Internship Report

On

Audio Deepfake Detection

At

Centre for Artificial Intelligence and Robotics (CAIR)
DRDO Complex, CV Raman Nagar
Bengaluru-560093

Submitted By
Samruddhi Kale

Under The Guidance of
Shri. Bilal Shah (Sc 'F')

Acknowledgment

I would like to express my deepest gratitude to Mr. Bilal Shah, Scientist 'F' at CAIR, DRDO, for providing me with the opportunity to work on this challenging and intellectually stimulating project on audio deepfake detection. Their expert guidance, insightful feedback, and unwavering support throughout the internship have been invaluable in shaping my understanding and approach to the problem domain.

I extend my heartfelt thanks to my academic mentor, Dr.G.V.Patil, Head of Department CSE-DS, DYPCET, whose continuous encouragement and mentorship have been a pillar of strength throughout my academic journey. Their guidance has been instrumental in helping me apply my theoretical knowledge to real-world applications.

Lastly, I would like to thank CAIR, DRDO, for providing me with this incredible opportunity and for the resources and environment that facilitated my growth, both professionally and personally.

Abstract

This report outlines the development and implementation of an audio deepfake detection system, which was the focus of my research internship at CAIR, DRDO. The objective of this project was to explore and implement machine learning-based approaches for detecting spoofed audio, which is increasingly relevant in security-sensitive applications such as communication systems and authentication. The dataset used in this study consisted of the Logical Access subset of the ASV Spoof 2019 challenge, containing bonafide and spoofed audio samples. To extract meaningful features from the audio data, Mel-Frequency Cepstral Coefficients (MFCC) were utilised, which provided a compact representation of the audio signal. Several models were trained for classification, including Support Vector Machines (SVM), Custom EfficientNet-B0, and ECAPA-TDNN. Among these, the ECAPA-TDNN model, fine-tuned through transfer learning, demonstrated the highest performance, achieving a validation accuracy of 93%. The project also involved the development of a user-friendly interface using Streamlit, allowing users to easily upload audio files and predict whether they are authentic or manipulated. This system, with its high accuracy and usability, provides a potential solution for enhancing the security of voice-based systems and services.

Contents

1. About the Organisation

The **Centre for Artificial Intelligence and Robotics (CAIR)** is a premier laboratory under the Defence Research and Development Organisation (DRDO), dedicated to pioneering advancements in the areas of artificial intelligence, robotics, command and control, and information and communication security. CAIR focuses on developing mission-critical products for secure communication and information management systems on the battlefield, contributing significantly to the nation's defence capabilities.

CAIR achieves its objectives through:

- Innovating and developing technologies to enhance the effectiveness and robustness of battle space information systems.
- Creating value by designing systems that deliver assured performance in constrained and disadvantaged battle space environments.
- Anticipating and resolving emerging cybersecurity challenges proactively.
- Leveraging advancements in cognitive and artificial intelligence systems to increase autonomy in unmanned systems.
- Driving national debate on critical technology policies for national security and self-sufficiency.

Vision

"Adding value to information. Enabling battle space dominance."

Mission

"To add value to information by delivering dependable information systems to the Defence services for battle space dominance. This is achieved by developing domain and technologies that ensure relevance, security, safety, resiliency, survivability, and trustworthiness, enabling their use in mission-critical applications with guarantees of assured performance."

CAIR's unwavering commitment to research and development places it at the forefront of innovation, contributing to the country's defence and technological self-reliance.

2. Introduction

In recent years, the emergence of synthetic media, particularly **audio deepfakes**, has raised critical concerns about the security and authenticity of audio content. Deepfake audio can imitate human voices with alarming accuracy, posing significant threats in areas such as cybersecurity, information integrity, and national defence. Detecting these manipulations has become a pressing challenge, motivating advancements in machine learning-driven solutions.

During my internship at **CAIR, DRDO**, I undertook a project to develop a robust **Audio Deepfake Detection System**. The primary objective was to build a machine learning-based framework capable of accurately distinguishing between authentic (bonafide) and manipulated (spoof) audio recordings. My work involved multiple stages:

1. **Dataset Analysis and Preprocessing**:
   I worked with a dataset comprising **22,983 audio files** organised into training, validation, and testing sets. The dataset included both bonafide and spoofed samples labeled in CSV files. I extracted meaningful audio features such as **Mel Frequency Cepstral Coefficients (MFCCs)** directly from raw audio to serve as input for the models.

2. **Exploration of Classical and Neural Network Approaches**:
   I began by implementing classical machine learning models like **Support Vector Machines (SVMs)** for initial performance benchmarks. Subsequently, I explored **Convolutional Neural Networks (CNNs)** to enhance feature extraction and classification, given their proven effectiveness in handling audio and image-like data.

3. **Leveraging Transfer Learning**:
   To further improve detection accuracy, I incorporated transfer learning techniques using the **ECAPA-TDNN** pre-trained model. This advanced architecture, originally designed for speaker verification, was fine-tuned to detect deepfake characteristics in audio samples.

4. **Model Training and Evaluation**:
   Using frameworks like **PyTorch**, I implemented the data pipelines, defined the model architectures, and conducted training on the preprocessed datasets. The models were evaluated based on their ability to generalise across diverse audio manipulations, ensuring their robustness in real-world applications.

This project not only sharpened my skills in audio signal processing and advanced machine learning techniques but also provided me with the opportunity to contribute to an area of significant national importance. The proposed system serves as a step toward mitigating the risks posed by audio deepfakes, particularly in sensitive domains like defence and intelligence.

3. Understanding Audio Deepfakes

Audio deepfakes refer to synthetic audio content generated using advanced machine learning models that mimic human speech with remarkable precision. These deepfakes are created using techniques such as text-to-speech synthesis and voice conversion, leveraging generative adversarial networks (GANs) or autoregressive models. While these innovations have legitimate applications in entertainment, accessibility, and virtual assistants, their misuse poses significant risks.

Types of Audio Deepfakes

1. **Text-to-Speech (TTS) Deepfakes**:
   These are generated by models trained to produce speech from text input. Given enough training data, such models can replicate the tone, pitch, and speaking style of a specific individual.

2. **Voice Conversion Deepfakes**:
   These involve altering one person's voice to sound like another. Using source and target voice samples, machine learning models can create audio that mimics the target speaker's voice characteristics.

Threats Posed by Audio Deepfakes

The ability of audio deepfakes to replicate human voices accurately introduces various security and ethical challenges:

- **Cybersecurity Risks**: Fraudulent activities like voice phishing can exploit deepfake technology to impersonate trusted individuals for financial gain or sensitive information theft.

- **Misinformation and Propaganda**: Manipulated audio clips can be used to spread false information, tarnishing reputations or inciting unrest.

- **Legal and Identity Risks**: Unauthorised voice replication can lead to legal disputes and privacy violations, especially in domains like forensics and surveillance.

- **National Security Concerns**: In defence and intelligence, audio deepfakes can disrupt communications, compromise security systems, or spread disinformation.

Addressing the Challenges

Given these threats, detecting audio deepfakes is imperative. A robust **Audio Deepfake Detection System** requires:

- **Feature Extraction**: Identifying unique audio patterns, such as inconsistencies in frequency, pitch, or temporal structure, which are common in synthesised audio.

- **Machine Learning Approaches**: Employing algorithms like Support Vector Machines (SVMs) and Convolutional Neural Networks (CNNs) to classify audio as bonafide or spoof.

- **Transfer Learning**: Leveraging pre-trained models, such as **ECAPA-TDNN**, that excel in recognising speaker-specific patterns to detect anomalies indicative of deepfake audio.

In my project at **CAIR, DRDO**, these challenges and solutions formed the foundation of my work. By analysing various types of audio manipulations and their associated threats, I designed a detection framework capable of identifying deepfake characteristics. The project explored state-of-the-art machine learning techniques, emphasising robustness and applicability in real-world defence scenarios.

4. Literature Survey

To effectively address the problem of audio deepfake detection, several methodologies and models have been explored in the literature. This section highlights key approaches, their performance metrics, and the challenges they face, providing a foundation for understanding the methodology adopted in this project.

1. Quadratic Support Vector Machine (Q-SVM)

- **Developed by**: Kumar-Singh and Singh
- **Performance**: Achieved an accuracy of 97.56% with a misclassification rate of 2.43%.
- **Comparison**: Outperformed other machine learning (ML) methods, such as Linear Discriminant, Linear SVM, and KNN, demonstrating its robustness in distinguishing genuine and fake audio.

Key Insight: Q-SVM effectively utilises quadratic decision boundaries, providing improved classification performance compared to linear models.

2. Support Vector Machine (SVM) with Random Forest (RF)

- **Developed by**: Borrelli et al.
- **Feature Used**: Short-Term Long-Term (STLT) features, capturing temporal and spectral audio characteristics.
- **Performance**: The SVM model demonstrated 71% better performance compared to Random Forest, showcasing the importance of sophisticated boundary-based classification techniques.

Key Insight: While SVM excelled in this study, the reliance on manual feature engineering and extensive preprocessing remained a limitation.

Challenges with Traditional ML Models

Traditional ML models, such as SVM and Random Forest, rely heavily on **manual feature extraction and preprocessing**. This dependency often:

- Is time-consuming, requiring domain expertise to identify relevant features.
- Introduces inconsistencies due to variations in feature extraction techniques.

These limitations paved the way for deep learning models, which offer automated feature extraction and better scalability.

4. Convolutional Neural Network (CNN) Models

a. EfficientCNN vs. RES-EfficientCNN

- **Developed by**: Subramani and Rao
- **Dataset**: ASV Spoof Challenge 2019
- **Performance**: RES-EfficientCNN achieved a higher F1-score of 97.61% compared to EfficientCNN's 94.14%, illustrating the impact of enhanced residual learning techniques.

Key Insight: Residual CNN architectures leverage skip connections to reduce vanishing gradient issues, leading to better generalisation in complex tasks like fake audio detection.

b. CNN vs. BiLSTM

- **Study by**: Lataifeh et al.
- **Dataset**: Arabic Diversified Audio (AR-DAD)
- **Performance**:
    - CNN achieved a detection rate of 94.33%, outperforming BiLSTM but falling short compared to traditional ML methods like SVM (99%).
    - CNN models efficiently automate feature extraction but require preprocessing audio into spectrograms or 2D visual representations, adding computational overhead.

Key Insight: While CNNs automate feature extraction and excel in capturing spurious correlations, their preprocessing requirements can limit real-time applicability.

Deep Learning (DL) Models: Challenges and Advancements

Deep learning, particularly CNN-based approaches, has significantly advanced the field of audio deepfake detection. Key advantages include:

- Automated feature extraction, eliminating the need for manual preprocessing.

- Superior performance in detecting nuanced differences between bonafide and spoofed audio.

However, challenges such as **overfitting**, dependency on large datasets, and high computational requirements still persist, highlighting the need for balanced approaches.

The reviewed studies underline the evolution of audio deepfake detection methodologies, from traditional ML models with manual feature extraction to DL models with automated processes. While traditional models like SVM continue to demonstrate high accuracy in certain scenarios, DL models such as CNN have shown promise in generalising across diverse datasets. This survey provides a solid foundation for selecting appropriate models and techniques for this project.

5. Methodology

5.1 Dataset Description

The dataset used for the audio deepfake detection project is sourced from publicly available datasets focused on spoof detection on Kaggle. The primary dataset used is **ASV Spoof 2019 Dataset**, which is specifically designed for evaluating spoofing countermeasures and automatic speaker verification. It contains both **bonafide** (genuine) and **spoof** audio files, making it ideal for training and testing spoof detection models.

The ASV Spoof 2019 database encompasses two partitions for the assessment of logical access (LA) and physical access (PA) scenarios. Both are derived from the VCTK base corpus which includes speech data captured from 107 speakers (46 males, 61 females). Both LA and PA databases are themselves partitioned into three datasets, namely training, development and evaluation which comprise the speech from 20 (8 male, 12 female), 10 (4 male, 6 female) and 48 (21 male, 27 female) speakers respectively. The three partitions are disjoint in terms of speakers, and the recording conditions for all source data are identical.

While the training and development sets contain spoofing attacks generated with the same algorithms/conditions (designated as known attacks), the evaluation set also contains attacks generated with different algorithms/conditions (designated as unknown attacks). Reliable spoofing detection performance therefore calls for systems that generalise well to previously-unseen spoofing attacks.

The dataset used in this project is the Logical Access (LA) subset from the ASV Spoof 2019 challenge. This dataset is specifically designed for evaluating the effectiveness of spoofing countermeasures and includes both genuine and spoofed audio samples. The LA dataset contains approximately 122,000 audio files, which are categorised into two classes: bonafide (genuine) and spoofed (fake). Each file is associated with a specific label in the protocol file provided with the dataset.

Each audio file is in **.flac** format, and accompanying CSV label files indicate whether each sample is bonafide or spoofed. The primary feature extraction for the dataset is based on **MFCC (Mel Frequency Cepstral Coefficients)**, which are used to extract key audio features necessary for distinguishing between genuine and spoofed speech.

5. 2 Using Support Vector Machine (SVM)

Objective:
The primary objective of this approach is to detect deepfake audio using a Support Vector Machine (SVM) model, which is well-suited for binary classification tasks. The SVM will utilise advanced feature extraction techniques to improve the accuracy of detecting spoofed audio.

Dataset Used:
We used the Logical Access (LA) subset from the ASV Spoof 2019 challenge, which contains both bonafide (genuine) and spoofed (fake) audio samples. A balanced sample was created by selecting 2,580 files each from the bonafide and spoof categories, ensuring an equal representation of both classes.

Data Preprocessing:

Label Mapping: The bonafide samples were labeled as 0, and the spoofed samples were labeled as 1.

Feature Extraction: To detect deepfake audio effectively, various features were extracted from the audio files.

The features used include:

- **MFCCs (Mel-Frequency Cepstral Coefficients)**: Capture the short-term power spectrum of audio, representing the phonetic content.
- **Chroma Features**: Reflect the energy distribution across the 12 distinct pitch classes.
- **Spectral Contrast**: Measures the difference in amplitude between peaks and valleys in the sound spectrum.
- **Tonnetz (Tonality)**: Encodes tonal characteristics like harmony and pitch.

These features were extracted using Python's `librosa` library and were used to create feature vectors that serve as input to the SVM model.

Feature Normalisation

To ensure that each feature contributes equally to the model, feature normalisation was applied using StandardScaler. This process transformed the features to have a mean of 0 and a standard deviation of 1, which improved the performance and stability of the SVM model.

Model Training

- **Model Choice**

The Support Vector Machine (SVM) model was chosen due to its effectiveness in handling high-dimensional spaces and its ability to find an optimal hyperplane that maximises the margin between different classes. The SVM is particularly well-suited for binary classification problems like this one.

- **Hyperparameter Tuning**

Hyperparameter tuning was performed using Grid Search, which systematically evaluates combinations of parameters such as C (regularisation), gamma (kernel coefficient), and kernel (type of SVM kernel). The optimal parameters were selected based on cross-validation performance.

- **Handling Class Imbalance**

Despite the balanced sample used in this project, the original dataset exhibited class imbalance. To address this, the SMOTE (Synthetic Minority Over-sampling Technique) was employed to generate synthetic samples for the minority class during model training. This technique helped improve the model's ability to generalise to unseen data.

Model Evaluation

Cross-Validation

Cross-validation was used to assess the model's performance, where the dataset was divided into several folds, and the model was trained and tested on each fold. The mean accuracy score across all folds was calculated, providing an unbiased estimate of the model's performance.

Final Model Performance

The final SVM model achieved the following performance metrics:

- **Accuracy**: 92.15%
- **Precision**: 94%
- **Recall**: 90%
- **F1-Score**: 92%

These metrics indicate the model's effectiveness in distinguishing between bonafide and spoofed audio. The confusion matrix further confirmed the model's robustness, showing a high true positive rate for both classes.

5. 3 Using Custom EfficientNet-B0

Objective:

This approach aims to detect deepfake audio using the Custom EfficientNet-B0 model. Initially, a basic CNN architecture was explored, but due to suboptimal performance, the project progressed to using a more advanced architecture, **Custom EfficientNet-B0**, for better detection accuracy.

Dataset Used:

Our original dataset had significantly more spoof files (22,800) compared to bonafide files (2,580). This imbalance could lead to biased model training, where the model might become overly sensitive to the majority class (spoof) and underperform on the minority class (bonafide). By augmenting the bonafide files, we increased their number from 2,580 to 7,422. Using data augmentation, we improved our dataset's balance and size, which is expected to enhance the performance and reliability of our CNN model in detecting deepfake audio.

Specifically, **14922 Mel-spectrograms** of the audio files were generated to serve as inputs to the deep learning model.

- **Spectrogram resolution:** Original dimensions of the spectrograms are **904 x 370** pixels.
- **Total number of files:** Around **14922** spectrogram images.
- **File format:** .png for the spectrogram images, .flac for audio files.

Initial CNN Approach:

The initial approach for this deepfake audio detection project involved using a basic **CNN** architecture to classify spectrogram images as real or spoofed audio. While the model showed some potential, it encountered several challenges during training and evaluation, particularly on the test set. After training, the model was evaluated on the test set. The accuracy obtained on the test data was **50.26%**, which is close to random guessing for a binary classification problem (where 50% accuracy could be achieved without learning any patterns from the data). Given the performance results, it became clear that the initial CNN approach was insufficient for the task, as it barely outperformed random guessing. Thus, an alternative approach was needed.

Switching to Custom EfficientNet-B0:

In light of the suboptimal performance of the basic CNN, a more sophisticated model architecture was adopted **Custom EfficientNet-B0.** EfficientNet's strength lies in its balanced scaling of model dimensions (depth, width, resolution), allowing for better utilisation of computational resources. By leveraging pre-trained EfficientNet-B0 weights, the model could benefit from learning representations already trained on large datasets (ImageNet), which drastically improved the performance compared to training from scratch.

The performance of the **Custom EfficientNet-B0** model on the test set can be summarised as follows:

- **Improved Results with Custom EfficientNet-B0:** The switch to the Custom EfficientNet-B0 architecture significantly improved the model's accuracy compared to the initial CNN approach. After training, the model achieved an accuracy of 99.85%, which was a substantial improvement from the accuracy achieved by the basic CNN.

Testing on Unknown Data

Additionally, we tested the trained model on **unknown data**—audio samples that were not part of the training or validation datasets. This scenario simulates real-world conditions where the model encounters data it has never seen before. Remarkably, the model achieved an accuracy of 81% on this unknown data, showing that it is capable of generalising well to unseen examples. This performance demonstrates the robustness of the Custom EfficientNet-B0 model, indicating that it is not overfitting to the training data and can adapt to new, real-world audio data with a reasonable degree of accuracy.

The Custom EfficientNet-B0 approach demonstrates significant potential for audio deepfake detection. With an accuracy of 99% on known data and 81% on unknown data, this model shows both high performance and generalisation ability. The efficient architecture, combined with the use of MFCC features, contributes to its effectiveness in distinguishing real from fake audio, which is crucial for combating deepfake threats in the realm of voice and audio security.

5. 4 Using ECAPA-TDNN

Objective: This approach utilises the ECAPA-TDNN model, a specialised neural network for audio processing, to detect audio deepfakes using transfer learning.

Dataset Used: The dataset consists of bonafide and spoofed audio samples from the ASV Spoof 2019 challenge, divided into three subsets: Train (18,386 files), Dev (2,298 files), and Eval (2,299 files). Each subset includes labels indicating whether the audio is real or spoofed.

Preprocessing:

- **Resampling**: Audio files were resampled to 16 kHz.
- **Padding/Truncating**: The audio files were standardised to 3-second durations to match the input requirements of ECAPA-TDNN.
- **File Conversion**: Audio files in .flac format were converted to .wav format.

Feature Extraction:
MFCCs were extracted from the audio files using the `librosa` library, as they capture important frequency characteristics that are critical for detecting audio manipulations.

Model Architecture: The ECAPA-TDNN (Enhanced Contextualised Acoustic-Phonetic TDNN) is a state-of-the-art model designed specifically for tasks such as speaker verification and audio deepfake detection. It builds on the TDNN (Time-Delay Neural Network) architecture but incorporates several enhancements for better performance in sequence-based audio tasks.

Key features of the model:

- **Input Features**: MFCCs to represent audio in the spectral domain.
- **Embedding Extraction**: Incorporates Res2Net modules and SE-Block to enhance feature representation.
- **Classifier**: Fully connected layers followed by a softmax activation for binary classification.

Training:
The model uses **transfer learning** to fine-tune on a specific task such as audio deepfake detection. Pre-trained models from SpeechBrain are adapted using the ASV Spoof dataset, which includes real (bonafide) and spoofed (manipulated) speech samples. The training process involves updating the model's weights using back

propagation and Adam optimiser to minimise the cross-entropy loss, which quantifies the difference between the model's predicted and actual labels.

The model was trained on the **Train** set and validated using the **Dev** set:

- **Loss Function**: Cross-entropy loss.
- **Optimiser**: Adam optimiser with a learning rate scheduler.
- **Batch Size**: 8
- **Epochs**: 10

Results

The ECAPA-TDNN model achieved the following:

- **Training Accuracy**: ~ 95%
- **Validation Accuracy**: ~ 93%

This demonstrates the feasibility of using the ECAPA-TDNN model for detecting audio deepfakes. The results highlight the effectiveness of transfer learning and advanced neural architectures in addressing emerging challenges in audio security. The system is robust and can be extended to real-world applications, including fraud detection and secure communications.

6. User Interface for Audio Deepfake Detection

The Streamlit app provides an intuitive and easy-to-use interface for uploading audio files and predicting whether they are bonafide (real) or deepfake (manipulated). The model, based on ECAPA-TDNN for feature extraction and a custom-built neural network classifier, processes the audio and outputs the prediction with associated confidence.

Key Features:

1. **Audio File Upload**: Users can upload audio files in various formats, such as WAV, MP3, or FLAC.
2. **Sample Dataset Download**: A button is provided to allow users to download a sample dataset for testing purposes.
3. **Deepfake Detection**: The app processes the uploaded audio using a trained model and provides a prediction of whether the audio is real or fake.
4. **Prediction Confidence**: Along with the prediction, the app shows the confidence score, indicating the likelihood of the audio being a deepfake or bonafide.

How It Works:

1. **Preprocessing the Audio**: Upon file upload, the audio is resampled to 16kHz and trimmed or padded to a 3-second duration, standardising the input for the model.
2. **Feature Extraction**: The **ECAPA-TDNN model** from SpeechBrain is used to extract relevant features from the audio file, providing robust embeddings for classification.
3. **Model Classification**: A custom deepfake detection model (built using PyTorch) then classifies the audio as either **real (bonafide)** or **deepfake** based on the extracted features.
4. **Result Display**: After processing, the app displays the prediction along with the confidence score, indicating how certain the model is about its classification.

Tech Stack Overview for UI:

1. **Streamlit** : Streamlit is the core framework used for building the web interface of the application. It provides a fast and easy way to create interactive web applications with minimal effort.

2. **PyTorch** : PyTorch is used for defining and training the deepfake detection model.

3. **SpeechBrain** : SpeechBrain is an open-source library used for speech processing tasks. The pre-trained **ECAPA-TDNN** model is used in this project for feature extraction from audio files.

4. **Librosa** : Librosa is used for audio processing tasks such as loading audio files, resampling, and normalising audio data.

5. **Soundfile** : Soundfile is used to read and write audio files in different formats, such as WAV or FLAC.

The tech stack for the UI of the Audio Deepfake Detection system is designed for simplicity, efficiency, and scalability. Streamlit serves as the framework for the interactive web interface, while PyTorch powers the deepfake detection model, and SpeechBrain, Librosa, Torchaudio, and Soundfile handle the essential audio processing tasks. This combination of technologies provides an effective solution for real-time audio analysis and manipulation detection.

This Streamlit app leverages powerful audio deepfake detection models and provides a user-friendly interface for real-time predictions. By integrating feature extraction using **ECAPA-TDNN** and classification through a custom neural network, this tool enables efficient and accurate identification of audio manipulations. The inclusion of a sample dataset download feature further enhances usability, allowing users to explore the system without requiring their own audio files initially.

7. Future Work:

- **Improving Model Accuracy**:
  Suggest further steps to improve the detection model's accuracy, such as experimenting with more complex architectures or using additional training data (e.g., from multiple audio sources or deepfake generation techniques).

- **Real-World Deployment**:
  Discuss how the system can be deployed for practical use, such as in a production environment or integrated into larger platforms for monitoring audio authenticity in communications, podcasts, or social media.

- **Cross-Language and Multi-Domain Detection**:
  Future research could involve extending the model to handle audio from multiple languages or domains (e.g., music, different accents, and noise conditions).

- **Adversarial Robustness**:
  Consider exploring the robustness of the model against adversarial attacks, where malicious actors could attempt to manipulate audio to evade detection.

- **Integration with Other Modalities**:
  Explore the possibility of integrating audio deepfake detection with other modalities, such as video deepfake detection, to create a more holistic solution.

8. Findings and Key Achievements

Findings:

- **High Accuracy**: The ECAPA-TDNN-based model achieved over 90% accuracy in detecting audio deepfakes, showcasing the effectiveness of transfer learning on the ASV Spoof dataset.
- **Impact of Preprocessing**: Resampling, padding, and MFCC feature extraction were crucial for preparing the audio data, ensuring consistent input for the model and enhancing performance.
- **Transfer Learning Success**: Using the pre-trained SpeechBrain ECAPA-TDNN model for feature extraction reduced training time and improved detection accuracy by leveraging large-scale speech datasets.
- **Real-Time Detection**: The system was able to provide immediate predictions, making it suitable for real-world applications that require quick deepfake identification.

Key Achievements:

- **End-to-End System**: The project successfully built an end-to-end audio deepfake detection system, integrating preprocessing, feature extraction, classification, and result presentation.
- **ECAPA-TDNN Application**: The model effectively utilised ECAPA-TDNN's speech recognition capabilities, achieving high detection accuracy with minimal data preprocessing.
- **User-Friendly Interface**: A Streamlit-based interface allowed users to upload audio files and receive real-time predictions, enhancing accessibility for non-technical users.
- **Practical Application**: The system's ability to detect deepfake audio with a high degree of confidence makes it a valuable tool for various applications in several domains.

9. Conclusion

The internship project successfully achieved its goal of developing a high-accuracy audio deepfake detection system. By comparing different machine learning models, it was determined that while traditional methods like SVM were effective, more advanced techniques such as Custom EfficientNet-B0 and ECAPA-TDNN yielded significantly improved results. The ECAPA-TDNN model, in particular, demonstrated strong generalisation capabilities, making it suitable for real-world application. The use of MFCC features for audio representation, combined with transfer learning, played a key role in achieving high detection accuracy.

Additionally, the development of a Streamlit-based interface allowed for seamless interaction with the system, making it practical for users in security-sensitive environments. Although the system performed well, there is still room for further improvement, particularly in terms of increasing model robustness against unseen data and expanding the training dataset. In future work, incorporating more diverse and larger datasets, as well as exploring advanced model optimisation techniques, will further enhance the system's capabilities. The insights gained from this project contribute to the ongoing efforts to mitigate the risks posed by audio deepfakes, with applications in security, fraud detection, and media verification.

Project Repository

10. References:

1. **ASVspoof 2019 Dataset**
   This is a database used for the Third Automatic Speaker Verification Spoofing and Countermeasuers Challenge, for short, ASVspoof 2019
   Link to Dataset

2. **ECAPA-TDNN Model**
   Brecht Desplanques, Jenthe Thienpondt, Kris Demuynck, "ECAPA-TDNN: Efficient Channel-Scaled TDNN for Speaker Verification," Interspeech 2020.
   Link to Paper

3. **Streamlit Documentation**
   "Streamlit: The fastest way to build custom ML tools," *Streamlit*
   Link to Streamlit Docs

4. **TensorFlow Documentation**
   "TensorFlow: An open-source machine learning framework," *TensorFlow*
   Link to TensorFlow Docs

5. **PyTorch Documentation**
   "PyTorch: An open-source machine learning library," *PyTorch*
   Link to PyTorch Docs

6. **Hugging Face Speech brain ECAPA MODEL**
   This repository provides all the necessary tools to perform speaker verification with a pre-trained ECAPA-TDNN model using SpeechBrain.
   Link to Hugging Face