Deepfake Speech Recognition: Evaluating Accuracy and Efficiency in Detection Algorithms

by

Md. Istiak Hossain
24341194
Swapnil Saha
21301217
Rahnuma Rued
24241317
S. M. Sazidur Rahman
21201232

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
June 2025

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:

Md. Istiak Hossain 24341194	Swapnil Saha 21301217
Rahnuma Rued 24241317	S. M. Sazidur Rahman 21201232

Approval

The thesis/project titled "Deepfake Speech Recognition: Evaluating Accuracy and Efficiency in Detection Algorithms" submitted by

1. Md. Istiak Hossain (24341194)
2. Swapnil Saha (21301217)
3. Rahnuma Rued (24241317)
4. S. M. Sazidur Rahman (21201232)

Of Spring, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on June 20, 2025.

Examining Committee:

Supervisor:
(Member)

Dr. Muhammad Iqbal Hossain
Associate Professor
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi
Associate Professor and Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

Artificial intelligence (AI), in recent times, is experiencing explosive growth, which comes with improvement and challenges in ethics. In this case, most of the problems raised by deepfake technologies revolve around the issue of changing or producing any sound using sophisticated technology for purposes of mimicking someone's voice and involves people ranging from ordinary citizens to public figures thus creating a danger to security and privacy. This study mainly concentrates on the important task of recognizing deepfake speech and evaluating the performance of selected speech recognition models in terms of their accuracy and efficiency. Current deepfake speech detection methodologies are explained and their strengths and weaknesses are discussed. The focus of this research work is therefore geared towards designing robust countermeasures against harmful use of deep fake voice technology which enhances trust in electronic communication.

Keywords: Voice Cloning, Fake Audio Detection, MFCC, DNS, CNN Model, Scenefake, BiLSTM Model, GRU Model, LSTM Model

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent times, the world has experienced significant changes as far as information and technology is concerned and there is no industry that has been left untouched, this cuts across health, art, communications and even safety. However, aside from the positive aspects of AI, such as more efficiency in work due to automation among other factors, there are also drawbacks that come with it. The greatest of these may be the birth of deepfake which simply refers to the technology that enables computer generated audio or video that is fictitious but realistic. Such innovations enable the production of specific voice recordings that are replicas of actual people giving rise to ethical concerns regarding privacy, security, and credibility of images, audio recordings, and videos done electronically. The greatest vice of deep fake technology is voice deep fake wherein the voice of anyone more so a popular person or a private citizen can be produced without that person's consent and this creates dangers like impersonation, and swindling among others and even the creation of falsehoods. This technology improvement leads to the ill effects whereby reproducing the voice becomes impossible, and this has a big impact on the communication process. Not only does it erode confidence in the use of language, and communication by devices due to security concerns, it also presents a risk to individuals and organizations. Thus, the situation has warranted the need for creating realistic and effective systems for recognizing deepfake audio. This study aims to investigate the performance of several speech recognition models built to tell voices that have been altered. Such models detect fakes or manipulated voices, trained on voice features such as tone, pitch, cadence, and differences thereof. The aim of this research is to investigate the detection of speech deep fake threats, to determine the most appropriate means for achieving this, and to help in implementing new measures that will minimize the impact of speech deep fakes. This would, consequently, help to prevent risks of security breaches associated with, especially, voice-based communication and ensure that such modes of communication will be trusted.

1.1 Problem Statement

The rapid usages of AI-generated deepfake audio is threatening digital trust by enabling the identity theft, fraud and misinformation. Current detection methods involve CNNs, BiLSTMs, LSTMs, and GRUs, which are known to struggle with generalization across languages/accents, computational inefficiency in real-world deployment, explainability gaps, and vulnerability to evolving adversarial techniques.

Although CNNs excel in spatial feature extraction, such as spectrograms, and RNN variants such as LSTM/GRU capture temporal dependencies; no single architecture robustly addresses diverse, noisy, or cross-lingual audio. It examines and improves these models to address critical gaps in:

- Hybrid architecture such as CNN-LSTM for spatial-temporal robustness.
- Lightweight GRUs for efficient, real-world deployment.
- Explainable AI in order to build user trust.
- Adaptive training against advanced generative models, for example GAN.

With the enhancement of reliable and interpretable detection systems, this research seeks to reduce risks due to synthetic voice manipulation and preserve digital communication integrity.

1.2 Research Objectives

- Test and Compare Models: It evaluates that how well different models such as (CNN, BiLSTM, LSTM and GRU) can spot fake voices by analyzing both the sound patterns (like spectrograms) and the flow of humans speeches.
- Build Smarter Systems: It combines the strengths of different models (e.g., CNN, LSTM) which helps us to create systems that work better in different languages, accents and noisy environments.
- Make it Faster and Lighter: It focuses on different type of simpler models like GRU. To ensure, the models could run quickly and efficiently. Also, it works on smartphone devices with limited power consumes.
- Explain How It Works: It added different features that help users to understand how the model makes decisions so that they can trust the results.
- Stay Ahead of Threats: We train models to handle new and tricky fake audio which is created by advanced tools like GANs.

1.3 Ethical Considerations

Deepfake voice technology can be dangerous if it is used in the wrong way. People can use it to harm others by spreading fake news or stealing someone's identity. In this current thesis work, we are using deepfake detection to protect people from such these harmful activities. We made sure not to create or share harmful fake voices. Our goal is to improve online safety rather not in misusing AI. In the future, it's very important for everyone to build such an environment and ensure to use this AI system responsibly so that they are fair, safe and protect everyone's privacy.

1.4 Thesis Structure

The introduction provides the growing power of AI and stresses the potential danger of deepfake audio, to which cloned voices can be used for scams or misinformation. It announces the goal of the study to detect such audio through machine learning techniques like CNN, LSTM, GRU, and BiLSTM while stressing the importance of ethical considerations. Literature review is an in-depth summary of research done on identifying forged voices, illustrating the performance of various models and areas of concern, especially in instances of noise or real environments. Data collection and analysis lay out the utilization of the SceneFake dataset, preprocessing techniques like MFCC and spectrogram extraction, and analysis of salient features in an effort to enhance model training and patterns of voice comprehension. The model architecture section explains how each model was built and trained, e.g., employing attention layers to add weight to important audio segments. The outcome analysis provides a comparison of performances based on metrics and graphics, illustrating BiLSTM's ability to perceive speech flow and CNN's performance speed. The conclusion reflects upon the positive outcomes, finds areas to improve such as noise and multi-language processing, and draws attention to the need for explainable models that can be applied in real life. The references offer all the works that consulted and guided this research.

Chapter 2

Literature Review

2.1 Related Work

A new method of communication is described in [1] with the help of physics, psychology, and technology in a bid to understand what is true and what is not. The study presents a high performance robust CNN-based Voice Activity Detection (VAD) system, designed to analyze audio signals in terms of detecting the speech activity and its intervals and reports performance metrics, such as, 99.13% percent of Accuracy in CENSREC-1-C Dataset and 97.60 percent of TIMIT Dataset, which are astonishing figures even in the presence of noise. Besides this level of tracking accuracy, the systems are indeed capable for real life deployments, a downside in any consequences if situational pressures are high. Moreover, a new combined structure is proposed for Deceptive Speech Detection (DSD) within this thesis, based on Convolutional Neural Networks (CNN) and Multi-Layer Perceptron (MLP). This hybrid approach uses features extracted in a fully automatic manner from the audio spectrograms and also includes hand-pick features like Mel Frequency Cepstral Coefficients (MFCCs) and prosody to improve the performance in detecting deceptive speech. The hybrid system of CNN-MLP performs successfully and presents unweighted accuracies of 63.7% and 62.4%, respectively, in the Real Life Data for Deception Detection (RLDD) and Romanian Deva Criminal Audio Recordings (RODeCAR) database. Such results demonstrate how efficient the model is in real-life environments, particularly in forums where truthful voice attributes can be discernibly altered to present lies. Furthermore, the study stresses the importance of state-of-the-art post processing techniques such as hysteretic thresholding, bilateral extension, and minimum duration filtering, which serve to further polish the results obtained from the VAD systems making them more accurate. While using the example of spoken lie detection the study manages to go beyond the traditional methods of this field which are almost entirely based on the polygraph machine. This technique introduces great possibilities concerning the improvement of forensic approaches aimed at the assessment of truthful vs deceitful speech from the audio alone, which is a radical enhancement in speech technology. All in all, this study has both academic merits and implications for practice, which may, for instance, affect major decisions in the Due to the Importance Of The Activities Involved Environments.

The document features and explains a system called SecureVision, which is aimed at

the recent rise of so-called deepfakes – forged images, videos, and sounds created with the help of computer programs, which are a great threat for cybersecurity because they help to spread fake news, steal personal identities and commit fraudulent activities are mentioned in [2]. With the view of the effects of sophisticated deepfakes, SecureVision employs high-level techniques such as multimodal techniques which enhance the auditory and visual deepfake detection by detecting them in working simultaneously. The system in this perspective does not rely too much on the audio media but rather counters its limitations by the image media, an approach which greatly increases the detection performance of the system. An important component of SecureVision is a particular self-supervised learning approach that allows the network to use weak labeling and therefore reduces the need for a large amount of labeled data. This is an important aspect when it comes to real-world use cases, where the data labeling is limited or inconsistent. In addition, the technology involves big data analytics, allowing the analysis of large volumes of data, which is ideal especially in digital environments that are media-rich and very active. The test results prove that SecureVision is better than the current models available and gives great results in terms of performance with Audio deepfakes detection at 92.34%, and image deepfake detection at 89.35%, while at the same time it is less computationally expensive than other approaches. The architecture of the model employs aSpectRNet for the audio detection task and integrates its acoustic features including Mel-frequency cepstral coefficients (MFCCs) and Whisper features to detect images leveraging Vision Transformer (ViT) models with have outperformed CNN quite significantly.In spite of its sophistication, the research paper does reveal some areas of concern including the requirement of less homogeneous datasets for training non-failing detection mechanisms, the need to enhance the accurateness level in a real-time manner in order to reduce latency without reducing the detection rates and the relevance of incessant updates of new deepfake strategies. SecureVision welcomes new challenges in deepfake detection by combining novel self-supervised learning with limb detection and shows promising results in scale and adaptation to new threats. In a growing competitive environment in the cyberspace industry having MEI's accuracy level and orienting towards cost-efficiency, SecureVision proves to be an indispensable utility. Future improvements focus on full-body deepfake detection for purposes of mastering new advanced technologies, and solving wider problems of cybersecurity in order to enhance its relevance on the safeguarding of digital landscapes in the media-sensitive society.

The authors examine the capacity of human beings to detect deepfake speech and consider factors that affect this capacity as described in [3]. In two experimental studies contributing to the research, the focus is on whether existing awareness about deepfakes can assist in accurate recognition and the degree of difficulty in detecting deepfakes with regard to the quality of the recordings. In the first experiment, the participants carried out an informal activity called 'Two Truths One Lie' in which they were fed deepfake audio without their knowledge. It is interesting to note that only 3.20% of the participants spotted the deepfake in the course of the interaction; nonetheless, an experimenter's contrary statement regarding presentation of deepfake technology was given at the end of the experiment, and 83.9% of subjects managed to identify altered speech. This example illustrates the problem

that the deepfake dialect synthesizer researched in the paper creates; this type of recording made it nearly impossible to tell the difference between recordings even when individuals knew that the audios had been tortured using other means. The second experiment analyzed how well subjects could recognize the effect of deepfake on a voice depending on the quality of the deepfake speech, and it was revealed that good quality speech was easier to hide. Headphone-wearing participants registered a 91.5% accuracy in detection whilst those using speakers registered an 80.7% accuracy which showed the role of embedded sound systems in deepfake detection. In addition, age-gender differences were also found to have an effect on the recognition accuracy, with males recording for example a 93.9% detection rate as opposed to only 77.2% for females, and where native Czechs were better than Slovaks at spotting a deepfake irrespective of the language of the audio. Women were suggested to focus upon women as those aware of the technology showed a better detection rate of 91.8% than those who were not aware at 67%. To summarize, the present research discusses in detail the increasing difficulty that experts in the field will face in the near future in deepfake detection. This is mainly due to advancements in generated sounding speeches which still poses a major threat on cyber security. It suggests that there should be more education campaigns to teach the public about deepfakes and algorithms, intensifying the call to avert the risks that deep fake technology poses in social and mass communication. The findings call upon the researchers to upgrade and develop better systems in addressing the challenges posed by peoples' weaknesses against these types of attacks.

A fresh way of detecting malicious audio, which is known as deep fake audio in the applications of deep learning, by understanding the natural speech pause patterns which bear the effects of biological activities like breathing, swallowing as well as thinking is mentioned in [4]. Studying the profound impacts of imitative voices, misinformation, and voice alteration in recent years through technology development has led to the idea of using speech properties for distinguishing human sounds from computerized reproduced ones. Experiments were performed with 49 subjects from which 49 real and more than fifty deep fake voice samples were created. Since speech pausing is usually an ignored element in communication, the research found that there was a keen difference in the presence of speech pauses in cloned speeches as compared to the original ones. This is because of the lack of some biological markers that accompany natural pseudo-human interactions making it easy to classify deepfakes using their syringes' unique pause structure. Five machine learning models were developed and tested for the purpose of speech analysis with respect to deepfake detection whereby the AdaBoost algorithm was the most efficient model which managed to give 79% accuracy in recognizing deepfake voices without having been trained on such a database. The results indicate that it is feasible for biological components of speech to carry out the job of deep fake voice detection systems and it is likely to work better with stability than the current systems which are based on sound that is without object for a short period of time. Taken as a whole, the research underlines the necessity of constituent biological elements in articulations as vital measurements for the detection of deepfakes and recommends a more inherently focused approach in fighting the advancements in audio fakes in digital communication. This approach complements the deepfake detection strategies

available today, as it also suggests a different line of investigation into the biological characteristics of speech as a way of ensuring voice data integrity. The very dangers of existing detection technologies and their limitations are the reasons for the very proposal of the method, hence the researchers make a big impact in audio forensics and combating the vice of deepfake technologies.

The identification of audio deepfakes generated using artificial intelligence, which is an ever-growing problem that affects the security, creates misinformation and sullies the media was highlighted in [5]. While the existing methods of detection have not done badly, they tend to be underwhelming when it comes to real-life applications due to issues associated with a lack of generalizability and lack of enough explanation which diminishes trust among lay persons. The authors present a conceptual structure that aims to improve the interpretability of transformer-based models. This structure is focused on attention roll-out and occlusion techniques that are used to explain the decisions made by the model. The research presents an innovative way of assessing the robustness that goes beyond the transition and includes performance on databases such as ASVspoof and FakeAVCeleb that the models have never seen. The findings show that for instance, Wav2Vec and AST transformer models have outperformed the common model such as gradient boosted decision trees with remarkable performance on the clean and unseen datasets of over 99% success rates. Despite their competent abilities in performance, these models fail when it comes to providing any level of explanation to an audience without supportive technical information. Finally, it also discusses how well transformers work with challenging real-world audio problems like sound that has gone through compression or been recaptured, illustrating the limitations of the current approach and the need for improvements in the technology to allow it to accommodate out-of-domain data. The authors argue that with regards to the methods for detecting audio deepfakes suspicion will never be the greater tendency. It is necessary to make them not only effective but also understandable to such users, which means that it is possible to promote irritated levels of trust in systems that rely on AI technologies by making their inner workings clear. This work is of contribution in terms of deepfakes detection as it covers the weaknesses of both strength and explainability and hence calls for extensive dataset expansion and better usable explanations. On the other hand, the results also indicate that research development already bridged the gap over detection performance, but there is still a need to perform research on a reliable detection method that is easy to understand and practical for use. The steeper the attention towards address explainability versus model performance with the two on aspects of the technology and the end user's purpose of detecting audio deepfakes will combat threatening elements of audio fakes deepfakes.

The growth of deep learning methods usage, more specifically in signal processing related to audio and in speaker diarization, which are very necessary for identifying deepfake voice over text in a group of people was mentioned in [6]. Noise reduction has also been studied using various methods including Spectral Subtraction, wavelet transform, and different types of filters like Gaussian filter, Butterworth filter and Chebyshev filters. Previous studies have also incorporated NLP where a conversa-

tion is recorded, x u convert audio into text and diarization applies both supervised text classification and unsupervised text clustering techniques. In addition, some works showed that the combination of CNN and RNN can be adopted in detection of synthetic speech, where CNN performed well in detailed extraction from the spectrogram while RNN worked efficiently in time sequences of the input. However, there remain issues where multiple speakers need to be accurately diarized and also manage unstructured speech in noisy contexts, which is ideal multiparty conversation referred to as the cocktail party problem. In a nutshell, advanced techniques such as noise suppression, speaker separation, and CNN/RNN deep learning frameworks have been applied in audio signal processing and the fight against deepfake. Together with the above, there has been research targeted on text-based speaker recognition using speech subtitling techniques. Nonetheless, there are still problems that address how to deal with overlapping speech, real-life conditions, and speaker separation within an actively noisy environment.

The prevalence of deepfake audios which uses advanced conversion and synthesis of voices for the purpose of bypassing automatic speaker verification systems was discussed in [7]. Spoofing detection systems based on prior systems easy to use to intercepted audio cannot defend against novel extreme deepfakes. To remedy this, they develop a system which integrates large margin cosine loss (LMCL) and frequency masking augmentation (FreqAugment) for training of robust feature embeddings in order to enhance the detection of known and unknown spoofing attacks. Based on the performance evaluation on the ASVspoof 2019 dataset, the system was able to lower the equal error rate (EER) from 4.04% to 1.26% which demonstrated better generalization. The system was also tested in the presence and in the call center conditions, and it performed well in all the different climates. The need to create detection systems that respond to the changes in the deepfake threat levels as more systems rely on voice authentication is emphasized in the paper.

Deep Neural Networks (DNNs) have been successfully applied in areas of Artificial Intelligence such as computer vision, speech recognition as well as robotics where plausible accuracy is achieved but at the expense of intensive computational power is mentioned in [8]. When it comes to DNNs, which enables their applications into the devices with constraints in resources is the need to enhance their processing potentials. The article critically examines various measures aimed at increasing performance and lowering the power consumption, by which both the architecture (GPUs for instance) and the DNN models are optimized. The issue of new hardware, memory and parallel processing advancement is addressed on how it contributes to improved turnaround time and reduced energy consumption while performing DNNs without degrading the performance metrics. Simply put, the objective is to design DNNs that are easy to use and cost effective to allow for a wide range of applications in their daily operations.

This paper focuses on the challenges that deepfake audio detection faces. This is particularly resolved given the fact that Text-to-Speech (TTS) and Voice Conver-

sion (VC) technologies have become more sophisticated and can create high quality and realistic voices albeit without any emotional variations in [9]. To mitigate this challenge, the scientists constructed a model that incorporates Speech Emotion Recognition (SER). The model transforms an audio signal into a spectrogram via a 3D Convolutional Neural Network (3D CNN) in order to identify and predict emotions as seen in an emotional feature vector. A Random Forest classifier and a Synthetic Voice Detector analyzes these vectors to evaluate whether the voice can be considered real, regardless of the features based on the emotions that are present in that voice – or quite the opposite – if it is a synthesized voice, disregarding the emotional features. The method was applied successfully to several datasets in addition to the above-mentioned, such as ASVspoof, LibriSpeech, LJ Speech, Cloud 2019, and IEMOCAP, and outperformed other more traditional approaches such as VGGish and RawNet2. And to make the performance better in a real workplace, where background sounds are inevitable, the authors applied so-called data augmentation, which is adding some noises in the training data. This was important for making the presented system endure real environments. While the focus on emotions is a limiting factor in the system to many other applications, it enables the system to address the very problem of deepfake audio detection in both silent and noisy settings quite conclusively.

The research paper is based on various types of deepfakes and methods to identify them. In particular, it explains how deepfake technology has been made readily available by brainy systems called GANs, which have made it harder to tell which one is true and which is false. The voice related technology is one of the subsections considered in the research, focusing on how GANs help develop realistic looking synthetic voices as a means of deepfake. This is all well and good because the same technology can be used to design voice assistants or digital therapists and teachers for different users. However, it gets a little more worrying since there are major ethical concerns surrounding the same. For instance, misinformation, and identity fraud among other privacy issues where people can sound like someone else or broadcast false information using people's voices that do not belong to them. To try and mitigate such dangers several measures have been put in place. One of them is voice-face matching where it is determined if the voice and the appearance of the individual in the video are from the same person. The other technique is called Audio-Visual Deepfake Detection (PVASS-MDD) where it uses sound-visual components within the deepfakes and finds out where discrepancies occur. In order to feed these detectors, a library consisting predominantly of audio frauds entitled WaveFake with more than 100.000 of fake audiocassettes was created. Yet, with regard to a certain regional dialect, the difficulty of detecting deepfakes increases especially because of the variations in speech patterns. While achievements in research have been attained, more robust deepfake detection systems are required to counter the more sophisticated modes of deepfake presentation. It's highlighted in [10].

The paper focuses on the emerging issue of identifying altering voice technology termed deep fake. This emerging technology is relatively easy to create and abuse

for reasons such as swindling and spreading false information in [11]. The authors investigated the effectiveness of their approach on the datasets drawn from the fictional audio samples, Fake-or-Real (FoR), consisting of 195000 real and fake audio clusters that have been partitioned into four groups: for-original, for-norm, for-2sec, and for-re rec. Different machine learning algorithms were tried such as Support Vector Machine (SVM), Random Forest (RF), Multi-Layer Perceptron (MLP), Extreme Gradient Boosting (XGB), in addition to introducing a deep learning algorithm VGG-16. The best results were achieved with the SVM, which yielded up to 98.83% accuracy on the for-relrec dataset and 97.57% accuracy on the for-2sec dataset. The best results for the datasets containing original samples were achieved with model VGG-16, which reached 93% accuracy. In order to aid these models in differentiating between genuine and synthetic speech, key audio characteristics were extracted using Mel-Frequency Cepstral Coefficients (MFCC). Even though the models worked efficiently, the major problem faced was detecting fake audio out of its original content which had a lot of noise or distortion reducing the accuracy by a small margin. Overall, the study demonstrates that techniques of machine learning, especially SVM and VGG-16, can be employed in detecting deepfake audio, however, there is still much to do dealing with noisy data.

The problem of detecting deepfake audio, which has become more difficult to detect due to the voice cloning technologies that are now in use. Deepfake audio can be harmful, because it can be used for criminal activities such as impersonation scams and identity theft making it difficult for forensic experts to tell whether the audio was altered. To do this, the authors turned to the Baidu Silicon Valley artificial intelligence laboratory with its dataset that contains original and imitation recordings in which the original 80% was for training and 20% was for validation. They experimented with a number of deep learning models VGG-16, ResNet, FG-LCNN and one Designed by Malik et al and others. These models used key Lossless coding and data reconnaissance features such as Mel Frequency Cepstral Coefficients (MFCC), Mel-spectrograms, Chromagram and Spectrograms which were later converted into images for analysis. Adam, SGD, and Adadelta optimizers were implemented to increase the learning rates and overall performance of the models. The Custom Architecture model exhibited the perfect performance overall with success rate of about 83.63% for chromagram audio while VGG-16 was the highly effective model for MFCC audio achieving accuracy of 86.906%. Nonetheless, this brings us again a major problem: the possible application of such algorithms on unclean or incomprehensible audio, which is a common trait of a majority of real-life situations. Further, as the scenarios created by the deepfakes get sophisticated, models and datasets that are more sophisticated will be required. Nonetheless, this research indicates that in combination with audio features current state-of-the-art models such as VGG-16 and Custom Architecture can provide substantial enhancement in deepfake audio detection, but not sufficient to the level of complexity existing in real situations. It's dealt with in the paper of [12].

The audio deepfake detection problem where deceiving audio is generated through artificial intelligence that mimics the sound of a natural voice was focused on [13].

Such technology could be regarded as amazing but it has potential drawbacks especially in incidences such as impersonation and therefore secure ways of identifying deep fakes should be taught. In order to resolve this issue, the researchers apply machine learning models with different data sets. The MAJ, ASVspoof, ADD, and Wave fake inclusively have helped in providing the authentic and synthetic audio files of the training models. In this two-dimensional research experiment, the following machine learning models were utilized: Support Vector Machine (SVM), Gaussian Mixture Models (GMM), Convolutional Neural Networks (CNNs) and their advanced fellows, like ResNet and SENet. In all these models, important components such as short, long, and prosodic (pitch and rhythm) features including the deep features from the audio were used. As for the outcomes, GMM and SVM models were effective in several challenges where they are commonly found in competitions but compared to the power based on the pattern systems of audio these two dimensional models did not work out as satisfactorily as the deep learning based systems. Nevertheless, the main problem still lies in how such models cope with real audio recordings, which usually entail noise, distortions or other ambient sounds. While these models succeed in theory, they often fail in 'the real world', such as a clear phone conversation or a noise-filled street while recording. All in all, individually audio deepfake detection has improved greatly over the years but to effectively deal with how deepfake technology is changing day by day, advanced dataset and models have to be employed.

The problem of protecting audio data from forgery within the growing circle of dangerous AI-related technologies such as deepfake was solved in [14]. A dataset of audio files classified into two groups: truthful audio signals assigned the label of 1 is used. Mel Frequency Cepstral Coefficients, which understand a few properties of audio signals such as the spectral characteristics, is the main feature Extraction technique. It turned out that the author is going with the Random Forest classifier for detection rather than other kinds which has proven to be one of the best workhorse classifiers. The implementation employs the Librosa library for audio operations and scikit-learn for effecting the Random Forest algorithm making it easy for rigorous training and testing of the model. Cross-validation techniques as a way of measuring the degree of overfitting have been made part of the implementation and in doing this, several performance measures such as accuracy, precision, recall, and F1-score as well as area under the ROC curve (AUC) have been applied. The results returned indicate the utmost need for well-looking deepfake audio detection systems for the development of most of the methodologies in the future. Nevertheless, there are frustrations since existing techniques are too limited in scope to address every possible instance of the misuse and deep fake technologies broadly continue to evolve with time. Future studies will focus more on devising advanced means of extracting features together with other methods that will include combination of both spatial and temporal data. Figure out some other more powerful means of deep learning systems which will include Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) respectively for better detection purposes.

A detailed examination of deep fake audio data and the various associated threats

with special emphasis on protective measures in the areas of media forensics and authentication systems was presented in [15]. For this purpose a wide variety of data containing original recordings and manipulated audio recordings such as deepfakes and clones is adopted. Hence this dataset incorporates various aspects so as to teach the model the differences between the real and fake sound recordings. The method makes use of signal processing functions that include the use of Mel Spectrograms to turn sound signals into images for properly displaying the varying frequency components of the signal with respect to time and temporally slicing which involves turning a complete sound into small overlapping parts to enhance regional sound analysis. After this process, those Mel Spectrograms are fed to the Convolutional Neural Networks (CNNs) to help the model learn different levels of features and also spot deceptions in the form of deepfake sound attacks. The model has been reported to perform at a remarkable accuracy level of around 95.08% and an overall confidence level of 96.50%. However, there are some challenges, for instance, due to the culture of innovation, developments in deepfake technologies may overtake available detection systems and there will always be concerns about trust with regards to the performances. There is a need for further work especially on improving image extracting processes, improving deep neural networks, and expanding the use cases in the context of real-time surveillance and content moderation.

This paper [19] introduces the SceneFake dataset, the first to focus on detecting fake audio generated by manipulating the acoustic scene rather than typical voice attributes like timbre or prosody. The dataset simulates realistic attacks by applying speech enhancement models to remove original background scenes and adding new ones, thereby altering the context without changing the spoken content. The study evaluates fake audio detection performance using traditional and neural speech enhancement methods across multiple signal-to-noise ratios. Benchmark results reveal that models trained on existing datasets like ASVspoof 2019 fail to generalize effectively to unseen scene manipulations, highlighting the uniqueness and challenge of this attack vector. SceneFake aims to fill this gap, offering a valuable benchmark for developing robust fake audio detection systems.

This study proposes a transfer learning approach—called MfC-RF—for detecting scene-manipulated fake audio. Using the SceneFake dataset, the method extracts MFCC features and applies a Random Forest classifier to generate class probability predictions, which are then used as enhanced input features [20]. This approach significantly improves detection accuracy, outperforming standard machine learning and deep learning models with up to 98% classification accuracy. The study compares multiple algorithms including Random Forest, K-Nearest Neighbors, Logistic Regression, Naive Bayes, and LSTM, with hyperparameter tuning and k-fold cross-validation. Results demonstrate that transfer features improve both precision and recall, particularly under complex acoustic conditions. The work offers a computationally efficient and highly accurate solution for identifying manipulated audio, emphasizing the importance of engineered features in audio forensics.

The research was dedicated to improving the detection of deepfake speech by means of incorporating an image-based method that assesses different spectrogram forms instead of just Mel-spectrograms which is the case currently is mentioned in [16]. It stresses on the importance of spectrogram diversity for the reason that different types of spectrograms can improve detection performance even if the model remains unchanged. The objectives of this study center on spectrogram maximization namely; identifying the most performance image spectrogram regarding the datasets' differences, exploring the accuracy-complexity relationship of processes, and looking at performance over time for the different spectrograms. For this purpose, using the Fake or Real (FoR) dataset, ASVSpoof 2019 LA dataset, and WaveFake datasets the Temporal Convolution Network model techniques were applied to spectrograms STFT, CQT, and MFCC. Analysis revealed MFCC-spectrogram proved to produce the highest accuracy level although the datasets used varied this performance. The STFT-spectrogram was somewhat demanding on resources but did quite well with regard to accuracy as opposed to CQT and VQT spectrograms, which were quite effective in accuracy in small datasets. Future studies will seek to enhance the scope of the datasets, create combined algorithms to test in practice in order to preserve the level of detection in the conditions of deepfake technology development.

The application of voicing onset time (VOT) in detecting deep fake audio content especially in settings where there is a challenge of discerning between original and fake speech was examined in [17]. To achieve this, the use of the ASVSpoof2019 LA dataset, which contains presentations of audio attacks from different text to speech engines was made, and this research was conducted by (1) applying the feature extraction, Google ASR API, Urdu ASR System and other tools such as Praat for formants tracking. Different classification models such as Random Forest (RF), Support Vector Machine(SVM), K-Nearest Neighbors (KNN) and Convolution Neural Networks (CNN) have been applied. The findings show the VOT of the real speech confined to 0–100 ms, while under spoofed audio this can stretch 0–250 ms with the RF model scoring VOT availing an F1-macro score of 71.5%. Coarticulation measures have also shown significant differences between true audio and counterfeit audio. VOT on its own achieves an Equal Error Rate (EER) of 25.2% while coarticulation has an EER of 39.3%, and using both features, the EER improves to 23.5%. The most successful classifier is the CNN model with an EER of as low as 3% when both features are incorporated. On the other hand, one of the major challenges is the pronunciation dictionaries which cause diverse outcomes and the concentration on a very small dimensional feature space. By enhancing the feature sets, making them more resilient, and applying the technique to the other languages with resources, further studies will be conducted.

The significance of forensics in the attribution of deepfakes, particularly focused on how to attribute audio deepfakes to their potential attackers using engaging genetic algorithm based signature development techniques with the ability to identify attackers to a high level of accuracy, which in this case is 97.10% has been examined in [18]. Using the ASVSpoof dataset, this research deals with bona fide speech as well as a range of deepfake speech created by diverse systems. There are Two methods:

first, the number of low level signal features such as the fundamental frequency, jitter, shimmer, speaker gender, duration, loudness and noise ratio are obtained and arranged into sixteen dimensional structures for analysis; secondly, a (BiLSTM) model with three long term recurrent memory layers and a layer dens projection layer of 256 dimensions is built to capture the attacker's signature embeddings. From the assessment, low signal level features present an average class conditional variance of 0.83 and a classification accuracy of 56.96%, while classification using neural embeddings performs remarkably better where the in-domain variance is 0.19 and the out-of-domain variance is 0.46. Some encouraging clustering outcomes on the ASVspoof 2021 dataset reveal the model's capacity to develop beyond that. It does however have limitations related to the dependence on the standard of training data, the big computational power needed, and difficulty in extending to new attackers, hence alluding to possibilities of work in model tuning, augmentation of features, deployment, and increasing user capacity in the future.

2.2 Model Definition

CNN Model

CNNs are an effective method for recognizing deepfake voices by finding minor patterns in audio. They analyze raw sound images using filters to identify minor features. Pooling layers simplify data, whereas fully linked layers identify whether the voice is genuine or manufactured. Activation functions such as ReLU increase complexity, while the model's output provides a probability score. To fine-tune the model, genuine and false audio samples are fed in during training. Despite their limitations, CNNs are critical in the battle against phony voices, and researchers are constantly enhancing their capabilities.

RNN Model

Deepfake Voice Recognition using Recurrent Neural Networks

- RNNs are good tools for deepfake voice recognition because they can handle sequential data.
- RNNs convert raw audio into features such as MFCCs or spectrograms, which track frequency changes over time.
- RNNs employ internal memory to track previous information, which is critical for comprehending the audio's context and flow.
- Variants such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs). are utilized to solve the vanishing gradient problem.
- RNNs use softmax or sigmoid functions to determine if the audio is real or false.
- Despite these problems, RNNs continue to be an effective method for detecting and countering deepfake audio.

LSTM Model

Long Short-Term Memory (LSTM) networks, a form of Recurrent Neural Network (RNN), are useful for deepfake voice recognition because they can capture long term dependencies in sequential input like audio. They overcome the constraints of typical RNNs by managing information flow and evaluating temporal patterns in audio sources. LSTMs are trained on big datasets and can predict complicated temporal correlations, making them excellent at recognizing deepfake voices.

GRU Model

Gated Recurrent Units (GRUs) are a more efficient variant of Recurrent Neural Networks (RNNs) that are useful for deepfake voice recognition. They analyze auditory data phase by phase, employing a system of gates to regulate information flow and maintain context. GRUs are commonly used for deepfake detection applications since they are trained on vast datasets of authentic and synthetic sounds. Despite drawbacks, GRUs are an effective method for fighting deepfake sounds.

BiLSTM Model

A Bidirectional Long Short-Term Memory (BiLSTM) model is a type of recurrent neural network (RNN) which enhances the sequences of learning by processing the data in both forward and backward directions. It consists of two LSTM layers. The first one can read the input sequence from the beginning to end and the another one can read it in a opposite directional way. This bidirectional setup could also allow the model to capture both past and future dependencies in the sequential way that could make it especially useful for more complicating tasks like NL (Natural language) processing, speech recognition of humans and real-time series analysis. A BiLSTM model could achieve more accurate predictions than a standard one way directional LSTM by leveraging information from both directions.

Chapter 3

Data Collection & Analysis

3.1 Data Collection

The dataset used in this paper is the **SceneFake** which is designed for audio deep fake detection. It includes both real and fake voice recordings and is divided into three main subsets: *train*, *development*, and *evaluation*. Each subset contains two categories:

- **Real**: Recordings of real human speech.
- **Faked**: AI-generated or manipulated voice samples.

This dataset is curated to support deepfake detection research. It includes high-quality, suitably labeled audio samples. The files are in `.wav` format, ensuring compatibility with most audio processing frameworks.

The process involved collecting natural audio samples besides the synthesis of speech using the advanced TTS and voice cloning models. The dataset presented herein will allow the learning model to differentiate between natural and AI-generated speech.

3.2 Data Pre-processing and Cleaning

Standardization, cleaning, and feature extraction of the audio data are key preprocessing steps (Fig: 3.1) applied to make the SceneFake dataset suitable for deep learning models. The following steps were involved:

3.2.1 Audio Loading and Standardization

Every `.wav` file is loaded and then transformed into a numerical representation of the waveform. The sampling rate is standardized so that all audio files are the same.

3.2.2 Feature Extraction

We extract a comprehensive set of acoustic features to capture both speaker voice and acoustic scene characteristics, tailored to the SceneFake dataset's manipulation patterns.

- **Mel-Frequency Cepstral Coefficients (MFCCs):** We are using 40 MFCCs per audio frame by using 128 Mel bands and approximately 344 time steps for a 4-second audio clip. MFCCs capture spectral and timbral characteristics, including speaker tone and acoustic scene details (e.g., noise patterns). By this we are making them effective for detecting manipulation artifacts in both voice and background sounds.

- **Decibel-Normalized Spectrograms:** We are generating log-Mel spectrograms by Short-Time Fourier Transform (STFT) with a window size of 25 ms and a hop length of 10 ms. We convert amplitude to decibel scale and normalize to a range of $[-100, 0]$ dB, emphasizing loudness variations critical for detecting scene manipulations (e.g., unnatural reverberation or synthetic noise). The spectrograms have a shape of 128 Mel bands by 344 time steps, suitable for CNN input.

- **Relative Acoustic Features:** We introduce novel features to enhance detection of dynamic manipulations:

  - **- Temporal MFCC Differences:** We calculate first-order differences between consecutive MFCC frames to capture dynamic changes in spectral patterns, such as abrupt shifts in speaker tone or scene noise.
  - **- Decibel Differences:** We compute temporal differences in decibel levels across spectrogram frames to highlight loudness variations, often altered in fake audio (e.g., inconsistent volume changes in synthetic scenes).
  - **- Speaker Tone Features:** We derive simplified tone features by averaging the first two MFCC coefficients (coefficients 1–2) across frames, capturing pitch-related characteristics of the speaker's voice, which may differ between real and manipulated audio.

- **Feature Concatenation:** For sequential models (GRU, LSTM, BiLSTM), we concatenate MFCCs, MFCC differences, decibel differences, and tone features into a feature vector of approximately 129 dimensions per time step (40 MFCCs + 40 MFCC differences + 48 frequency bins from decibel differences + 1 tone feature). For CNN, we use the 2D spectrogram (128×344) directly, with relative features integrated via a custom layer during model processing.

3.2.3 Normalization and Standardization

We normalize the extracted features to ensure consistent input ranges for model training:

- **MFCCs:** We standardize MFCCs to have zero mean and unit variance across the dataset that reduce the sensitivity to amplitude different variations.

- **Decibel Spectrograms:** We clip decibel values from -100 to 0 dB, in short $[-100, 0]$ dB and normalized it to $[0, 1]$ to use it for CNN input that ensures uniform scaling.

- **Relative Features:** We normalize temporal differences (MFCC and decibel) to $[-1, 1]$ based on their observed ranges, preserving dynamic information.

- **Tone Features:** We scale tone features to $[0, 1]$ based on their minimum and maximum values, ensuring compatibility with other features.

3.2.4 Output Formats

We prepare two output formats to support all four models:

- **For CNN (SceneFakeNet):** We save decibel-normalized spectrograms as 2D arrays with shape (128 Mel bands $\times$ 344 time steps $\times$ 1 channel), suitable for convolutional processing.
- **For GRU, LSTM, BiLSTM (SceneFakeGRU, SceneFakeLSTM, SceneFakeBiLSTM):** We as usually save the concatenated feature sequences with the shape (344 time steps $\times$ 129 dimensions) and optimized it for sequential processing in further steps.

We store the preprocessed features of the data sets and labels in such a structured format (e.g., NumPy arrays or HDF5 files) so that we could facilitate efficient loading during the model training.

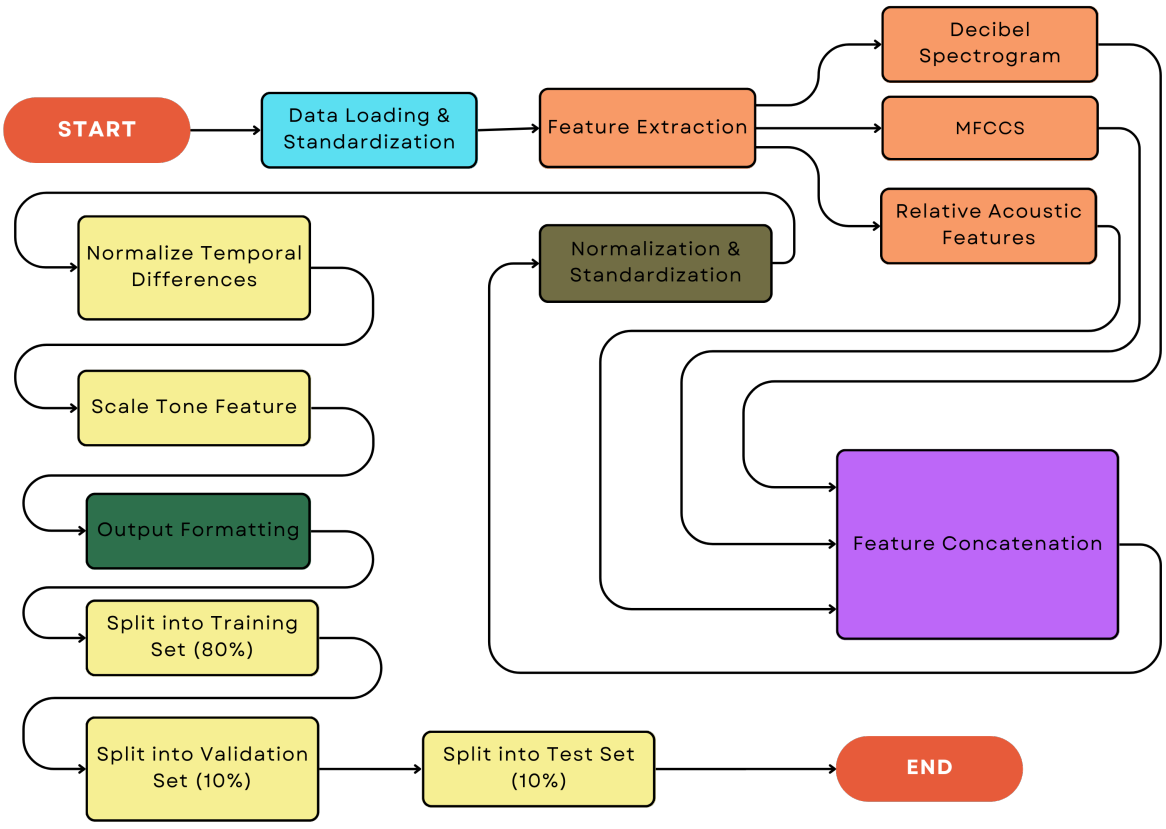

Figure 3.1: Data Pre-processing Workflow

3.2.5 Data Splitting

We split the whole dataset into disjoint subsets so that would be used for training and evaluation of deep learning models:

- **Training Set (80%):** Used to train the models.
- **Validation Set (10%):** Used for hyperparameter tuning and overfitting prevention.
- **Test Set (10%):** Used to evaluate model performance on unseen data.

This makes the SceneFake dataset structured in such a way that any deep learning model, like LSTM, CNN, and GRU, can learn effectively for the differentiation of real and fake recordings.

3.3 Data Analysis:

3.3.1 Distribution of Audio Durations:

This histogram (Fig: 3.2) is visualizing the frequency of audio clips across different durations for both real and fake categories. By observing the shape and spread of these distributions, we could gain insights into the typical lengths of audio recordings within each particular category. For example, if one category continuously consists of shorter clips, it could indicate a potential bias in the data collection process or there could be a differences in typical recording scenarios for real and manipulated audio.

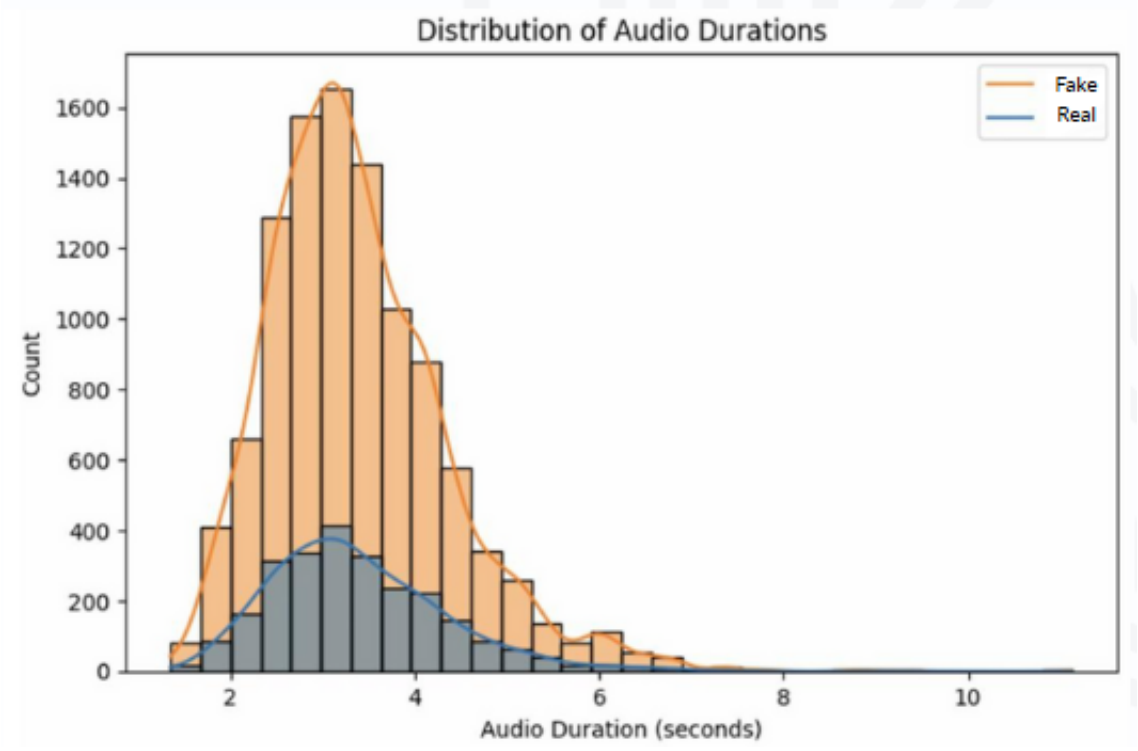

Figure 3.2: Distribution of Audio Durations

3.3.2 Boxplot of Audio Durations:

This boxplot (Fig: 3.3) is showing an exact summary of the distribution of audio durations for each category (real and fake). It also visually highlighting the key statistical measures such as the median, quartiles (25th and 75th percentiles) and the presence of outliers. We can readily observe differences in the central tendency, dispersion, and skewness of audio durations between real and fake recordings by comparing all the boxplots. For example, if the median duration is significantly different between the two above 2 categories or if one category represents a wider range of durations, it could be a suggestion of potential differences in recording conditions or manipulation techniques

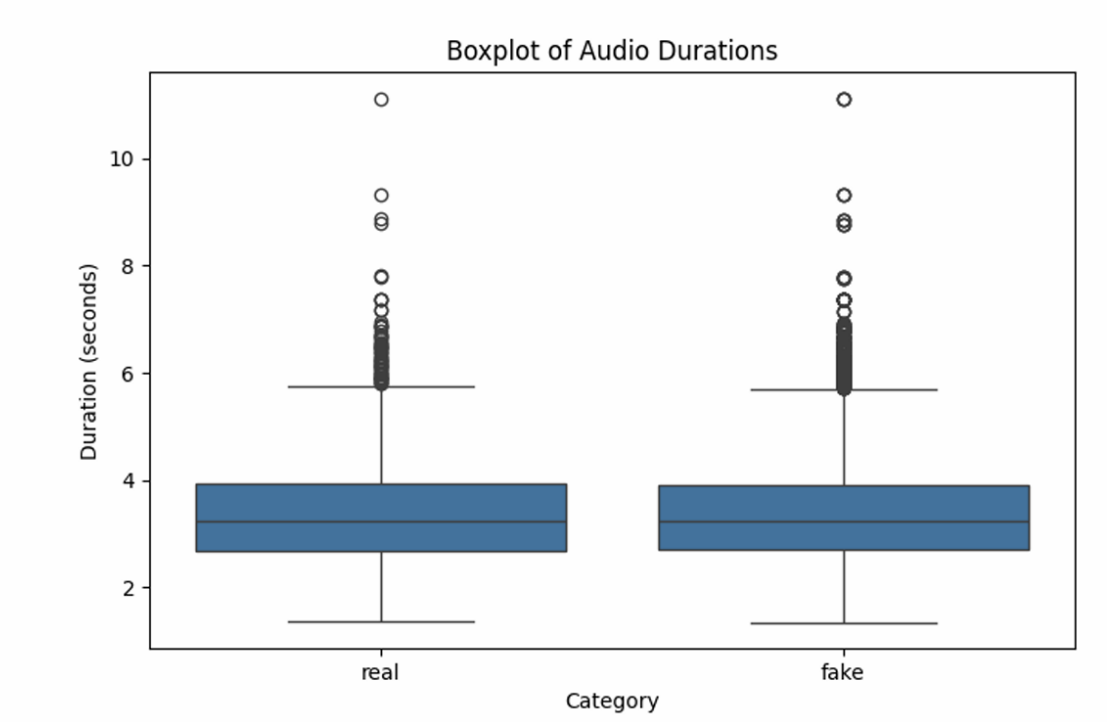

Figure 3.3: Boxplot of Audio Durations

3.3.3 MFCC Mean Values per Class:

This line plot (Fig: 3.4) depicts the mean values of each MFCC coefficient for both real and fake audio. By observing the trends and patterns in these mean values across different coefficients, we can identify potential spectral characteristics that distinguish real and manipulated audio. For example, if the certain coefficients consistently is showing us the higher or lower mean values in one category comparing to the other one, it can indicate specific spectral features that are indicative of scene-faked audio, such as the presence of artificial voice or noise introduced during the manipulation process.

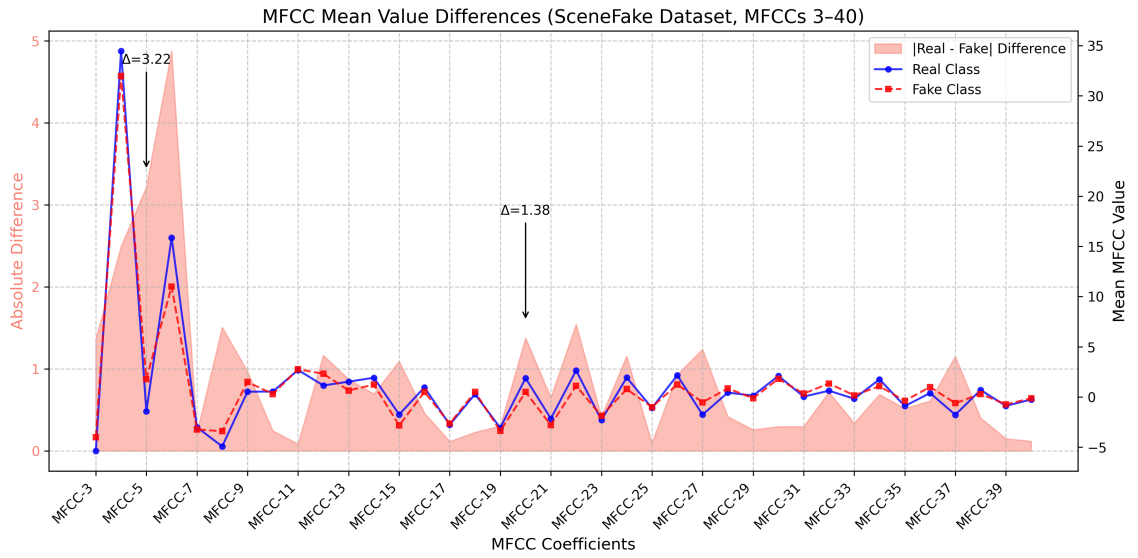

Figure 3.4: MFCC Mean Values per Class

3.3.4 Scatter Plot of Duration vs. Zero-Crossing Rate:

This scatter plot (Fig: 3.5) is explaining the relationship between audio duration and the zero crossing rate for both real and fake audio. By visualizing the data points for each category, we could identify potential correlations, trending or clustering patterns. For example, we may observe that longer audio clips can tend to have higher zero-crossing rates in one category compared to the other one that could suggest a relationship between these features and the presence of specific acoustic phenomena. Such as the background noise or the presence of high-frequency components.

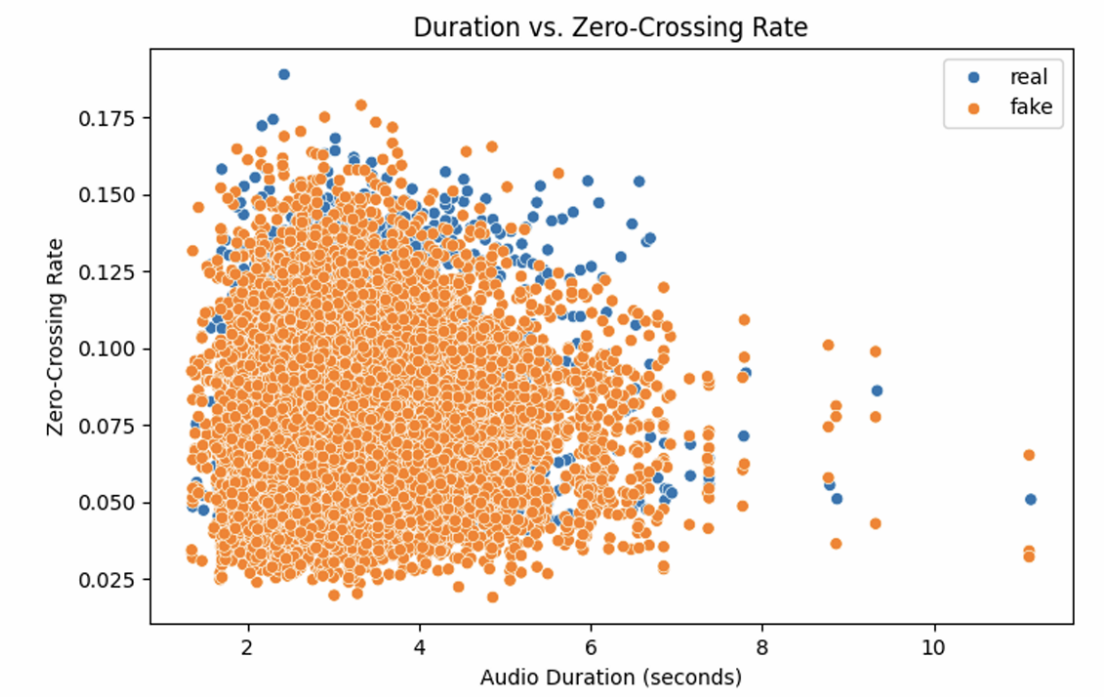

Figure 3.5: Duration vs Zero Crossing Rate

3.3.5 Density Plot of Zero-Crossing Rates:

This plot (Fig: 3.6) is showing the probability density function of zero-crossing rates for real and fake audio. We can discover the exact differences in the dynamic properties of audio signals in each category by comparing the distributions' forms and characteristics. For example, if one category has a higher fraction of high zero-crossing rates, it may imply the presence of more high-frequency components or faster signal fluctuations, both of which may be hallmarks of specific manipulation techniques.

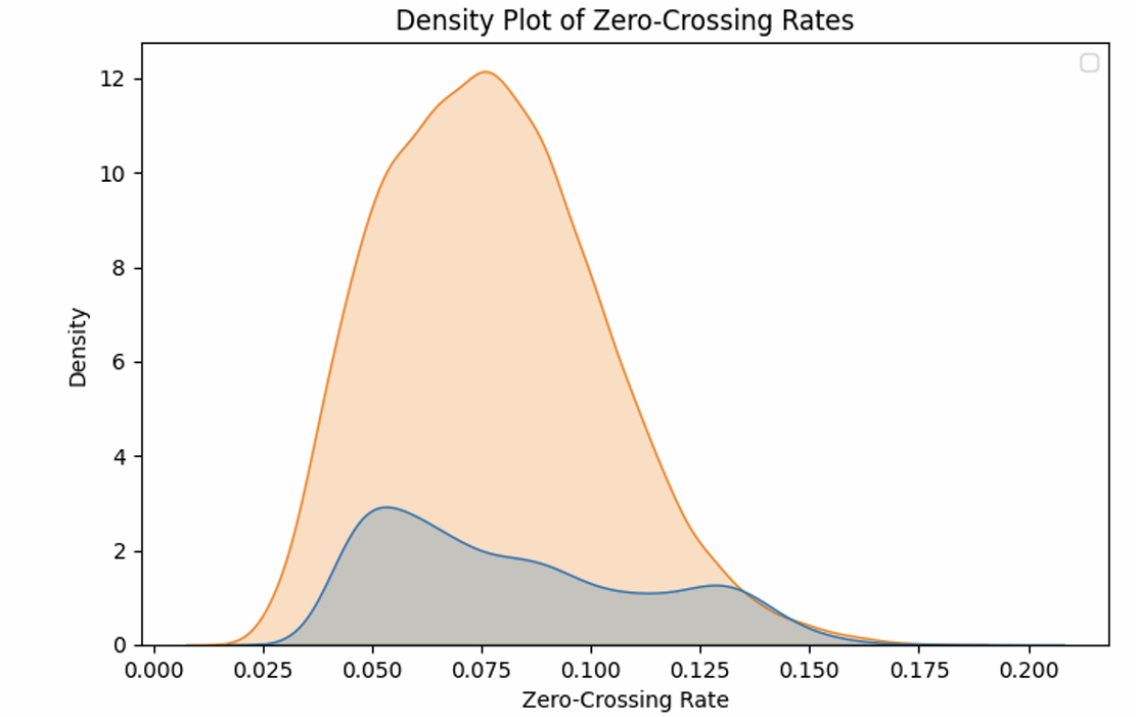

Figure 3.6: Density Plot of Zero-Crossing Rates

3.4 Spectrograms:

3.4.1 Mel-Frequency Cepstral Coefficients Spectrograms:

Real Sample:

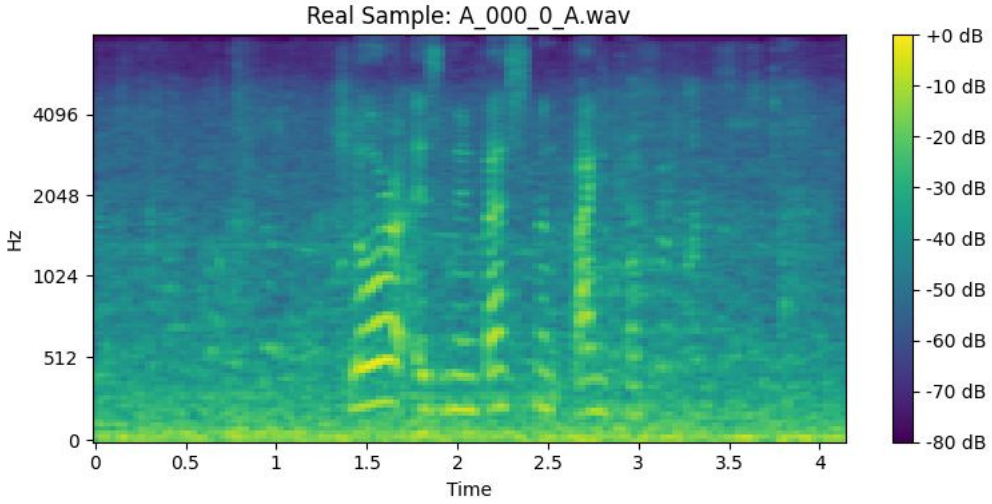

Figure 3.7: Real Sample: A_000_0_A.wav

Fake Sample:

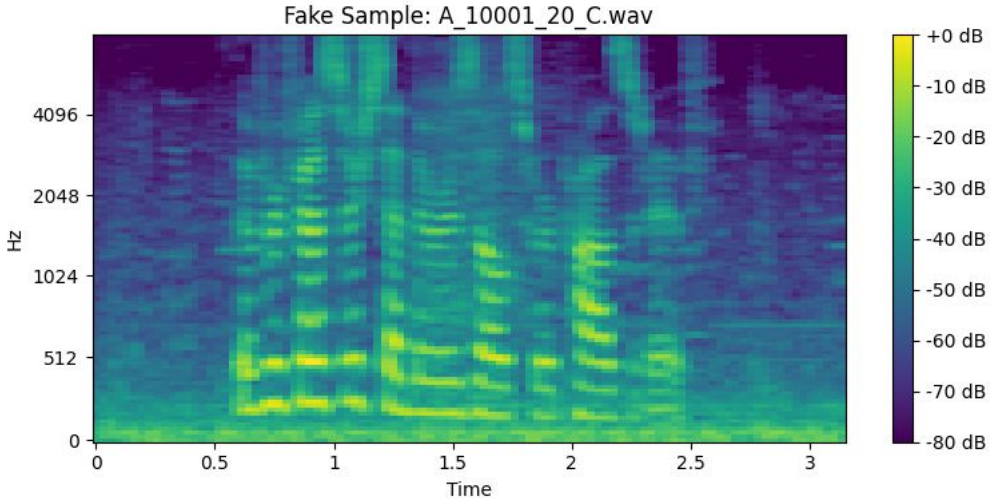

Figure 3.8: Fake Sample: A_10001_20_C.wav

3.4.2 Decibel-Normalized Spectrograms:

Real Sample:

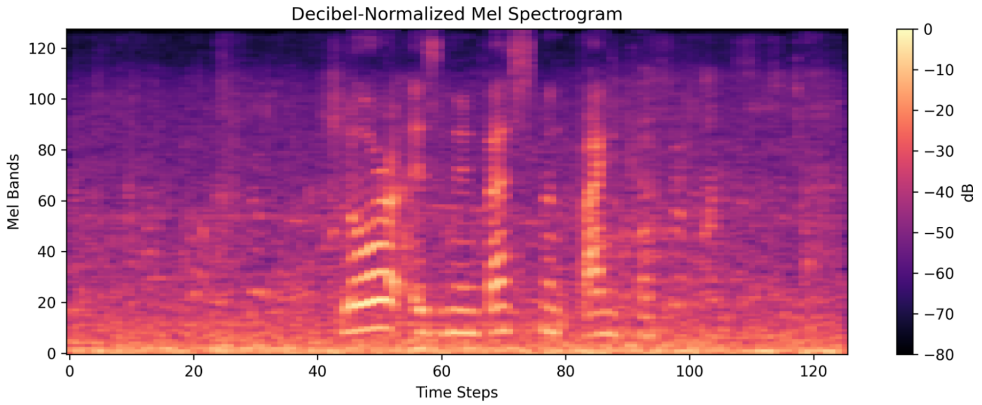

Figure 3.9: Decibel-Normalized Mel Spectrograms - Real

Fake Sample:

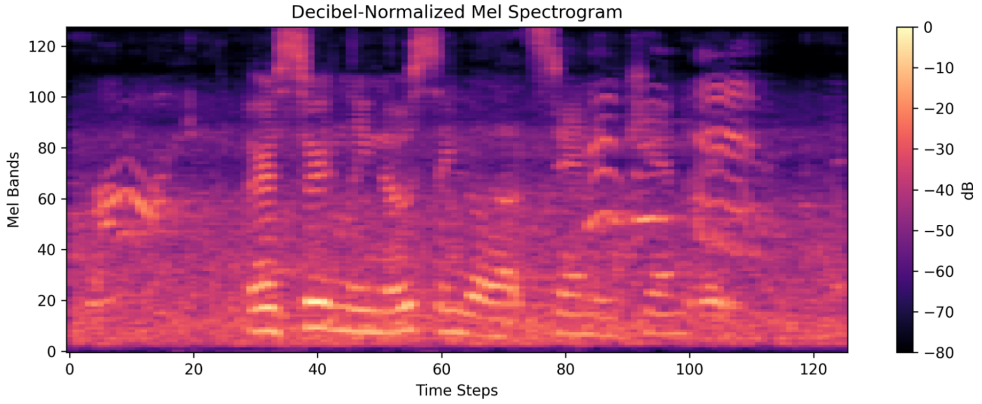

Figure 3.10: Decibel-Normalized Mel Spectrograms - Fake

Chapter 4

Model Architecture

4.1 CNN Architecture

We propose a custom CNN architecture (Fig: 4.1), named SceneFakeNet, designed to capture relative acoustic features, speaker tone variations, and decibel-based characteristics for real vs. fake audio classification. The architecture processes Mel-Frequency Cepstral Coefficients (MFCCs) and decibel-normalized spectrograms as input features.

Architecture Details

- **Input Layer:** Accepts 2D spectrograms (MFCCs and decibel-normalized spectrograms) with shape (batch_size, 1, 128, 344) (128 Mel bands, 344 time steps).
- **Convolutional Block 1:**
  - Conv2D: 32 filters, 3x3 kernel, ReLU activation, stride 1
  - Batch Normalization
  - MaxPooling2D: 2x2 pool size
  - Dropout: 0.25
- **Convolutional Block 2:**
  - Conv2D: 64 filters, 3x3 kernel, ReLU activation, stride 1
  - Batch Normalization
  - MaxPooling2D: 2x2 pool size
  - Dropout: 0.25
- **Convolutional Block 3:**
  - Conv2D: 128 filters, 3x3 kernel, ReLU activation, stride 1
  - Batch Normalization
  - MaxPooling2D: 2x2 pool size
  - Dropout: 0.3

- **Relative Feature Extraction Layer:**
  - Custom layer to compute relative acoustic features (e.g., differences in MFCC coefficients across time frames to capture speaker tone dynamics).
  - Computes relative decibel differences between adjacent time frames to emphasize loudness variations.
- **Fully Connected Layers:**
  - Flatten output from convolutional blocks
  - Dense Layer 1: 256 units, ReLU activation, Dropout 0.4
  - Dense Layer 2: 128 units, ReLU activation, Dropout 0.4
  - Output Layer: 2 units (real/fake), Softmax activation
- **Output:** Binary classification (real or fake audio).

Loss Function

- Binary Cross-Entropy Loss
- Optimizer: Adam with learning rate 0.001
- Metrics: Accuracy, Equal Error Rate (EER)

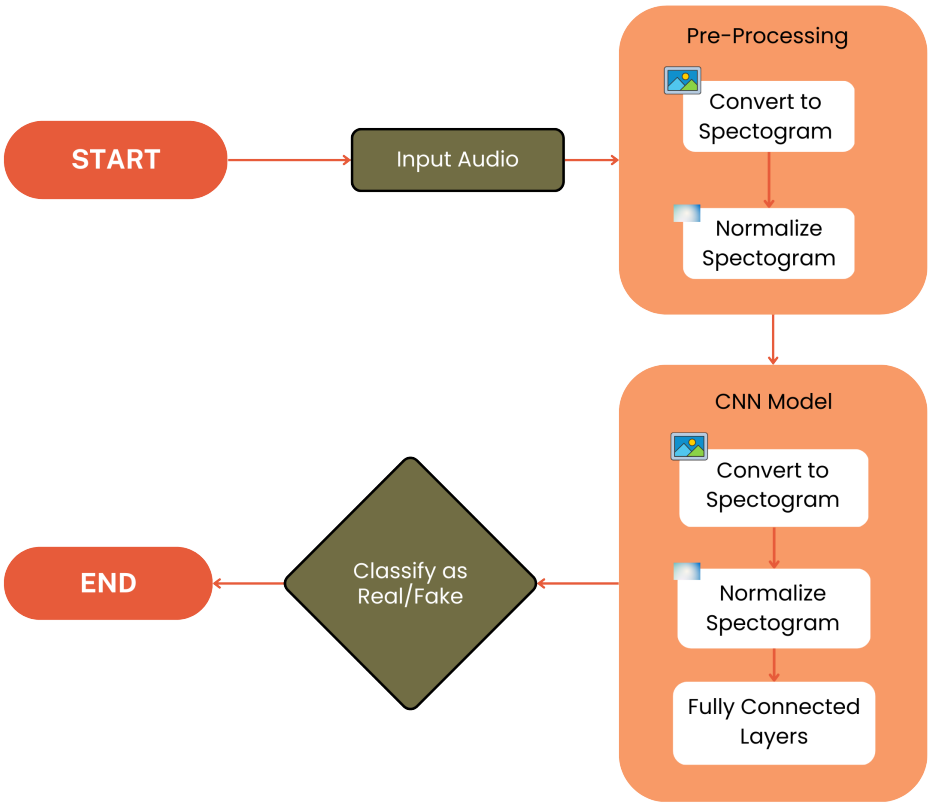

Figure 4.1: CNN Architecture

We train the SceneFakeNet model using a dataset called SceneFake, which has both real and fake audio clips. First, we go through each audio file and pull out useful details like MFCC features (which describe the sound), loudness levels, and how the sound changes over time. We also check things like tone of voice. We gathered all these details into a single set of features. If the audio is genuine, we mark it as 0; if it's fake, we mark it as 1. Next, we divide the data into two sections: 80% for training the model and 20% for testing it, ensuring that both sections have a good mix of real and fake samples. We utilize a tool known as a data loader to feed this data into the model in small groups, which we call batches. We set up the SceneFakeNet model, choose the Adam optimizer to help it learn (with a learning speed of 0.001), and use a method called binary cross-entropy to measure how well the model is doing. For each round of training (called an epoch), we pass the data through the model, get predictions, check how far off the predictions are, and then adjust the model to do better. After each round, we test the model on the test data and measure how accurate it is, as well as the Equal Error Rate (EER), which shows how often the model mixes up real and fake. Once training is done, we save the model to a file named `"scenefakenet_model.pth"`. We also explain how we calculate the extra features, like changes in sound over time, changes in loudness, and tone of voice. During testing, we keep track of how many answers the model got right and average the EER scores to see how well it performs.

4.2 GRU Architecture

We propose SceneFakeGRU, a Gated Recurrent Unit (GRU) based architecture (Fig: 4.2) designed to classify real and fake audio samples from the SceneFake dataset by capturing temporal dependencies in acoustic features. The model processes sequential features, including Mel-Frequency Cepstral Coefficients (MFCCs) (Fig: 3.7 & Fig: 3.8), Decibel-Normalized spectrograms (Fig: 3.9 & Fig: 3.10), and novel relative features.

Architecture Details

- **Input Layer:** Accepts feature sequences of shape (batch_size, sequence_length, feature_dimension), where feature_dimension is approximately 129 (40 MFCCs + 40 MFCC differences + 48 frequency bins + 1 tone feature).
- **GRU Block 1:**
  - GRU Layer: 128 hidden units to capture temporal patterns in acoustic features.
  - Dropout: 0.3 to prevent overfitting.
- **GRU Block 2:**
  - GRU Layer: 128 hidden units, stacked to model higher-level temporal dependencies.
  - Dropout: 0.3.
- **Fully Connected Layers:**

  - Dense Layer 1: 64 units, ReLU activation, Dropout 0.3.
  - Output Layer: 2 units (real/fake), Softmax activation.
- **Output:** Binary classification (real or fake audio).
- **Loss Function:** Binary Cross-Entropy Loss.
- **Optimizer:** Adam with a learning rate of 0.001.
- **Metrics:** Accuracy and Equal Error Rate (EER).

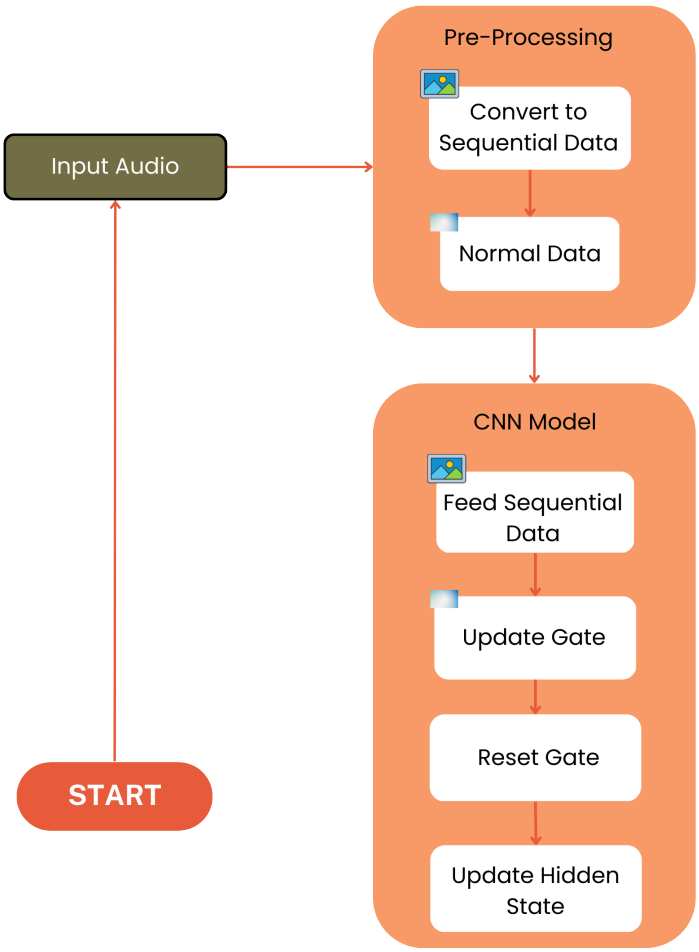

Figure 4.2: GRU Architecture

We train the SceneFakeGRU model using audio clips from a dataset called SceneFake, which has both real and fake recordings in separate folders. For each clip, we pull out helpful sound details like 40 MFCCs (these show the shape and tone of the sound), a spectrogram (which shows loudness over time), and extra features like how the sound and loudness change from moment to moment, and the speaker's tone or pitch. We put all these features together and make sure every clip is the same length by trimming or padding. We label real clips as 0 and fake ones as 1. Then, we split the data — 80% for training the model and 20% for testing it — keeping a good mix of real and fake in both parts. The model, SceneFakeGRU, uses GRU layers (a type of memory unit) to learn how the sound changes over time. It also has an attention layer that helps it focus on the most important parts of the audio. We rely on a handy tool called the Adam optimizer to train our model, along with a loss function known as binary cross-entropy to gauge how far off the model's predictions are from the correct answers. During each training cycle, or epoch, we feed the model small batches of data, let it make guesses about whether clips are real or fake, evaluate its performance, and then guide it to learn from any errors. After every round, we check how many clips it got right (that's its accuracy) and how often it confuses real with fake (which we refer to as EER, or Equal Error Rate). We present the results after each training session. Once training wraps up, we save the model for future use. We can also experiment with different settings, like adjusting the learning speed or the number of layers, to see if we can enhance its performance even further.

4.3 LSTM Architecture

We propose SceneFakeLSTM, a Long Short-Term Memory (LSTM) based architecture (Fig: 4.3) designed to classify real and fake audio samples from the SceneFake dataset by capturing long-term temporal dependencies in acoustic features. The model processes sequential features, including Mel-Frequency Cepstral Coefficients (MFCCs), decibel-normalized spectrograms, and novel relative features.

Architecture Details

- **Input Layer:** Accepts feature sequences of shape (batch_size, sequence_length, feature_dimension), where feature_dimension is approximately 129 (40 MFCCs + 40 MFCC differences + 48 frequency bins + 1 tone feature).
- **LSTM Block 1:**
  - LSTM Layer: 128 hidden units to capture temporal patterns in acoustic features, with forget, input, and output gates to manage long-term dependencies.
  - Dropout: 0.3 to prevent overfitting.
- **LSTM Block 2:**
  - LSTM Layer: 128 hidden units, stacked to model higher-level temporal dependencies.
  - Dropout: 0.3.
- **Fully Connected Layers:**
  - Dense Layer 1: 64 units, ReLU activation, Dropout 0.3.
  - Output Layer: 2 units (real/fake), Softmax activation.
- **Output:** Binary classification (real or fake audio).
- **Loss Function:** Binary Cross-Entropy Loss.
- **Optimizer:** Adam with a learning rate of 0.001.
- **Metrics:** Accuracy and Equal Error Rate (EER).

We train the SceneFakeLSTM model using sound clips from a dataset called SceneFake, which has two folders—one with real clips and one with fake ones. For each clip, we pull out sound features like 40 MFCCs (which help describe how the sound looks), a loudness map called a spectrogram (measured using decibels), and some extra features like how the sound and loudness change over time and the speaker's pitch (using the first couple of MFCCs). We put all these details together for each small part of the audio. We also make sure every clip is the same length by cutting or padding them. Then, we label real clips as 0 and fake ones as 1. We split the data into two parts—80% for training and 20% for testing—making sure both parts

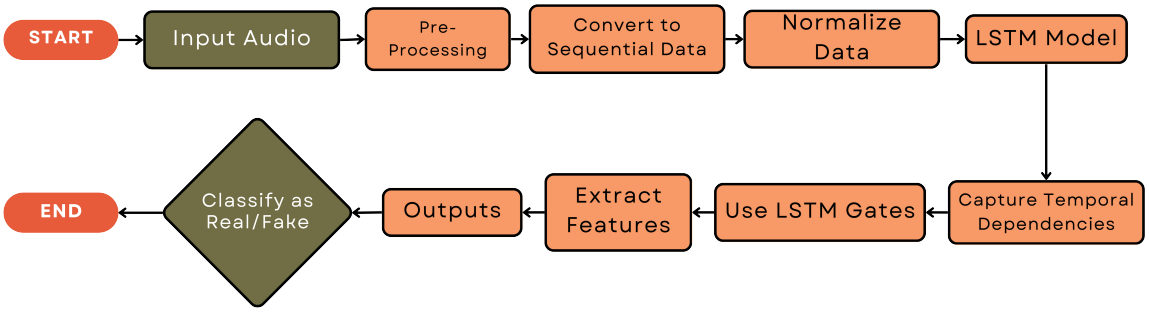

Figure 4.3: LSTM Architecture

have a good mix of real and fake clips. The model we use, SceneFakeLSTM, has two LSTM layers that help it remember the order of sounds, an attention part that helps it focus on the most important sounds, and some final layers that decide if the clip is real or fake. We use something called the Adam optimizer to help the model learn (with a learning speed of 0.001), and we check how wrong the model's guesses are using a method called binary cross-entropy loss. In each training round (called an epoch), we give the model small batches of clips, let it guess, check how it did, and help it improve. After each round, we test how many guesses were right (accuracy) and how often it mixes up real and fake (called EER). We share the results after each round of training. Once the training is complete, we save the model for future use. If necessary, we can tweak some settings, like the learning speed or the number of LSTM layers, to enhance its performance even further.

4.4 BiLSTM Architecture

We propose SceneFakeBiLSTM, a Bidirectional Long Short-Term Memory (BiLSTM) based architecture (Fig: 4.4) designed to classify real and fake audio samples from the SceneFake dataset by capturing both forward and backward temporal dependencies in acoustic features. The model processes sequential features, including Mel-Frequency Cepstral Coefficients (MFCCs), decibel-normalized spectrograms, and novel relative features.

Architecture Details

- **Input Layer:** Accepts feature sequences of shape (batch_size, sequence_length, feature_dimension), where feature_dimension is approximately 129 (40 MFCCs + 40 MFCC differences + 48 frequency bins + 1 tone feature).
- **BiLSTM Block 1:**
  - BiLSTM Layer: 128 hidden units (64 forward + 64 backward) to capture bidirectional temporal patterns in acoustic features, with forget, input, and output gates.
  - Dropout: 0.3 to prevent overfitting.
- **BiLSTM Block 2:**
  - BiLSTM Layer: 128 hidden units (64 forward + 64 backward), stacked to model higher-level bidirectional dependencies.
  - Dropout: 0.3.
- **Fully Connected Layers:**
  - Dense Layer 1: 64 units, ReLU activation, Dropout 0.3.
  - Output Layer: 2 units (real/fake), Softmax activation.
- **Output:** Binary classification (real or fake audio).
- **Loss Function:** Binary Cross-Entropy Loss.
- **Optimizer:** Adam with a learning rate of 0.001.
- **Metrics:** Accuracy and Equal Error Rate (EER).

We train the SceneFakeBiLSTM model using sound clips from the SceneFake dataset, which has real and fake audio in different folders. For each clip, we pull out sound features like 40 MFCCs (which show how the sound behaves), a loudness chart called a spectrogram (adjusted using decibels), and extra features like how the sound and loudness change over time and the speaker's tone or pitch. We combine all these features and make sure all clips are the same length by trimming or padding them. We label real clips as 0 and fake ones as 1. Then we split the data—80% for training and 20% for testing—while keeping a good mix of real and fake clips. The model, SceneFakeBiLSTM, has two BiLSTM layers that learn from the sound both forward

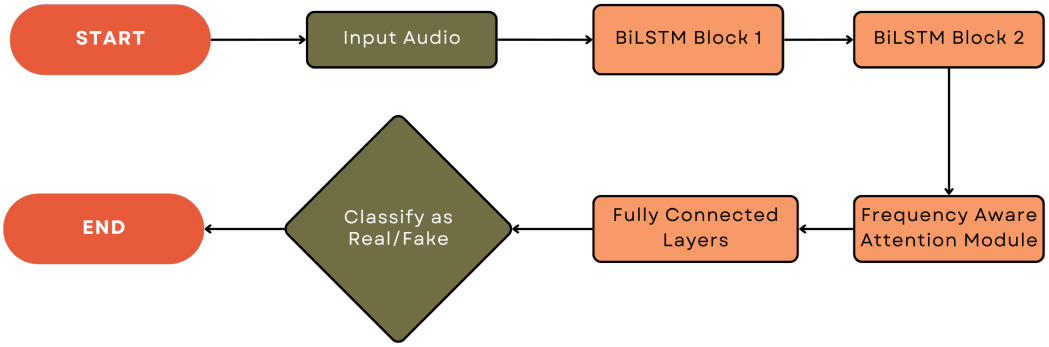

Figure 4.4: BiLSTM Architecture

and backward, an attention layer that helps focus on the most important parts, and final layers that decide if the clip is real or fake. We utilize the Adam optimizer with a learning rate of 0.001 to guide the model's learning process. To evaluate its performance, we employ a loss function known as binary cross-entropy, which helps us understand how far off its predictions are. During each training cycle, or epoch, we feed the model small batches of clips, allowing it to make predictions, assess its accuracy, and refine its approach. After every round, we check how well the model performed by looking at its accuracy (the number of clips it got right) and the Equal Error Rate (EER), which indicates how often it misidentifies real clips as fake. We share these results after each round. Once the training is complete, we save the model for future use. Additionally, we can tweak parameters like the learning rate or the number of layers to enhance its performance even further.

Chapter 5

Result Analysis

5.1 Evaluation Metrices

Accuracy

In general, accuracy denotes the overall performance metric, which measures the proportion of correctly classified instances or cases (both positives and negatives) with respect to the total number of instances. Accuracy is suitable when the data set is balanced and the number of positive and negative instances is roughly equal.When dealing with datasets where one class dominates, interpreting accuracy can get a bit tricky. Take, for instance, a scenario where 95% of the data points are negative and only 5% are positive. In this case, if the model predicts every single case as negative, it would still boast an impressive 95% accuracy. This shows that relying solely on accuracy can be misleading, especially if we overlook the positive class. So, it's crucial to approach accuracy with caution when working with imbalanced datasets. The formula for calculating accuracy is:

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Positive}} \tag{5.1}$$

where TN (True Negatives) are the correctly predicted negative instances. While accuracy provides an overall measure of performance, it does not indicate the importance of the false positives and false negatives.

Precision

Precision is the ratio of positive instances that are correctly predicted to all positive instances predicted, an important measure when there is a heavy penalty for false positives, like in spam detection, where classifying a legitimate email as spam can be quite detrimental. A model achieving high precision implies that most of the instances predicted as positive were indeed positive and therefore would account for fewer incorrect positive classifications. This still does not mean that it is the best model to use; low recall means that it has missed a sizable number of actual positives. The precision can therefore be stated mathematically as:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{5.2}$$

Where TP(true positives) is interpreted as correctly predicted positive instances and FP (false positives) as instances that are incorrectly labeled positive.

Recall

Recall, otherwise called Sensitivity, measures a model's ability to classify actual positive cases. It's primarily used in situations of high risk for missing a positive instance, such as a medical diagnosis. For example, in cancer diagnosis, the false negative of a non-identification of a cancerous tumor would be life-threatening; hence, recall forms an extremely important parameter. High recall indicates that many of the actual positive cases are being properly identified; however, this often leads to a high number of false positives. Recall can be mathematically defined by the equation below:

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{5.3}$$

where FN (False Negatives) are these instances that were truly positive but were incorrectly classified by the model as being negative. High recall would mean that the model is missing fewer positive cases after all, which is a good thing in high-risk applications.

F1 Score

The F1 Score is the only metric which defines a balance between precision and recall by taking the harmonic mean of the two. This scenario is helpful where there is a serious imbalance between positive and negative classes when optimizing only one of the measures could lead to very misleading conclusions. If there exists high precision but low recall, it would mean that the model is able to correctly predict a handful of positives while missing a great number of others. On the contrary, the model may achieve high recall but low precision in cases where it captures a majority of the actual positive cases, but with a good number of false positives. Thus, it serves a single measure to evaluate both factors. The F1 Score can be calculated using (Formula). The larger the F1 Score, the better is the balance between precision and recall which connotes that it is the best measure whenever both false positives and false negatives are of concern. On the other hand, the clear advantage of the F1 Score is that it does not get fooled by the imbalanced datasets like accuracy does; thus, it gives a very realistic assessment of the model performance particularly in cases of classification problems where classes have different distributions.

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.4}$$

To wrap things up, precision, recall, accuracy, and the F1 Score each play unique roles in evaluating how well the model performs.While accuracy gives a rough idea, precision and recall help one to better understand how the model works for positive cases. F1 Score improves upon this/differentiates itself by balancing precision and recall; it is thus very useful for handling imbalanced data. The selection of the relevant metric depends on the unique demands of the task, be it the minimization of false positives, minimization of false negatives, or finding an optimal trade-off in

between.

Equal Error Rate (EER)

Equal Error Rate (EER) is the rate at which False Rejection Rate (FRR) and the False Acceptance Rate (FAR) are equal. It is used for measuring the performance of biometric and authentication systems. The goal is to minimize both false positives (admitting unauthorized users incorrectly) and false negatives (denying authorized users incorrectly). The lower the EER, the better the system performance.
Mathematically, the EER is defined as:

$$\text{EER} = \text{False Positive Rate} = \text{False Negative Rate} \tag{5.5}$$

Where:

- **False Acceptance Rate (FAR):** The probability with which an unauthorized user is mistakenly accepted.
- **False Rejection Rate (FRR)**: The percentage of times when valid users are in correctly rejected.

The False Positive Rate (FPR) is given by:

$$\text{FPR} = \frac{\text{False Positive}}{\text{False Positive} + \text{True Negative}} \tag{5.6}$$

The False Negative Rate (FNR) is given by:

$$\text{FNR} = \frac{\text{False Negative}}{\text{False Negative} + \text{True Positive}} \tag{5.7}$$

EER provides a compromise between these two types of errors and is particularly convenient when both false positives and false negatives are to be estimated.

5.2 Model Result

5.2.1 CNN Model Result Analysis:

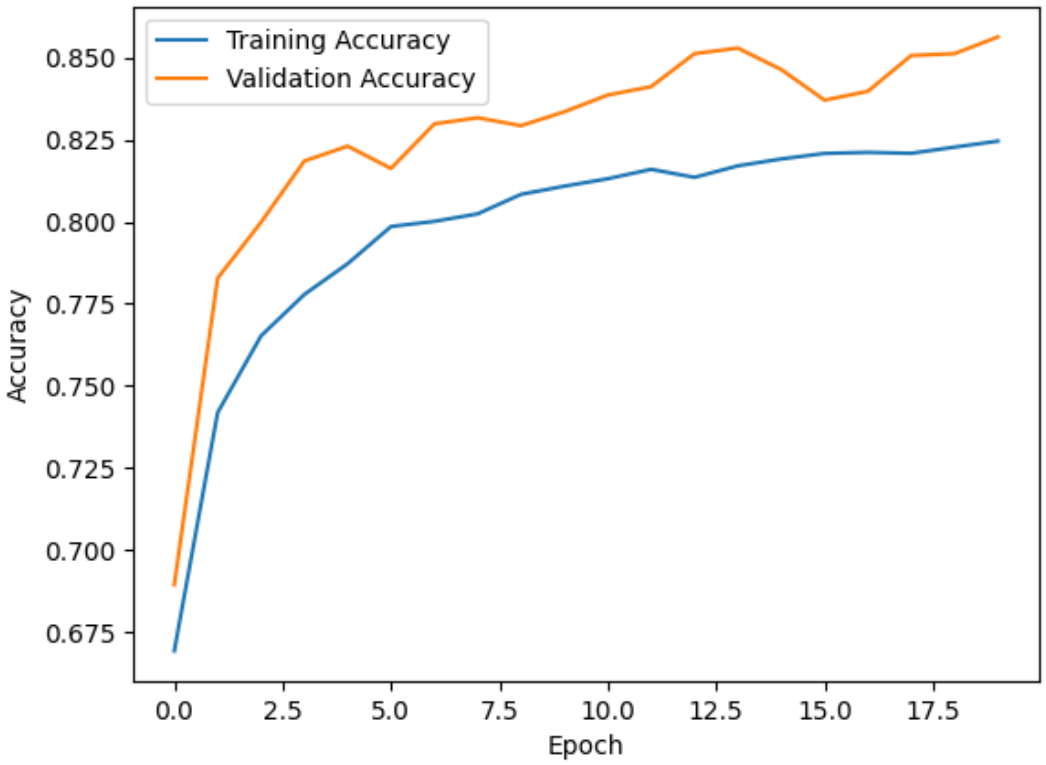

Figure 5.1: CNN Model Training & Validation Accuracy

The CNN model is demonstrating solid learning and generalization. If we take a look at the accuracy graph (Fig: 5.1), it's clear that both the training and validation accuracy are steadily climbing as the epochs progress. What's even more encouraging is that the validation accuracy remains higher than the training accuracy, which is a fantastic indicator. This implies that the model isn't overfitting and is effectively absorbing the information from the data.

Looking at the confusion matrix, the model predicted:

- **Class 0 (negatives)** correctly 4,398 times and incorrectly 487 times.
- **Class 1 (positives)** correctly 3,122 times and incorrectly 822 times.

This indicates that the model is quite adept at recognizing both classes, with a slight edge in identifying Class 0. Overall, its performance is well-balanced, demonstrating a solid understanding of both categories.

In the final training results (Tab: 5.1), your model achieved approximately 82.6% accuracy during training and 85.6% during validation, with a validation loss around 0.3854, which is impressive. This suggests that the model actually performs a bit

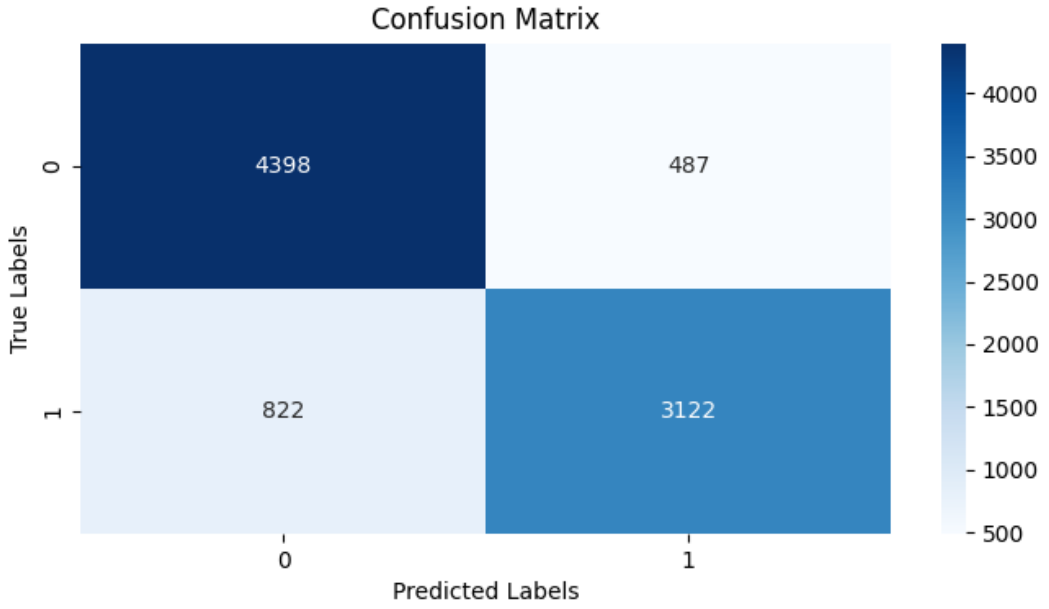

Figure 5.2: CNN Model Confusion Matrix

better on new, unseen data compared to the training data—a bit unusual, but it can occur when the data is well-balanced or when regularization techniques are effectively aiding the model.

To wrap it up, the CNN model is doing quite well, boasting high accuracy, low loss, and a balanced classification as indicated by the confusion matrix (Fig: 5.2). There are no obvious signs of overfitting, and the validation outcomes imply that your model is capable of generalizing effectively to new data.

Metric	Training Accuracy	Validation Accuracy	Validation Loss	Precision	Recall	F1-Score
CNN	82.6%	85.6%	0.385	86.4%	79.2%	82.6%

Table 5.1: CNN Model Performance

5.2.2 LSTM Model Result Analysis:

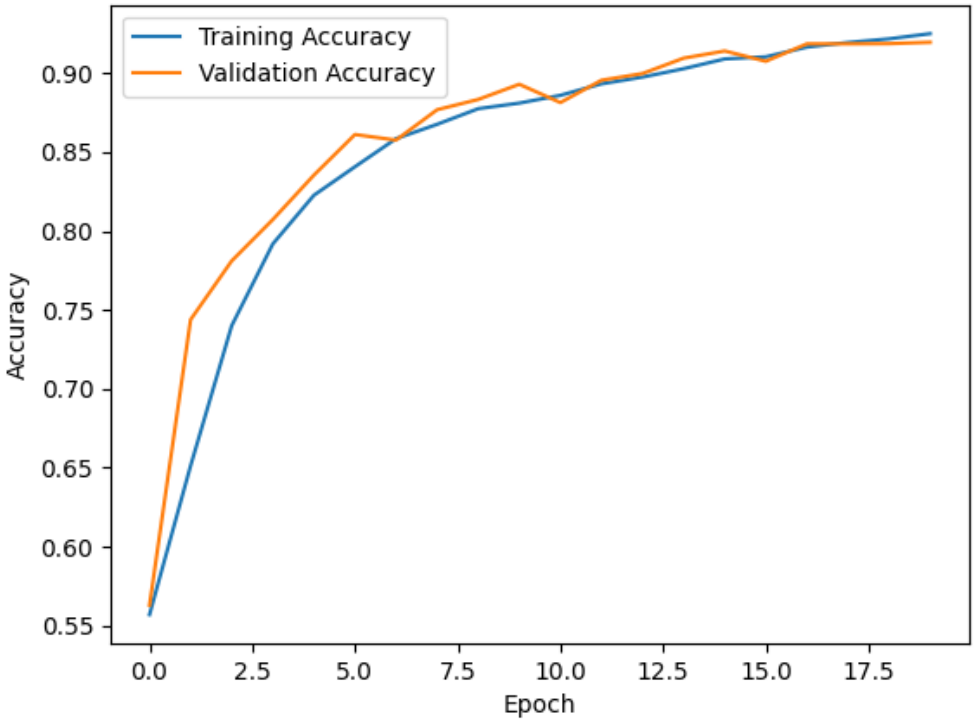

Figure 5.3: LSTM Model Training & Validation Accuracy

The graphs for training and validation accuracy (Fig: 5.3) indicate that the model is picking up the learning process quite effectively. We can see a steady improvement in accuracy over the epochs (Tab: 5.2), hitting around 92% for training and approximately 91.97% for validation. The training and validation lines are closely aligned, which suggests that the model isn't overfitting and is also performing well with new data.

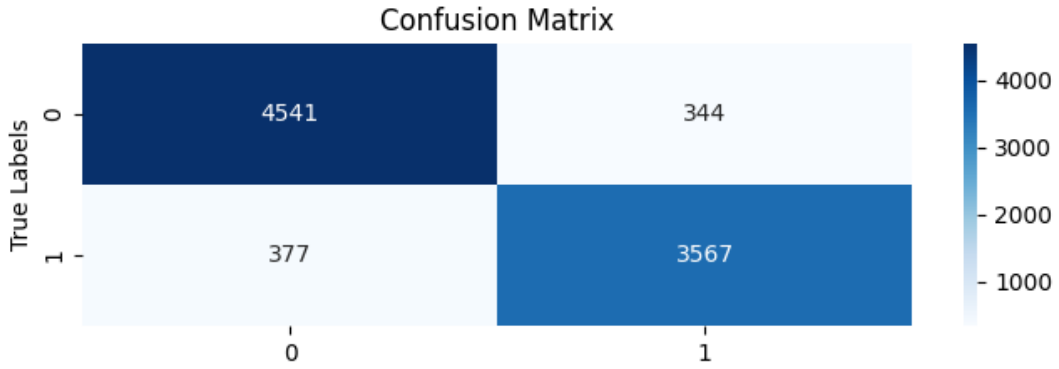

Figure 5.4: LSTM Model Confusion Matrix

In the confusion matrix (Fig: 5.4), we see:

- Class 0: Correctly predicted 4541 times, misclassified 344 times
- Class 1: Correctly predicted 3567 times, misclassified 377 times

This demonstrates a well-balanced classification. The model effectively manages both classes without showing a preference for one over the other. Looking at the training logs, we can observe that the loss values are impressively low for both training and validation, hovering around 0.22–0.23, which indicates that the learning process is both stable and accurate.

In summary, the LSTM model shines with over 91% accuracy on both the training and validation sets. The low loss values and the confusion matrix reveal balanced predictions for each class. There's no indication of overfitting, making the model stable and trustworthy for future predictions.

Metric	**Training Accuracy**	**Validation Accuracy**	**Validation Loss**	**Precision**	**Recall**	**F1-Score**
LSTM	92.2%	91.97%	0.2296	91.8%	90.5%	91.1%

Table 5.2: LSTM Model Performance

5.2.3 GRU Model Result Analysis:

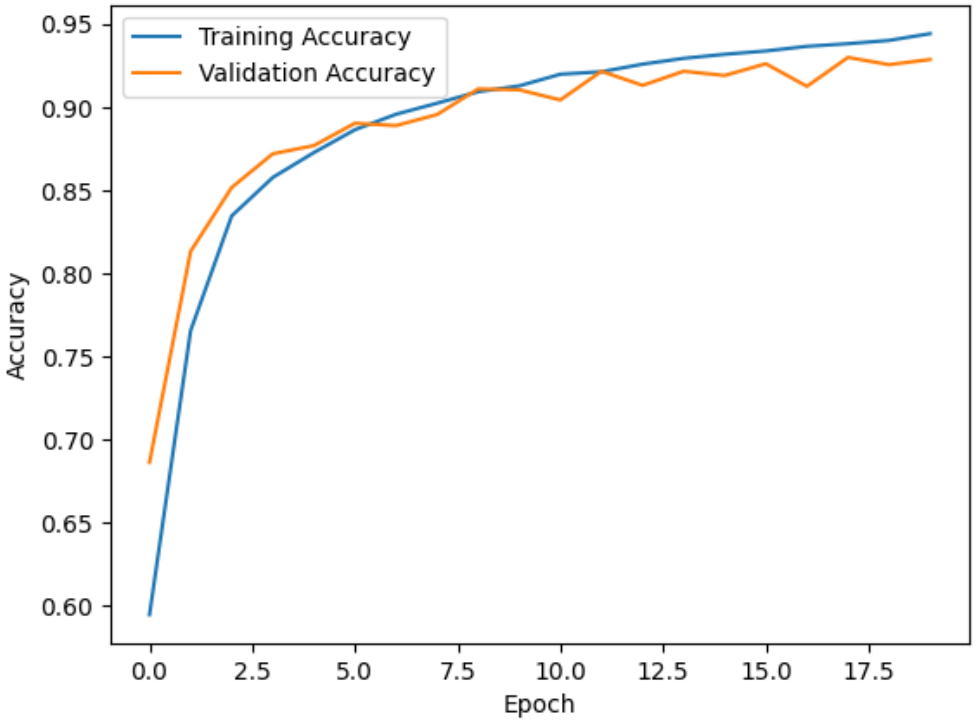

Figure 5.5: GRU Model Training & Validation Accuracy

The accuracy graph (Fig: 5.5) shows a clear improvement during training. The training accuracy hits about 94.5%, while the validation accuracy is just a tad lower at around 92.9%. This small difference indicates that the model is doing a great job of generalizing without any significant overfitting.
Looking at the confusion matrix:

- Class 0: Predicted correctly 4776 times, only 189 misclassifications
- Class 1: Predicted correctly 3450 times, only 413 misclassifications

From the training logs (Tab: 5.3), we can see that the loss is impressively low for both training (around 0.168) and validation (around 0.2118), which suggests that the model is performing strongly and consistently. This indicates excellent performance across both classes, with significantly fewer errors compared to earlier models.

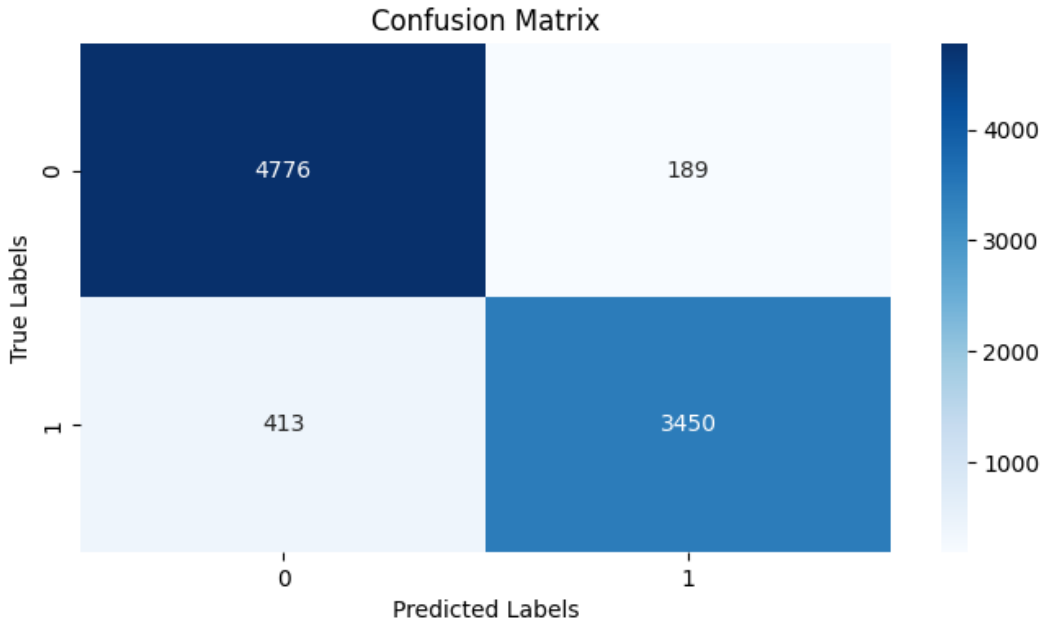

Figure 5.6: GRU Model Confusion Matrix

To sum it up, the GRU model is doing exceptionally well, boasting over 94% training accuracy and about 92.9% validation accuracy. The low loss and accurate predictions show a nice balance across both classes. The confusion matrix (Fig: 5.6) backs this up, revealing that the model is making very few errors. All in all, this is a highly dependable and effective model with great generalization capabilities.

Metric	Training Accuracy	Validation Accuracy	Validation Loss	Precision	Recall	F1-Score
GRU	94.5%	92.88%	0.2118	92.6%	91.3%	91.9%

Table 5.3: GRU Model Performance

5.2.4 BiLSTM Model Result Analysis:

The accuracy graph (Fig: 5.7) shows that both training and validation accuracy increase quickly and reach high values. The model achieves an impressive training accuracy of about 97.1% and a validation accuracy of 96.6%. The lines are closely aligned, indicating that the model is learning effectively without falling into the trap of overfitting.

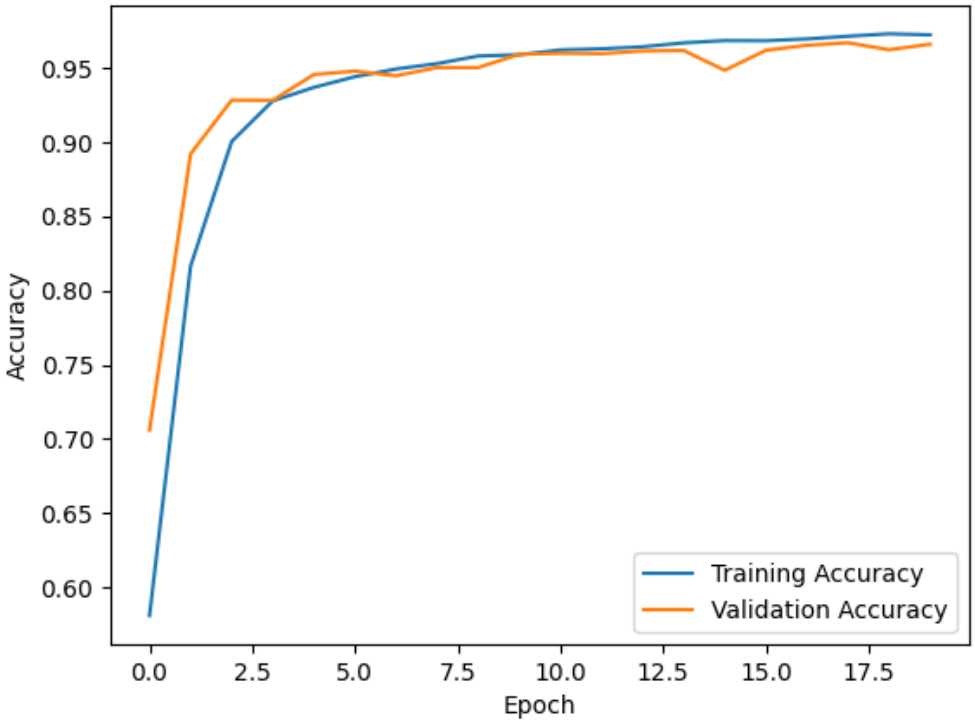

Figure 5.7: BiLSTM Model Training & Validation Accuracy

From the confusion matrix:

- Class 0: Predicted correctly 4,824 times, only 141 wrong
- Class 1: Predicted correctly 3,704 times, only 159 wrong

This means the model has very high accuracy and very few mistakes for both classes (Tab: 5.4). The loss values are very low at the end: around 0.1121 for training and 0.1237 for validation. This indicates that the predictions align closely with the actual values.

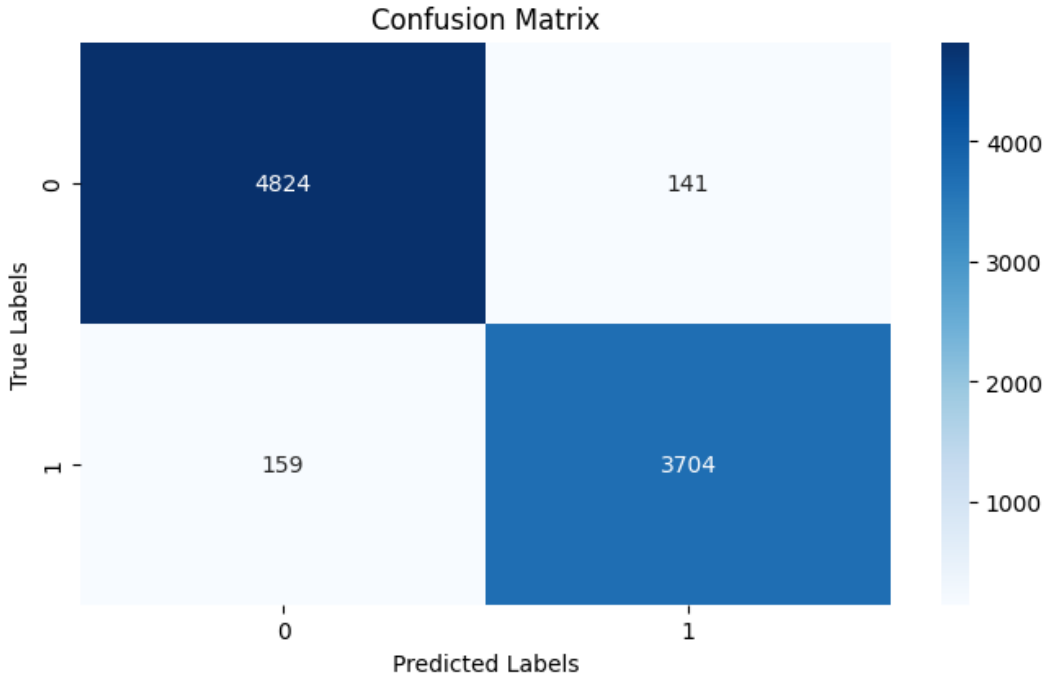

Figure 5.8: BiLSTM Model Confusion Matrix

To sum it up, the BiLSTM model is performing exceptionally well, boasting over 97% accuracy during training and 96.6% during validation. The loss is impressively low, and the confusion matrix (Fig: 5.8) reveals almost flawless classification. This model is not only accurate and balanced but also adapts well to new data, making it a robust and trustworthy choice.

Metric	Training Accuracy	Validation Accuracy	Validation Loss	Precision	Recall	F1-Score
BiLSTM	97.1%	96.6%	0.1237	96.7%	95.9%	96.3%

Table 5.4: BiLSTM Model Performance

5.3 Model Performance Comparison

The performance of different deep learning models was evaluated across various metrics. Notably, the CNN model was trained using decimal normalized spectrogram features (Tab: 5.5), whereas the LSTM, GRU, and BiLSTM models (Tab: 5.6) were trained using MFCC (Mel-Frequency Cepstral Coefficients) features.

Metric	**CNN**
Training Accuracy	82.6%
Validation Accuracy	85.6%
Validation Loss	0.3854
Precision	86.4%
Recall	79.2%
F1-Score	82.6%
Equal Error Rate (EER)	∼15.1%
Correct Class 0	4,398
Correct Class 1	3,122
Total Errors	1,309
Learning Stability	Moderate
Training Speed	Fast
Sequential Strength	Weak

Table 5.5: CNN Performance Metrics

The CNN (Convolutional Neural Network), which typically does well in detecting fixed spatial patterns, achieved 82.6% training accuracy and 85.6% validation accuracy. Though the scores are impressive, they were the lowest among the test models. CNN performed poorly when dealing with ordered sequences of information. Thus, it did worse when it came to correct labelling of data, especially if the data was of a time-series nature. Due to these shortcomings, CNN was ranked last in this task and fared poorly when dealing with sequential data.

Metric	**LSTM**	**GRU**	**BiLSTM**
Training Accuracy	92.2%	94.5%	97.1%
Validation Accuracy	91.97%	92.88%	96.6%
Validation Loss	0.2296	0.2118	0.1237
Precision	91.8%	92.6%	96.7%
Recall	90.5%	91.3%	95.9%
F1-Score	91.1%	91.9%	96.3%
Equal Error Rate (EER)	∼8.9%	∼7.2%	∼3.4%
Correct Class 0	4,541	4,776	4,824
Correct Class 1	3,567	3,450	3,704
Total Errors	721	602	300
Learning Stability	Good	Good	Excellent
Training Speed	Slow	Moderate	Slow
Sequential Strength	Strong	Strong	Very Strong

Table 5.6: LSTM, GRU, and BiLSTM Performance Metrics

The LSTM (Long Short-Term Memory) model performed much better in acquiring temporal relationships in data with 92.2% training accuracy and 91.97% validation accuracy. LSTM is especially very good at being aware of the relationship between entities in a sequence and is specifically well-equipped for such purposes. It performed fewer errors and better in loss than CNN. This made LSTM the choice in operations where the sequence and timing of the data mattered. It was still not as fast as the GRU but provided excellent performance in handling sequential data.

The GRU (Gated Recurrent Unit) model provided a good balance between speed and performance. It had 94.5% training accuracy and 92.88% validation accuracy, which was slightly better than LSTM in training accuracy. GRU is similar to LSTM but less computationally expensive, therefore faster and more efficient. GRU also did well with sequential data but not quite as well as the best that came next, as a good compromise middle-of-the-road option. The GRU model did fairly well without much loss in precision, therefore an economical solution for those applications where speed is important.

The BiLSTM (Bidirectional LSTM) was the winner. It achieved the best accuracy of all the models with 97.1% training accuracy and 96.6% validation accuracy. The BiLSTM outperformed other models by scanning the data both from forward and backward directions so that it could recognize patterns more effectively. The bidirectional aspect enabled it to learn sequence contexts more effectively, and therefore the predictions were more precise and the confusion matrix was well-balanced. BiLSTM also recorded the best loss values, which also testify to its best performance.

Chapter 6

Conclusion

Artificial intelligence (AI), specifically deepfake technology, is developing at a fast rate and, as a result, has raised concerns regarding ethics, security, and privacy. In this study, we sought to detect deepfake speech and compare the accuracy and efficiency of different speech recognition models—CNN, LSTM, GRU, and BiLSTM. After comparing the models, we determined that each model had some strengths but their performance differed. BiLSTM performed the best with the highest accuracy and fewest mistakes and the best generalization capability. The CNN performed the worst with the lowest accuracy and the most errors. GRU provided a good performance-computation efficiency trade-off and was appropriate for sequence processing, but computationally intensive and more complex. In short, BiLSTM is optimum when accuracy and sequence understanding are critical, and CNN is more suitable for basic tasks. This study highlights the necessity of robust action in avoiding the malicious use of deepfake voice technology so that faith in electronic communications is upheld.

In the future, this work can be improved by testing the models on different languages and accents to check their reliability. Advanced models like Transformers could be used for better accuracy, and combining both voice and video might improve deepfake detection. Making the models smaller and faster would help run them on mobile or low-power devices for real-time use. It's also important to make the models explainable so users understand why a voice was flagged as fake. However, there are some limitations. The dataset may not include all types of voices, accents, or background noises, which can affect accuracy. Models like LSTM and BiLSTM also need a lot of time and resources to train. Lastly, while the models detect fake voices well, they may still struggle with newer, more advanced deepfake techniques, and their decision-making process remains hard to interpret.

Bibliography

[1] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[2] T. Chen, A. Kumar, P. Nagarsheth, G. Sivaraman, and E. Khoury, "Generalization of audio deepfake detection.," in *Odyssey*, 2020, pp. 132–137.

[3] R. Wijethunga, D. Matheesha, A. Al Noman, K. De Silva, M. Tissera, and L. Rupasinghe, "Deepfake audio detection: A deep learning based solution for group conversations," in *2020 2nd International conference on advancements in computing (ICAC)*, IEEE, vol. 1, 2020, pp. 192–197.

[4] H. Dhamyal, A. Ali, I. A. Qazi, and A. A. Raza, "Fake audio detection in resource-constrained settings using microfeatures.," in *Interspeech*, 2021, pp. 4149–4153.

[5] E. Conti, D. Salvi, C. Borrelli, *et al.*, "Deepfake speech detection through emotion recognition: A semantic approach," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022, pp. 8962–8966.

[6] 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.

[7] S. Mihalache and D. Burileanu, "Using voice activity detection and deep neural networks with hybrid speech feature extraction for deceptive speech detection," *Sensors*, vol. 22, no. 3, p. 1228, 2022.

[8] N. M. Müller, F. Dieckmann, and J. Williams, "Attacker attribution of audio deepfakes," *arXiv preprint arXiv:2203.15563*, 2022.

[9] F. G, S. Kiruthika, M. Malar, and T. Nivethini, "Deepfake audio detection model based on mel spectrogram using convolutional neural network," *International Journal of Creative Research Thoughts (IJCRT)*, 2023.

[10] M. Mcuba, A. Singh, R. A. Ikuesan, and H. Venter, "The effect of deep learning methods on deepfake audio detection for digital investigation," *Procedia Computer Science*, vol. 219, pp. 211–219, 2023.

[11] J. Yi, C. Wang, J. Tao, X. Zhang, C. Y. Zhang, and Y. Zhao, "Audio deepfake detection: A survey," *arXiv preprint arXiv:2308.14970*, 2023.

[12] G. Channing, J. Sock, R. Clark, P. Torr, and C. S. de Witt, "Toward robust real-world audio deepfake detection: Closing the explainability gap," *arXiv preprint arXiv:2410.07436*, 2024.

[13] A. Firc, K. Malinka, and P. Hanáček, "Deepfake speech detection: A spectrogram analysis," in *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*, 2024, pp. 1312–1320.

[14] T. Kapoor, A. Gaur, T. Rajora, Y. Prashar, and J. Parashar, "Deepfake audio detection system," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 12, no. 5, 2024.

[15] N. V. Kulangareth, J. Kaufman, J. Oreskovic, and Y. Fossat, "Investigation of deepfake voice detection using speech pause patterns: Algorithm development and validation," *JMIR biomedical engineering*, vol. 9, e56245, 2024.

[16] N. Kumar and A. Kundu, "Securevision: Advanced cybersecurity deepfake detection with big data analytics," *Sensors*, vol. 24, no. 19, p. 6300, 2024.

[17] K. Malinka, A. Firc, M. Šalko, D. Prudkỳ, K. Radačovská, and P. Hanáček, "Comprehensive multiparametric analysis of human deepfake speech recognition," *EURASIP Journal on Image and Video Processing*, vol. 2024, no. 1, p. 24, 2024.

[18] V. K. Sharma, R. Garg, and Q. Caudron, "A systematic literature review on deepfake detection techniques," *Multimedia Tools and Applications*, pp. 1–43, 2024.

[19] J. Yi, C. Wang, J. Tao, *et al.*, "Scenefake: An initial dataset and benchmarks for scene fake audio detection," *Journal of Pattern Recognition*, 2024, https://arxiv.org/abs/2211.06073v2.

[20] A. S. Al-Shamayleh, H. Riasat, A. S. Alluhaidan, A. Raza, S. A. El-Rahman, and D. S. AbdElminaam, "Novel transfer learning based acoustic feature engineering for scene fake audio detection," *Scientific Reports*, vol. 15, no. 1, 2025. DOI: 10.1038/s41598-025-93032-2.