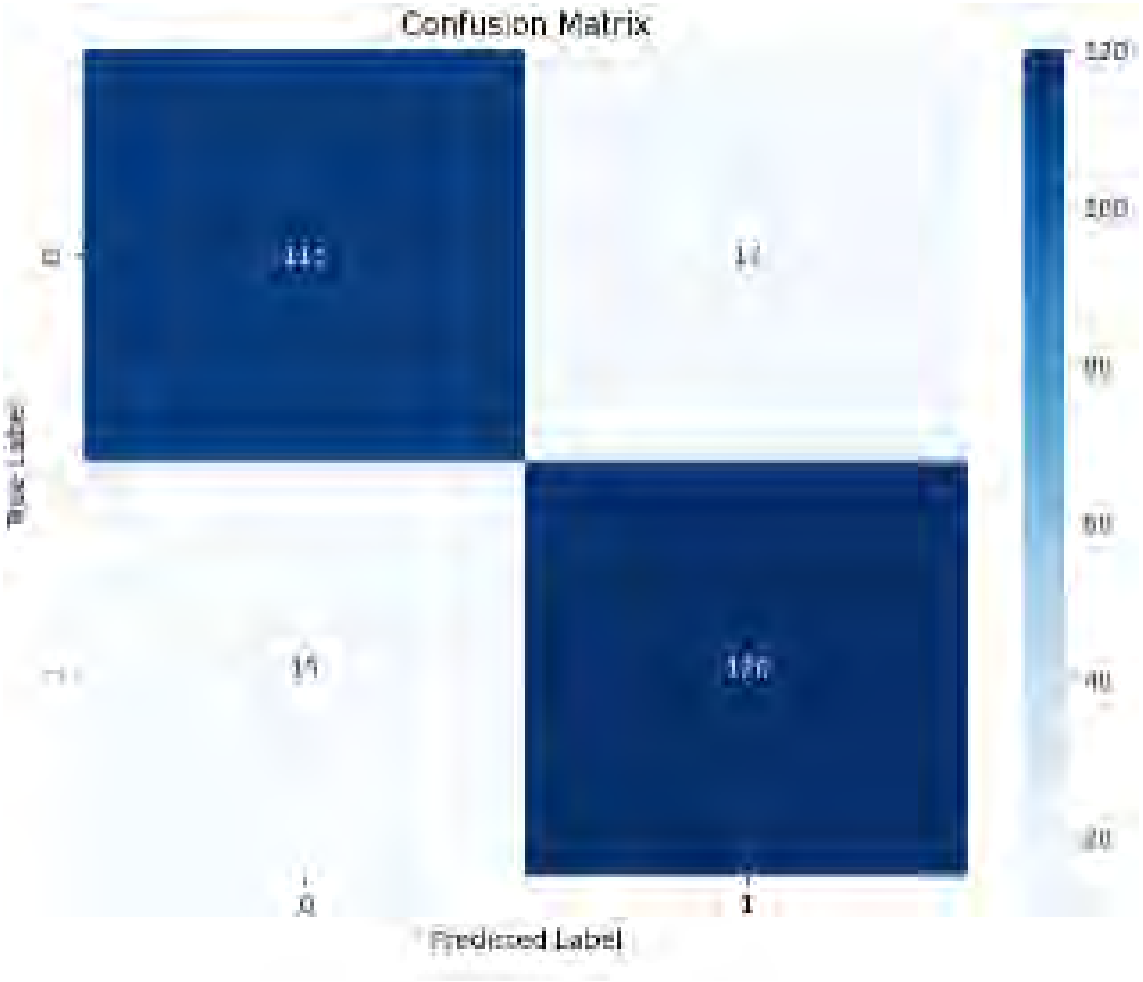

Figure 4.1: LSTM Confusion Matrix

True Positives (TP): 115 Real voices were predicted correctly as real.
True Negatives (TN): 120 Deepfake voices were predicted correctly as fake.
False Positives (FP): 17 Deepfake voices were predicted incorrectly as real.
False Negatives (FN): 15 Real voices were predicted incorrectly as fake.

Classification Report: The classification report offers extensive matrics regarding the success of the classification model, including recall score, precision score and F1-score at every class.

1. **Precision:** The proportion of accurately identifying positive outcomes corresponding to the number of predicted values.

    - Real (Class 0): 0.88
    - Fake (Class 1): 0.88
    - Weighted Average: 0.88

2. **Recall:** he proportion of accurately forecasted favorable observations to all of the class's observations.
    - Real (Class 0): 0.87
    - Fake (Class 1): 0.89
    - Weighted Average: 0.88
3. **F1-score:** Precision and Recall weighted average is counted as F1 score.
    - Real (Class 0): 0.88
    - Fake (Class 1): 0.88
    - Weighted Average: 0.88
4. **Support:** The quantity of instances of the class that actually occur in the given dataset.
    - Real (Class 0): 132
    - Fake (Class 1): 135

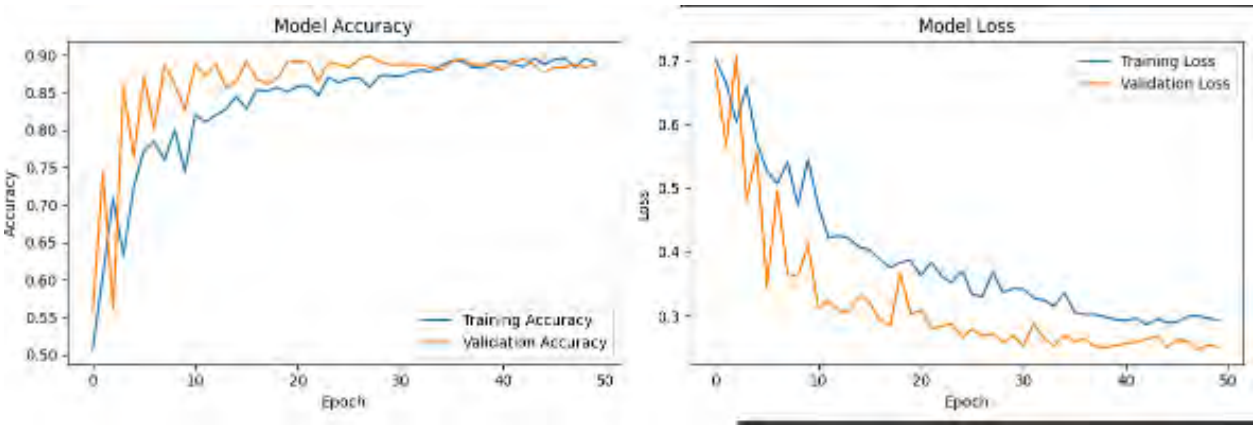

Figure 4.2: LSTM Graph

4.2.1 Overall Model Performance:

- **Accuracy:** Ratio for accurately predicting observations to total observations. LSTM model obtained an accuracy of 0.88, showing that 88% of test samples were classified accurately.
- **Macro Average:** The arithmetic mean of recall score, precision score and F1-score throughout all classes.
    - Precision: 0.88
    - Recall: 0.88
    - F1-score: 0.88
- **Weighed Average:** Average of recall score, precision score and F1-score are adjusted based on the number of correct instances at each class.

- Precision: 0.88
- Recall: 0.88
- F1-score: 0.88

Interpretation: LSTM model shows moderately strong performance in differentiating between real and deepfake voices obtaining an accuracy of 88%. Precision score and recall score specifies that it is moderately good at identifying real voices(precision score and recall score are equal to 0.88) and deepfake voices. (precision score equal to 0.88 and recall score are equal to 0.87) accurately. F1 score reflects moderately fine steadiness within precision score and recall score for both classes. The model's adaptability for detecting deepfake voices is shown at confusion matrix below.Substantially, evaluation matrices above indicates that LSTM model is greatly effective for the classification task, giving an accurate and much more reliable predictions.

4.3 Evaluation Matrix for RNN Model

Confusion Matrix: The confusion matrix presented below illustrates the overall prediction results for the test set. Confusion matrix is constructed below:

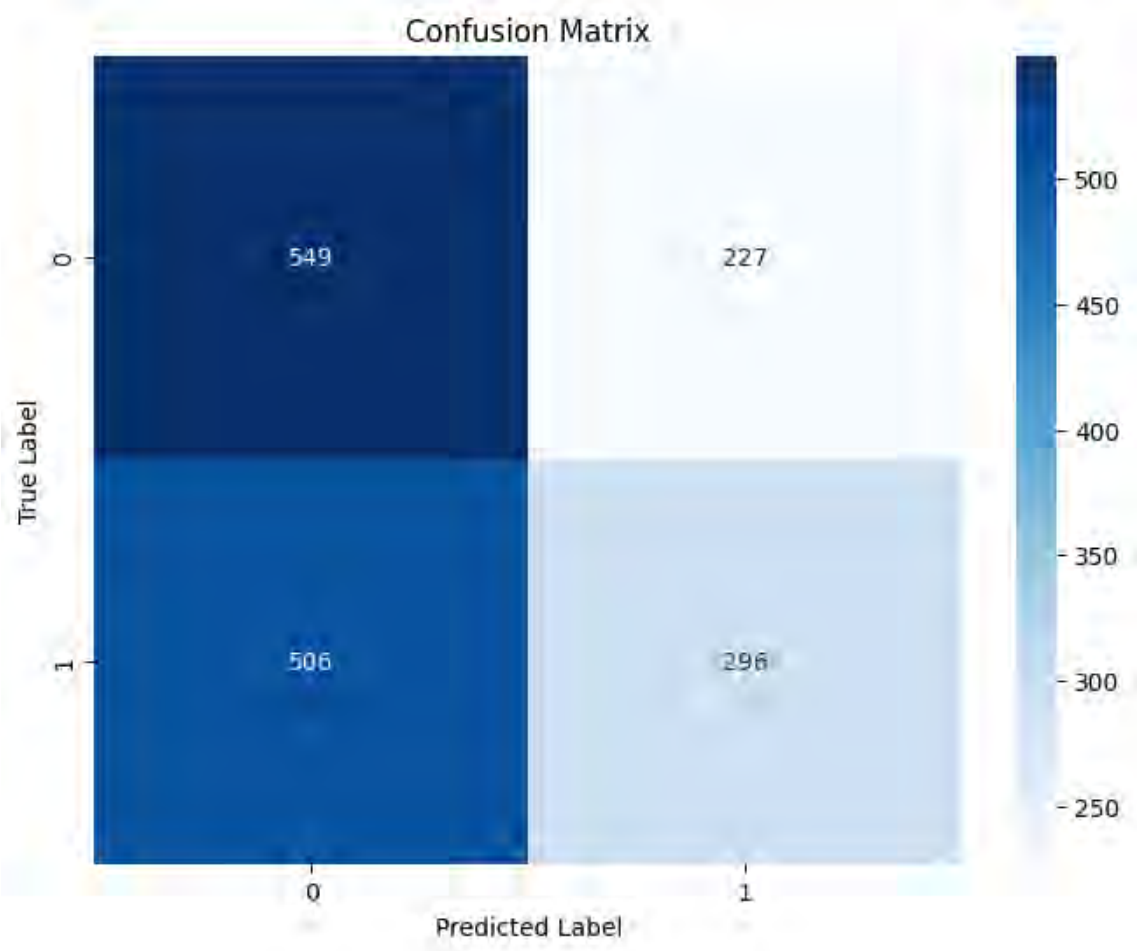

Figure 4.3: RNN Confusion Matrix

True Positives (TP): 549 Real voices were predicted correctly as real.
True Negatives (TN): 296 Deepfake voices were predicted correctly as fake.
False Positives (FP): 227 Deepfake voices were predicted incorrectly as real.
False Negatives (FN): 506 Real voices were predicted incorrectly as fake.

Classification Report: The classification report offers extensive matrics regarding the success of the classification model, including recall score, precision score and F1-score at every class.
1.Precision: The proportion of accurately identifying positive outcomes corresponding to the number of predicted values.

- Real (Class 0): 0.52
- Fake (Class 1): 0.57
- Weighted Average: 0.54

2. Recall The proportion of accurately forecasted favorable observations to all of the class's observations.

- Real (Class 0): 0.71
- Fake (Class 1): 0.37
- Weighted Average: 0.54

3. F1-Score The weighted average of Precision and Recall.

- Real (Class 0): 0.60
- Fake (Class 1): 0.45
- Weighted Average: 0.52

4. Support The quantity of instances of the class that actually occur in the given dataset.

- Real (Class 0): 776
- Fake (Class 1): 802

4.3.1 Overall Model Performance

- **Accuracy:**Accuracy: Ratio for accurately predicting observations to total observations. RNN model obtained an accuracy of 0.54, showing that 54% of test samples were classified accurately.
- **Macro Average:** The arithmetic mean of recall score, precision score and F1-score throughout all classes.
- **Weighted Average:** Average of recall score, precision score and F1-score are adjusted based on the number of correct instances at each class.
- **Precision:**0.54
- **Recall:**0.54
- **F1-score:**0.52

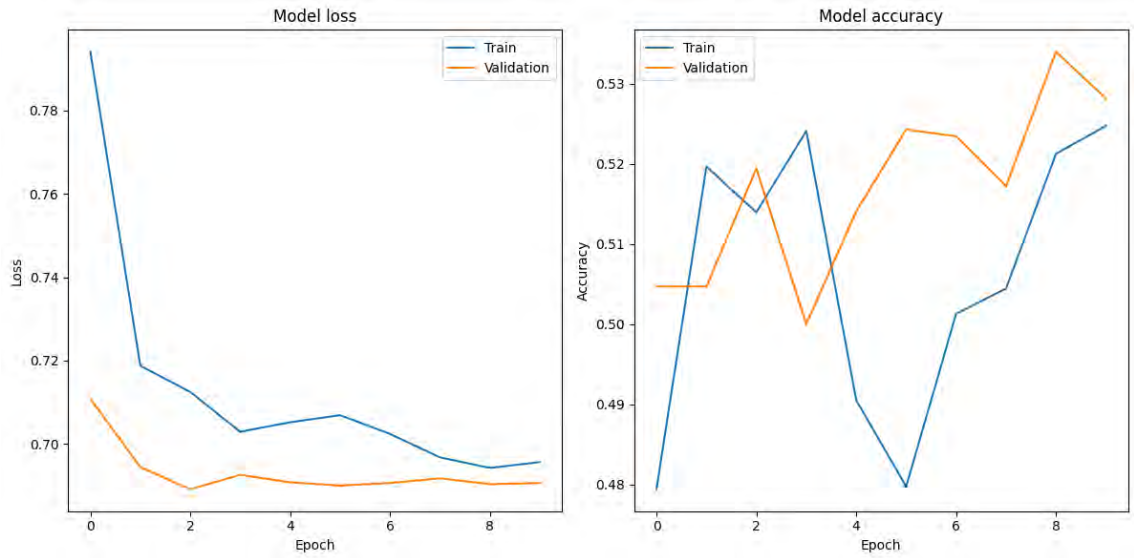

Figure 4.4: RNN Graph

4.4 Evaluation Matrix for BiLSTM Model

Confusion Matrix: The confusion matrix presented below illustrates the overall prediction results for the test set. Confusion matrix is constructed below:
True Positives (TP): 59 Real voices were predicted correctly as real.
True Negatives (TN): 59 Deepfake voices were predicted correctly as fake.
False Positives (FP): 5 Deepfake voices were predicted incorrectly as real.
False Negatives (FN):5 Real voices were predicted incorrectly as fake.

Classification Report: The classification report provides detailed metrics about the performance of the classification model, including precision, recall, F1-score, and support for each class.
1. Precision The proportion of accurately identifying positive outcomes corresponding to the number of predicted values.

- Real (Class 0): 0.921
- Fake (Class 1): 0.921
- Weighted Average :0.921

2. Recall The proportion of accurately forecasted favorable observations to all of the class's observations.

- Real (Class 0) :0.921
- Fake (Class 1): 0.921
- Weighted Average: 0.921

3. F1-score Precision and Recall weighted average is counted as F1 score.

- Real (Class 0): 0.921

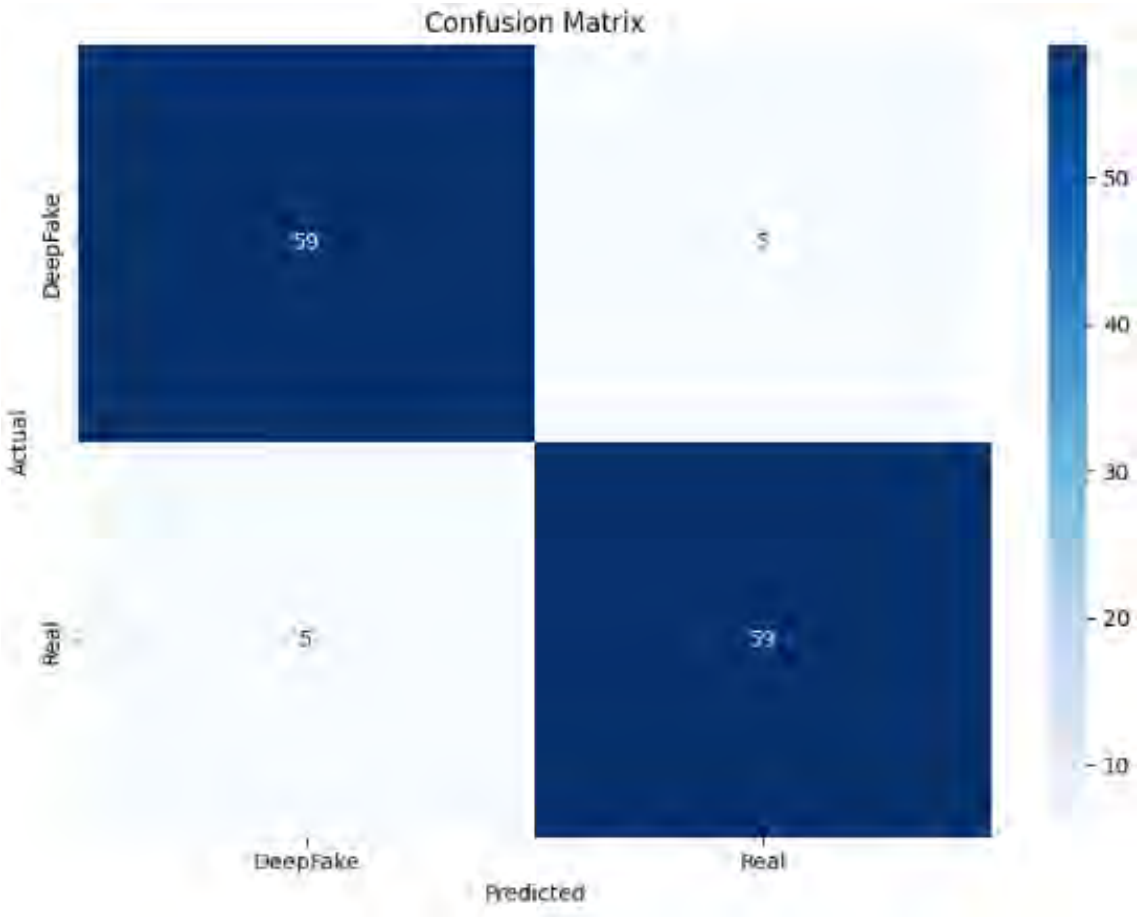

Figure 4.5: BiLSTM Confusion Matrix

- Fake (Class 1): 0.921
- Weighted Average: 0.921

4. Support The quantity of instances of the class that actually occur in the given dataset.

- Real (Class 0): 64
- Fake (Class 1): 64

4.4.1 Overall Model Performance

- **Accuracy:** Ratio for accurately predicting observations to total observations. BiLSTM model obtained an accuracy of 0.921, showing that 92.19% of test samples were classified accurately.
- **Macro Average:** The arithmetic mean of recall score, precision score and F1-score throughout all classes.
- **Weighted Average:** Average of recall score, precision score and F1-score are adjusted based on the number of correct instances at each class.

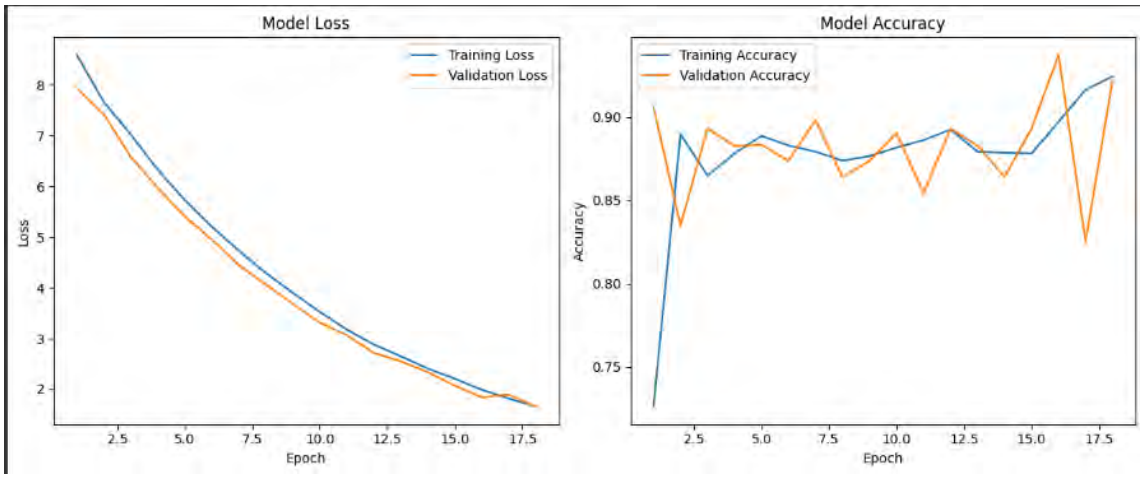

Figure 4.6: BiLSTM Graph

- **Precision:**0.921
- **Recall:**0.921
- **F1-score:**0.921

4.4.2 Evaluation Metrics for a Larger Dataset in BiLSTM

Confusion Matrix: The confusion matrix presented below illustrates the overall prediction results for the test set. Confusion matrix is constructed below:
True Positives (TP): 1338 Real voices were predicted correctly as real.
True Negatives (TN): 181 Deepfake voices were predicted correctly as fake.
False Positives (FP): 3 Deepfake voices were predicted incorrectly as real.
False Negatives (FN): 13 Real voices were predicted incorrectly as fake.

Classification Report: The classification report offers extensive matrics regarding the success of the classification model, including recall score, precision score and F1-score at every class.
1.Precision: The proportion of accurately identifying positive outcomes corresponding to the number of predicted values.

- Real (Class 0): 0.99
- Fake (Class 1): 0.98
- Weighted Average :0.99

2.Recall: The proportion of accurately forecasted favorable observations to all of the class's observations.

- Real (Class 0) :1.00
- Fake (Class 1): 0.93
- Weighted Average: 0.99

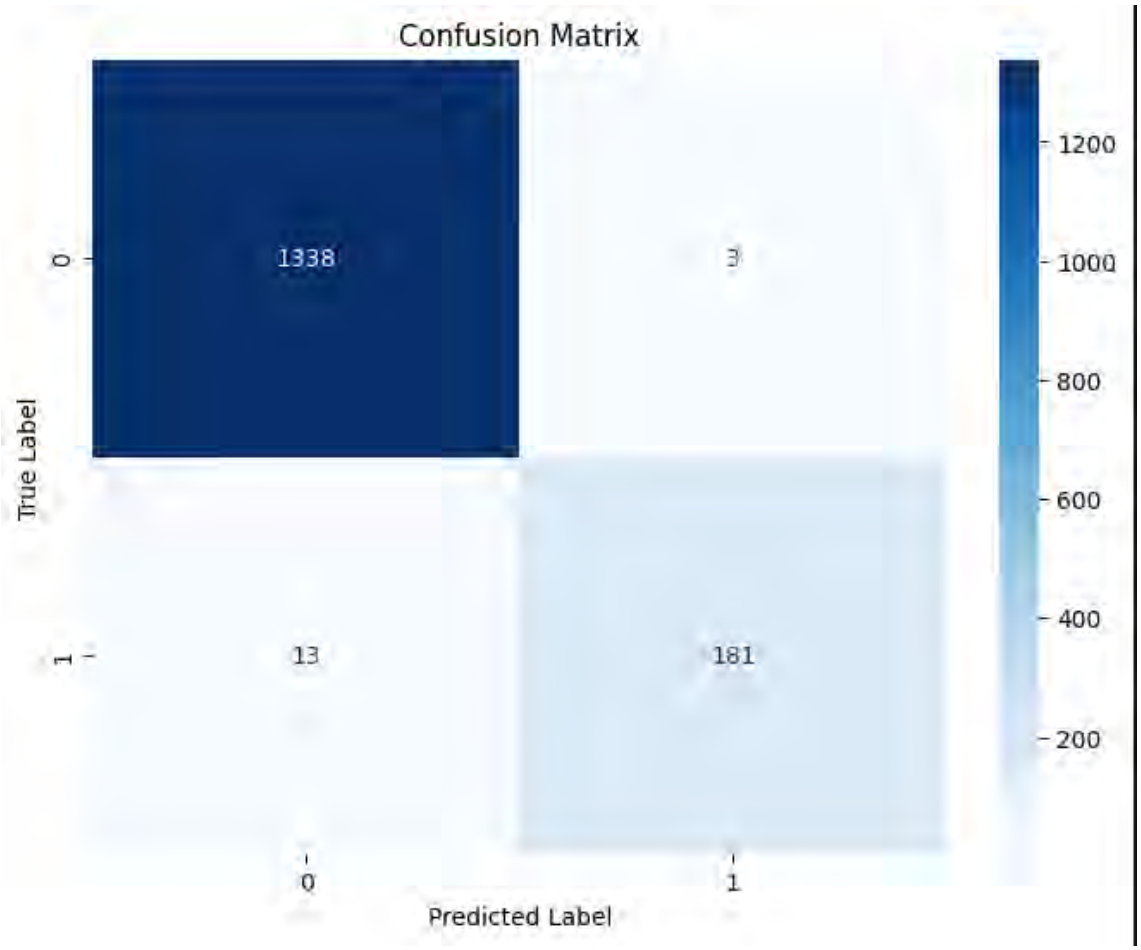

Figure 4.7: BiLSTM Confusion Matrix (2)

3.F1-Score: Precision and Recall weighted average is counted as F1 score.

- Real (Class 0): 0.99
- Fake (Class 1): 0.96
- Weighted Average: 0.99

4. Support The quantity of instances of the class that actually occur in the given dataset.

- Real (Class 0): 1341
- Fake (Class 1): 194

4.4.3 Overall Model Performance

- **Accuracy:** Accuracy: Ratio for accurately predicting observations to total observations. BiLSTM model for larger dataset obtained an accuracy of 0.99, showing that 99% of test samples were classified accurately. We could not run the code for the entire dataset having 14400 audios. We divided it into 10 epoch of batch size 1535 having 1341 real and 194 fake audio. We were able

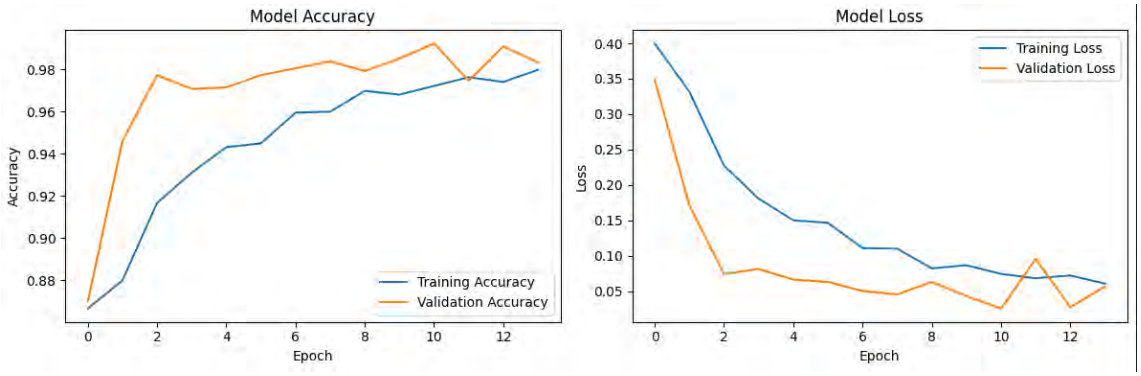

Figure 4.8: BiLSTM Graph (2)

to execute just one epoch. Within that the model was unable to detect 3 deep fake audios. Early stopping for this code was enabled.

- **Macro Average:** The arithmetic mean of recall score, precision score and F1-score throughout all classes.
- **Weighted Average:** Average of recall score, precision score and F1-score are adjusted based on the number of correct instances at each class.

-Precision: 0.99
-Recall: 0.99
-F1-score: 0.99

4.5 Accuracy Comparison

We tried to evaluate our research to related studies in the field of deep fake audio detection. However, we are not trying to show a direct comparison. Because all these papers are using different datasets, classifiers, and techniques.A comparison may be drawn based on accuracy. We only showed the accuracy of our best performing model in this table.

Existing Work	**Classifier**	**Dataset**	**Accuracy**
Ameer Hamza et al [5]	Random Forest, SVM, MLP, Extreme Gradient Boosting	ASVspoof2017 training and test dataset	93%
Dee H. Hakan Killinc et al [13]	CNN, MLP, SVC, Decision Tree	28 different speech data	75-88%
Suk-Young Lim et al [7]	CNN, LSTM, XAI	ASVspoof 2019 Logical Access Dataset	88.29%
Islam Altalahin et al [11]	CNN-LSTM	ASVspoof Challenge 2019 dataset	88%
Abu Qais et al [8]	CNN	The ASVspoof 2017 dataset	80.47%
Basant Kumar et al [6]	CNN with XCeption net	DFDC dataset	90%
Kawshik Barua et al [21]	LSTM based RNN	FoR (Fake or Real) dataset by AptyLab	98.33%
The proposed BiLSTM Model	BiLSTM with MFCC Features	Bengali Speech Dataset by Mozilla Common Voice	99%

Table 4.1: Accuracy Comparison Table

Chapter 5

Conclusion

The increasing presence of deepfake audio presents serious challenges, particularly by eroding trust in digital media platforms. In Bangladesh, the spread of manipulated audio has fueled public skepticism, making it harder to rely on online content. Our research responds to this growing concern by applying machine learning techniques and sophisticated audio analysis methods to differentiate between authentic and fake speech. Using a Bangla speech dataset from Mozilla Common Voice, we trained LSTM, BiLSTM, and RNN models, which analyze key features such as MFCC, CQT, and STFT to capture the nuances of speech patterns.

Our preprocessing pipeline ensures data quality by applying padding, label encoding, scaling, and train-test splitting, making the data suitable for model training. The LSTM model captures dependencies over time, BiLSTM strengthens performance by processing sequences in both directions, and RNN effectively handles sequential inputs. While achieving perfect accuracy is difficult due to the evolving nature of deepfake technologies, our solution aims to improve accuracy and precision in identifying manipulated audio.

By developing models that can detect fake audio, we hope to restore public confidence in digital content. Although our focus has been on Bangla speech, the models and methods used in this research can be applied across other languages and contexts, ensuring the solution remains adaptable. As misinformation and fake content continue to spread, this approach offers a meaningful step toward verifying the authenticity of online content and safeguarding trust in digital platforms, both in Bangladesh and beyond.

Bibliography

[1] H. Dhamyal, A. Ali, I. A. Qazi, and A. A. Raza, "Using self attention dnns to discover phonemic features for audio deep fake detection," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, IEEE, 2021, pp. 1178–1184.

[2] K. Li, X. Lu, M. Akagi, and M. Unoki, "Contributions of jitter and shimmer in the voice for fake audio detection," *IEEE Access*, 2023.

[3] A. Hamza, A. R. R. Javed, F. Iqbal, *et al.*, "Deepfake audio detection via mfcc features using machine learning," *IEEE Access*, vol. 10, pp. 134 018–134 028, 2022.

[4] X. Liu, X. Wang, M. Sahidullah, *et al.*, "Asvspoof 2021: Towards spoofed and deepfake speech detection in the wild," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[5] J. Yi, J. Tao, R. Fu, *et al.*, "Add 2023: The second audio deepfake detection challenge," *arXiv preprint arXiv:2305.13774*, 2023.

[6] J. Frank and L. Schönherr, "Wavefake: A data set to facilitate audio deepfake detection," *arXiv preprint arXiv:2111.02813*, 2021.

[7] M. H. Rahman, M. Graciarena, D. Castan, C. Cobo-Kroenke, M. McLaren, and A. Lawson, "Detecting synthetic speech manipulation in real audio recordings," in *2022 IEEE International Workshop on Information Forensics and Security (WIFS)*, IEEE, 2022, pp. 1–6.

[8] H. H. Kilinc and F. Kaledibi, "Audio deepfake detection by using machine and deep learning," in *2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, IEEE, 2023, pp. 1–5.

[9] S.-Y. Lim, D.-K. Chae, and S.-C. Lee, "Detecting deepfake voice using explainable deep learning techniques," *Applied Sciences*, vol. 12, no. 8, p. 3926, 2022.

[10] I. Altalahin, S. AlZu'bi, A. Alqudah, and A. Mughaid, "Unmasking the truth: A deep learning approach to detecting deepfake audio through mfcc features," in *2023 International Conference on Information Technology (ICIT)*, IEEE, 2023, pp. 511–518.

[11] M. A. Rahman, B. Paul, N. H. Sarker, Z. I. A. Hakim, S. A. Fattah, and M. Saquib, "Syn-att: Synthetic speech attribution via semi-supervised unknown multi-class ensemble of cnns," *arXiv preprint arXiv:2309.08146*, 2023.

[12] Y. Zhao, J. Yi, J. Tao, C. Wang, X. Zhang, and Y. Dong, "Emofake: An initial dataset for emotion fake audio detection," *arXiv preprint arXiv:2211.05363*, 2022.

[13] A. R. Ferreira and C. E. Campelo, "Deepfake audio as a data augmentation technique for training automatic speech to text transcription models," *arXiv preprint arXiv:2309.12802*, 2023.

[14] A. Cohen, I. Rimon, E. Aflalo, and H. H. Permuter, "A study on data augmentation in voice anti-spoofing," *Speech Communication*, vol. 141, pp. 56–67, 2022.

[15] V. Sree Katamneni and A. Rattani, "Mis-avoidd: Modality invariant and specific representation for audio-visual deepfake detection," *arXiv e-prints*, arXiv–2310, 2023.

[16] A. Qais, A. Rastogi, A. Saxena, A. Rana, and D. Sinha, "Deepfake audio detection with neural networks using audio features," in *2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP)*, IEEE, 2022, pp. 1–6.

[17] B. Kumar and S. R. Alraisi, "Deepfakes audio detection techniques using deep convolutional neural network," in *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*, IEEE, vol. 1, 2022, pp. 463–468.

[18] Y. Gao, J. Lian, B. Raj, and R. Singh, "Detection and evaluation of human and machine generated speech in spoofing attacks on automatic speaker verification systems," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2021, pp. 544–551.

[19] K. Barua, A. Rahim, P. S. Parizat, M. A. U. Noor, and M. Jannah, "Voice impersonation detection using lstm based rnn and explainable ai," 2021.

[20] *Mozilla Common Voice*. [Online]. Available: https://commonvoice.mozilla.org/bn/datasets.