WaveTruth - A Transfer Learning-Based Deep Learning Approach for Detecting Deepfake Audio

Hisham Iqbal Khokhar

March 2025

Contents

Abstract

The rapid advancement of artificial intelligence in speech synthesis has led to an increase in AI-generated voices that closely mimic human speech. This project aims to develop a system capable of distinguishing AI-generated voices from real human voices while also classifying gender. A custom dataset of 1,956 audio clips comprising of a total of around 2.7 hours of audio samples was compiled, consisting of both AI-generated and real human voices across diverse accents. Key acoustic and prosodic features were extracted for model training.

A comparative analysis of 16 machine learning and deep learning models was conducted to evaluate their effectiveness in identifying synthetic speech. Traditional models kNN and Naïve Bayes exhibited lower performance, whereas tree-based models such as Random Forest, XGBoost, and CatBoost achieved higher accuracy. Deep learning approaches, including Feedforward Neural Networks (FFNN) and Recurrent Neural Networks (RNN), outperformed traditional methods, with the Transfer Learning FFNN emerging as the most effective model, hence being selected as the model for deployment.

The selected Transfer Learning Based Feed Forward Neural Network was integrated into a web application for voice classification incorporated with a UI, providing users with an interactive tool to determine whether a voice is AI-generated or real and whether it is male or female. The results highlight the effectiveness of deep learning in detecting synthetic speech and pave the way for further improvements using larger datasets and advanced feature extraction techniques.

Keywords: Deepfake detection, AI-generated speech, machine learning, deep learning, transfer learning, voice classification.

1. INTRODUCTION

In recent years, AI-generated voices have become increasingly sophisticated, particularly in applications like media production and customer service automation. These AI voices are often generated using deep learning models, making it difficult to distinguish them from real human speech. This presents serious challenges in the fields of security, fraud detection, and content moderation. Motivated by these challenges, this project aims to develop a transfer learning-based feedforward deep learning model capable of detecting AI-generated voices and differentiating them from human speech. By using a pre-trained model and fine-tuning them for this specific task, the system seeks to improve accuracy and efficiency in identifying synthetic speech.

1.1 Background Research

The ability to detect AI-generated speech has become an important research area mainly due to advancements in text-to-speech (TTS) and voice cloning technologies. Various approaches have been proposed, ranging from spectral analysis techniques to deep learning models, each aiming to identify distinguishing characteristics of synthetic voices. However, challenges remain in ensuring robustness against increasingly sophisticated synthesis methods.

1.1.1. Literature Review

Early Methods: Spectral & Statistical Analysis

One of the first approaches to detecting AI-generated speech involved spectral and statistical analysis. AlBadawy et al. (2019) introduced bispectral analysis to identify non-linear dependencies in speech signals, which were indicative of synthetic origins. This method effectively detected early AI-generated speech, which exhibited clear spectral irregularities. However, as AI synthesis techniques improved, these anomalies became less apparent, limiting the reliability of bispectral analysis for detecting modern synthetic speech.

Similarly, early research explored traditional machine learning models trained on handcrafted features such as Mel-frequency cepstral coefficients (MFCCs), spectral sub band energies, and linear predictive coding (LPC). While these methods showed promise,

they often failed to generalize well across different synthesis models, highlighting a gap in their adaptability to evolving AI speech generation technologies.

Neural Vocoder Artifacts & Their Limitations

A more recent approach to AI-synthesized speech detection involves analyzing neural vocoder artifacts, subtle distortions left behind by vocoders used in TTS systems. Sun et al. (2023) demonstrated that these artifacts could be used to differentiate AI-generated voices from human speech, particularly for models like WaveNet and Tacotron. However, as synthesis models became more advanced, vocoder imperfections were significantly reduced, making this method less effective for detecting newer AI-generated voices.

This limitation underscores a key challenge in AI speech detection: methods that rely on fixed feature sets struggle to keep up with advancements in synthesis technology. As AI models improve, they produce speech that closely mimics natural human voices, requiring detection methods to adapt dynamically.

Deep Learning for AI-Synthesized Speech Detection

Given the limitations of traditional approaches, researchers have increasingly turned to deep learning models for AI-synthesized speech detection. Ren et al. (2024) proposed a deep learning-based method that integrates temporal and spatial audio analysis using models like Light Convolutional Neural Networks (LCNN). Their approach demonstrated improved detection accuracy by learning complex, high-dimensional speech patterns that traditional methods often missed.

Despite the success of deep learning models, generalization remains a significant challenge. Many models perform well on known datasets but struggle when tested on unseen speech synthesis techniques.

This issue highlights a critical research gap: how to develop AI-speech detection models that maintain high accuracy across a wide range of synthesis methods without frequent retraining.

1.1.2. Addressing Gaps in Existing Research

While prior research has explored various detection techniques, several challenges remain:

- **Generalization to New Speech Synthesis Models** - Many detection methods are trained on specific AI-generated datasets, making them less effective against newer synthesis techniques.
- **Handling Low-Resource Scenarios** - Deep learning models typically require large amounts of labelled data, which may not always be available for AI speech detection.
- **Computational Efficiency** - Many high-performing models are computationally expensive, limiting their real-world applicability in resource-constrained environments.

Advancements in AI-Synthesized Voice Detection: SiFSafer Framework

AI-synthesized voices have become increasingly prevalent in various applications. While their high quality enables numerous innovations, they also present significant security and trust concerns. Many studies have focused on detecting fake voices to mitigate these risks, claiming promising detection performance. However, recent research has revealed a major limitation: existing fake voice detectors often suffer from overfitting to speaker-irrelative features (SiFs), making them ineffective in real-world applications. This issue arises because current models often prioritize the differences between human and synthetic voices, rather than focusing on identifying the true characteristics of human speech.

To address this, the study by (Hai et al., 2024) introduces a novel design philosophy for detecting AI-synthesized voices. This philosophy emphasizes learning human voice features instead of focusing on synthetic voice distinctions. The authors propose a new detection framework called SiFSafer, which utilizes pre-trained speech representation models to enhance the learning of human voice feature distributions. Additionally, they introduce adapter fine-tuning to optimize detection performance.

The experimental results show that existing detectors can achieve error rates (EERs) above 20% when SiFs, such as silence segments, are removed from the ASVspoof datasets. In contrast, SiFSafer achieves an EER of less than 8%, demonstrating its robustness against SiFs and its strong resistance to existing spoofing attacks, marking it as a promising advancement for real-world voice detection systems.

1.1.3. Existing Approaches

Speech Forensics Using Machine Learning

Recent approaches in the research field of speech forensics investigate machine learning techniques for detecting synthetic speech, a growing concern in fraud and misinformation. A very recent approach by (Bhagtani, K, 2025) explores various feature extraction methods, including spectral and prosodic analysis, to capture distinguishing characteristics between human and AI-generated voices. The study evaluates machine learning models such as Support Vector Machines (SVM), Convolutional Neural Networks (CNN), and deep neural networks (DNNs) to detect speech synthesis artifacts. Additionally, it examines generalization challenges across different AI voice synthesis techniques and datasets. A key contribution of this research is the focus on domain adaptation, ensuring models remain effective against unseen AI voice generation methods. The findings emphasize the need for robust forensic tools that integrate explainable AI techniques to provide transparency in decision-making.

Detecting AI-Synthesized Speech Using Bispectral Analysis

AlBadawy et al., (2019 presents a unique approach to AI-synthesized speech detection by leveraging bispectral analysis, a signal processing technique that captures higher-order statistical dependencies in audio signals. Unlike conventional spectral analysis, bispectral features help distinguish natural speech from AI-generated voices by detecting subtle nonlinear artifacts introduced during synthesis. The study evaluates various AI speech synthesis methods, demonstrating that bispectral analysis improves detection accuracy compared to traditional machine learning and deep learning approaches. The research suggests that bispectral features provide an effective way to identify artifacts that persist across different synthesis techniques, making it a promising tool for forensic speech analysis.

Robust AI-Synthesized Speech Detection Using Feature Decomposition

Another research by Zhang et al., (2024) proposes a feature decomposition-based approach to improve the detection of AI-synthesized speech. Traditional methods focus primarily on spectral artifacts, but this study introduces two novel strategies:

- **Feature Decomposition Learning** - Separates human voice characteristics from synthesis artifacts, allowing the model to learn distinctive speech patterns.

- **Synthesizer Feature Augmentation** - Enhances training data by simulating unseen synthetic voices, improving generalization across AI-generated speech techniques.

By implementing these techniques, the model achieves better performance on unseen AI voice synthesis methods. The study evaluates various datasets and shows that the approach outperforms existing deepfake speech detection models in generalization and accuracy. The findings suggest that incorporating domain-invariant speech features allows AI-generated speech to be effectively distinguished from real voices, making this method suitable for real-world applications in security and forensic analysis.

Neural Vocoder Artifacts

Sun et al. (2023) proposed an approach that focuses on detecting neural vocoder artifacts in AI-generated speech. Vocoders are integral components of speech synthesis models, such as WaveNet and Tacotron, which are used to convert feature vectors into waveforms. These vocoders, however, often introduce subtle artifacts or distortions in the speech signal, which can be used as indicators of synthetic speech. The study identifies specific neural vocoder artifacts that are characteristic of AI-generated voices and demonstrates how these can be exploited for voice detection. This method is particularly useful for identifying synthetic speech generated by well-known vocoder-based synthesis systems. However, with the continuous improvement of neural vocoder models, newer versions may reduce these artifacts, making it more difficult for this approach to remain effective over time.

Generalization for AI-Synthesized Voice Detection

Ren et al. (2024) addressed the issue of generalization in detecting AI-synthesized voices by proposing a model that integrates both temporal and spectral patterns of speech. Their method leverages deep learning architectures like LCNN (Local Convolutional Neural Networks) to capture the dynamic features across both time and frequency domains. The key contribution of this work is its focus on enhancing the ability of AI-synthesized voice detection models to generalize across different synthesis techniques and datasets. Traditional methods often struggle with this generalization, as synthetic speech can exhibit various characteristics depending on the underlying model used (e.g., WaveNet, Tacotron, etc.). By using advanced deep learning models, Ren et al. argue that their

approach offers better adaptability and robustness, allowing it to perform well across diverse speech synthesis systems and datasets.

Despite significant progress in AI-synthesized speech detection, there are still major gaps that need to be addressed. One of the major issues is generalization, many existing models such as those using bispectral analysis (AlBadawy et al., 2019) or neural vocoder artifacts (Sun et al., 2023) work well on specific datasets but fail when tested on new AI-generated voices. This makes them unreliable in real-world scenarios, where deepfake speech keeps evolving.

Another major challenge is the lack of diverse datasets. Most studies rely on pre-existing datasets that don't fully represent the variety of AI-generated voices, especially across different accents, languages, and speaker demographics. Many studies train their models on limited datasets that do not sufficiently cover different accents, languages, and speaker demographics. For instance, while deep learning models like LCNN (Ren et al., 2024) and feature decomposition methods (Zhang et al., 2024) have improved detection accuracy, they still require extensive labelled data to maintain performance across various synthesis techniques. Without a more comprehensive dataset, detection models struggle to remain effective against newer synthesis techniques. Moreover, many datasets are imbalanced in gender representation, which can lead to biases in classification models.

There's also the issue of computational efficiency. While deep learning models have improved detection accuracy, many high-performing methods, such as deep learning-based approaches (Ren et al., 2024; Bhagtani, 2025) require high processing power and large amounts of labelled data, making real-time deployment difficult, especially in resource-limited environments.

1.2. How This Project Fills the Gap

This project aims to bridge these gaps in the following ways: -

1. Creating a diverse dataset featuring real and AI-generated voices across multiple accents, including Canadian, Australian, Indian, American, English, and various European and African accents. This ensures better generalization in detection models. This ensures better generalization for detection models.

2. Developing machine learning and deep learning models to effectively detect deepfake voices, ensuring they adapt well to new AI speech synthesis techniques educing dependence on speaker-irrelative features.
3. Building a gender classification system to analyse both human and AI-generated voices, helping to understand potential biases in AI-synthesized speech addressing biases in speech datasets and improving fairness in AI detection.
4. Evaluating multiple machine learning models to identify the best-performing approach for AI speech detection.
5. Deploying an interactive web application that provides real-time voice authenticity and gender detection, making this research applicable in real-world scenarios.

By addressing these challenges, this project contributes to the development of more adaptive, robust, and computationally efficient AI-synthesized speech detection methods providing a scalable and real-world-ready solution to counter emerging deepfake voice threats while ensuring inclusivity and fairness in detection across different speech characteristics.

1.3. Aim & Objectives

The aim of this project is to develop a voice detection system that can accurately differentiate between AI-generated (deepfake) voices and real human speech. An additional goal is to classify the gender of the speaker (male or female) regardless of whether the voice is synthetic or human. To do this, a custom dataset is built with both AI-generated and real audio samples. The AI voices will come from Canva's "AI Voiceover" text-to-speech software, which will produce both male and female voices in various accents like Canadian, Australian, Indian, American, English, and several European and African accents. For the real voices, audio from podcasts, speeches, lectures, and YouTube tutorials is gathered featuring male and female speakers.

The plan is to extract different numerical features from these audio clips to find patterns that can separate AI-generated speech from real human speech. Supervised learning techniques are used to identify these patterns and test out 15 different machine learning models, including deep learning, decision trees, KNN, Naive Bayes, and SVM, to see which one gives the highest accuracy in distinguishing between AI and real voices. Since the dataset will be relatively small, the use transfer learning technique is adopted to boost

the model's performance. The end goal is to find the best model for classifying audio as AI-generated or real, while also telling whether the voice is male or female. Finally, the model will be deployed in an interactive web app to allow real-time voice detection.

1.4. Objectives

The main objectives of this research project include:

- Building a diverse dataset featuring real and AI-generated voices across multiple accents (Canadian, Australian, Indian, American, English, and several European and African accents).
- Detecting deepfake voices using machine learning and deep learning methods.
- Classifying gender (male/female) of both AI and human speakers.
- Evaluating multiple machine learning models to identify the most effective solution.
- Deployment of an interactive web app for real-time voice authenticity and gender detection.

To achieve these objectives, the starting point would have always been collecting excellent data. For the real voices, audio from podcasts, speeches, lectures, and YouTube tutorials featuring male and female speakers is gathered and split. All these audio files are split into 5 second audio clips and normalized to a fixed decibel volume using Audacity for uniformity.

Next, these audio clips are processed, and several key features are extracted, such as Spectral Centroid Mean, Spectral Bandwidth Mean, RMS Mean, Zero-Crossing Rate Mean, Spectral Contrast Mean, Pitch Mean, Pitch Confidence Mean, Mel Spectrogram Mean, Mel Spectrogram Variance, Energy Mean, and Speech Rate. These features help identify the unique patterns that distinguish AI-generated speech from real human speech.

Once these features are extracted, a model is then trained using supervised learning techniques to spot those differences. In the process of Model selection, the performance of a range of models is tested, including KNN, Bernoulli Naive Bayes, Gaussian Naive Bayes, Decision Trees, Linear Discriminant Analysis (LDA), Logistic Regression, Gradient Boosting, Random Forest, XGBoost, CatBoost, LightGBM, SVM, Extra Trees, Feedforward Neural Networks, and RNNs to see which one gives the best accuracy in distinguishing

between AI and real voices. Since the dataset is a bit small, a transfer learning approach is used to boost performance and make sure the model works as effectively as possible.

In addition to identifying whether the voice is AI-generated or real, the project is also incorporated with gender classification into the model, so it can tell whether the voice is male or female. The final model will then be deployed in a web app, allowing users to do real-time voice detection and classification, helping them figure out whether the voice is AI-generated or real, and also whether it's male or female.

1.5. Product description

For professionals and platforms needing reliable AI-generated speech detection, WaveTruth is a Django-based web application that accurately classifies AI vs. human voices while also distinguishing speaker gender in real time. Unlike basic detection tools, WaveTruth uses a transfer learning-based Feedforward Neural Network (FNN) trained on a diverse, custom dataset, ensuring higher accuracy and broader accent coverage.

WaveTruth addresses key challenges in deepfake detection and speaker gender classification by integrating advanced audio analysis and deep learning techniques into a practical, real-world tool. WaveTruth overcomes these limitations of lack of generalization and struggle with diverse accents by using efficient deep learning architectures optimized for real-time detection within a lightweight Django framework.

The WaveTruth system is trained on a carefully curated dataset of 1,956 five-second audio clips, evenly split between AI-generated and real human speech. AI voices were sourced from various Text-To-Speech models, covering diverse accents including Canadian, Australian, Indian, British, American, African, and European. Real voices were collected from publicly available content such as podcasts, lectures, and interviews.

To ensure high detection accuracy, all audio samples were normalized to a certain audio decibel volume and processed to extract key features, including:

- **Spectral Features** including Centroid, bandwidth, contrast
- **Energy & Amplitude Features** including RMS energy, zero-crossing rate
- **Pitch Features** including Pitch mean, pitch confidence
- **Temporal & Rate Features** including Speech rate, Mel spectrogram statistics

These extracted features were used to train and evaluate multiple machine learning and deep learning models**,** with the Feedforward Neural Network (FNN) outperforming traditional ML models. Transfer Learning was applied to enhance performance despite the dataset size, allowing WaveTruth to generalize better across unseen voices.

WaveTruth is designed to support a wide range of practical applications, including:

- **Fake Audio Detection in Journalism & Social Media,** helping media outlets verify voice authenticity in interviews and reports.
- **Deepfake Threat Detection in Security**, enhancing forensic investigations and fraud detection.
- **AI-Generated Content Filtering**, assisting platforms in identifying and managing synthetic voice content.
- **Voice Biometrics & Authentication**, strengthening security systems by validating real vs. AI-generated voices.

With WaveTruth, organizations gain a fast, accurate, and scalable solution for AI speech detection and speaker classification, ensuring trust and authenticity in digital audio content**.**

1.6. Contribution

This dissertation makes several key contributions to the field of deepfake audio detection and voice classification:

Custom Dataset Creation

A dataset consisting of 1,956 audio clips (approximately 2.7 hours of AI-generated and real human speech, each lasting five seconds) was created. This dataset includes diverse accents including Chinese, British, American, Canadian, African, Australian, Indian etc. and speech types, making it a valuable resource for training and evaluating deepfake detection models.

Gender Classification in Synthetic Speech Detection

Integrated gender classification as an additional layer of analysis, demonstrating that deep learning models can accurately distinguish between male and female voices alongside detecting synthetic speech.

Evaluation of Machine Learning and Deep Learning Models

Conducted a comparative analysis of traditional machine learning models (k-Nearest Neighbors, Naïve Bayes, Random Forest, XGBoost, and LightGBM) for AI vs. human speech classification.

- Demonstrated that tree-based models outperform classical methods, achieving 75% to 90% accuracy, while kNN performed the worst (<40% accuracy).

Advancing Deep Learning for Synthetic Speech Detection

- Designed and evaluated Feedforward Neural Networks (FFNNs) and Recurrent Neural Networks (RNNs), achieving 90%+ accuracy in distinguishing real vs. AI-generated voices.
- Highlighted the effectiveness of Transfer Learning FFNN, which outperformed all models, proving its potential for detecting deepfake speech with limited training data.

Theoretical Contributions

- Provided insights into feature importance for synthetic speech detection, confirming that spectral and energy-based features (e.g., Spectral Centroid, Spectral Bandwidth, RMS Energy, and Pitch Confidence) are highly discriminative.
- Contributed to transfer learning applications in speech forensics, showing that leveraging pre-trained models enhances classification accuracy and efficiency.

2. THE DATASET

The dataset used in this study consists of 1,956 audio clips, each with a duration of 5 seconds, resulting in a total of over 2.7 hours of audio data. The dataset is equally distributed among four distinct classes, ensuring a balanced classification task. Each class contains 489 audio clips, categorized as Female AI (F_AI), Male AI (M_AI), Female Real (F_REAL), and Male Real (M_REAL).

2.1. Feature Extraction

Each audio clip is represented using a set of extracted acoustic, prosodic and other speech related features that provide meaningful insights into the characteristics of the sound. The features included in the dataset are:

Spectral Centroid (Mean) indicates the "center of mass" of the spectrum, which is linked to the perceived brightness of a sound. AI-generated voices often have a more consistent spectral distribution, while real voices exhibit more variability, making spectral centroid a useful feature for distinguishing synthetic speech from human speech. (Anon., 2024)

$$\text{centroid} = \frac{\sum_{n=0}^{N-1} f(n)x(n)}{\sum_{n=0}^{N-1} x(n)},$$

Spectral Bandwidth (Mean) measures the width of the frequency spectrum and reflects sound sharpness. AI-generated voices typically have narrower bandwidths, while human speech displays more variability, helping the model differentiate between synthetic and real voices by capturing this spectral richness. (Librosa, 2025)

$$\left(\sum_{k} S(k)(f(k) - f_c)^p\right)^{\frac{1}{p}}$$

RMS (Mean) represents the average amplitude of the audio signal, highlighting loudness variations. Real speech shows more dynamic fluctuations in RMS energy, while AI voices tend to have more uniform energy profiles, allowing the model to identify synthetic speech based on energy consistency. (Anon., 2025)

Zero-Crossing Rate (Mean) measures how often the signal changes sign, linked to noisiness. AI-generated voices tend to have fewer zero-crossings due to their smoother nature, whereas real speech has more irregularities, making this feature key for distinguishing synthetic and natural speech (Anon., 2025).

$$zcr = \frac{1}{T-1}\sum_{t=1}^{T-1} |\mathrm{sgn}[s(t)] - \mathrm{sgn}[s(t-1)]|$$

Spectral Contrast (Mean) Spectral contrast measures the difference between peaks and valleys in the spectrum, capturing timbral characteristics. Real human speech exhibits more varied spectral contrast, whereas synthetic voices tend to have smoother and more uniform spectra, aiding in the detection of AI-generated speech (So et al., 2020).

Pitch (Mean) refers to the average fundamental frequency of the speech. Real human voices naturally fluctuate in pitch, while AI-generated voices often maintain a more consistent, unnatural pitch, making pitch mean a useful feature for distinguishing between synthetic and human voices. (Anon., 2025)

Pitch Confidence (Mean) represents the reliability of detected pitch values. AI voices typically have more stable pitch values, while real voices fluctuate more due to emotional and contextual changes, enabling the model to identify synthetic speech based on pitch consistency. (Bradbury, n.d.)

Mel Spectrogram (Mean) captures the frequency distribution over time. AI-generated voices usually lack the complexity and variation seen in real human speech spectrograms, making this feature useful for distinguishing synthetic voices, especially in short audio clips. (Zhang et al.)

Mel Spectrogram Variance indicates fluctuations in frequency content. Real human speech exhibits more variance due to natural vocal changes, while AI-generated voices tend to be more consistent, allowing this feature to detect synthetic speech effectively. (Zhang et al.)

Energy (Mean) measures the average energy of the audio signal, indicating loudness. Real human voices vary in energy due to emotional context and articulation, whereas AI voices typically maintain steady energy levels, making this feature useful for detecting synthetic speech. (Librosa, 2025)

$$\sum_n |x(n)|^2$$

Speech Rate measures the speed of speech. AI-generated voices may have a more consistent or unnatural speech rate, while real human voices exhibit more variability depending on emotional expression and context, making speech rate a key distinguishing feature.

Table 1 - Feature Description

Features	Description
Spectral Centroid (Mean)	Brightness
Spectral Bandwidth (Mean)	Spread
RMS (Mean)	Loudness
Zero-Crossing Rate (Mean)	Noisiness
Spectral Contrast (Mean)	Timbre
Pitch (Mean)	Frequency
Pitch Confidence (Mean)	Reliability
Mel Spectrogram (Mean)	Frequency-Avg
Mel Spectrogram Variance	Frequency-Var
Energy (Mean)	Power
Speech Rate	Speed

2.2. Dataset Composition

Dataset composition is shown in Table 2. Each of the 1,956 clips is approximately 5 seconds long, resulting in a dataset that is well-balanced across gender and voice origin (AI vs. real).

Table 2 - Dataset Composition

Class	Number of Audio Clips	Total Class Data Duration (approx.)
AI Male Voices	489	~41 minutes
AI Female Voices	489	~41 minutes

Real Male Voices	489	~41 minutes
Real Female Voices	489	~41 minutes
Total	**1,956**	**~2.7 hours**

2.2.1. Limitations and challenges in the dataset

- Although balanced, the total number of samples is relatively small. This limitation was addressed through the use of transfer learning in model development.
- While accent diversity was prioritized, the range of unique speakers per category is limited, which may affect generalizability.
- Despite pre-processing, some real clips retained mild background noise, which could influence feature extraction and model performance.

2.3. Data Collection & Pre-processing

A custom dataset was built with both AI-generated and real audio samples. The AI voices dataset was generated and collected from Canva's "AI Voiceover" text-to-speech software, which produced both male and female voices in various accents like Canadian, Australian, Indian, American, English, and several European and African accents. Whereas the real voices audio clips were extracted from podcasts, speeches, lectures, and YouTube tutorials gathering featuring male and female speakers' voices.

Audio Normalization was done after all audio clips were gathered, to achieve the same range of volume across all audio data points. This was done by normalizing all audio clips in Audacity FFmpeg Software that normalized all audio clips to a peak amplitude of -1.0 dB.

These normalized audio clips were then organised in labelled folders for easier accessibility and feature extraction was done by running the following python script: -

```
import librosa
import numpy as np
import os
import pandas as pd
import syllapy
import speech_recognition as speech
from pydub import AudioSegment
```

```
# Path to the folder containing audio clips
audio_folder_path = 'Audio_clips_FolderPath'

# List to store feature data for all audio clips
audio_features_list = []

def mp3_to_wav(mp3_path, wav_path):
   # Load the MP3 file
   audio = AudioSegment.from_mp3(mp3_path)
   # Export as WAV
   audio.export(wav_path, format="wav")

for filename in os.listdir(audio_folder_path):
   if filename.endswith('.mp3'):
      audio_path = os.path.join(audio_folder_path, filename)

      file_size_kb = os.path.getsize(audio_path) / 1024  # Convert bytes to KB

      y, sr = librosa.load(audio_path, sr=None)

      # -------Extract features----------

      spectral_centroid = librosa.feature.spectral_centroid(y=y, sr=sr)
      spectral_centroid_mean = np.mean(spectral_centroid)

      spectral_bandwidth = librosa.feature.spectral_bandwidth(y=y, sr=sr)
      spectral_bandwidth_mean = np.mean(spectral_bandwidth)

      spectral_contrast = librosa.feature.spectral_contrast(y=y, sr=sr)
      spectral_contrast_mean = np.mean(spectral_contrast)

      rms = librosa.feature.rms(y=y)
      rms_mean = np.mean(rms)

      zero_crossing_rate = librosa.feature.zero_crossing_rate(y)
      zero_crossing_rate_mean = np.mean(zero_crossing_rate)

      pitch, pitch_confidence = librosa.piptrack(y=y, sr=sr)
      pitch_mean = np.mean(pitch[pitch > 0])  # Filter to include only non-zero pitches
      pitch_confidence_mean = np.mean(pitch_confidence)

      mel_spectrogram = librosa.feature.melspectrogram(y=y, sr=sr)
      mel_spectrogram_mean = np.mean(mel_spectrogram)
      mel_spectrogram_var = np.var(mel_spectrogram)

      energy = librosa.feature.rms(y=y)  # calculates energy (RMS) over time
      energy_mean = energy.mean()


      mp3_file = audio_path
      wav_file = "audio_clip.wav"

      # Convert MP3 to WAV
      mp3_to_wav(mp3_file, wav_file)
```

```
    recognizer = speech.Recognizer()
    # Load the audio file using SpeechRecognition
    with speech.AudioFile(wav_file) as source:
      audio = recognizer.record(source)  # Read the entire file

    try:
      transcription = recognizer.recognize_google(audio)  # Using Google Web
Speech API

      words = transcription.split()
      syllable_count = sum(syllapy.count(word) for word in words)  # Count
syllables in each word

      audio_duration = AudioSegment.from_wav(wav_file).duration_seconds  #
Duration in seconds
      speech_rate = syllable_count / audio_duration  # Syllables per second

    except speech.UnknownValueError:
      speech_rate = 0
    except speech.RequestError as e:
      speech_rate = 0


    feature_row = {
      'Filename': filename,
      'File Size (KB)': file_size_kb,
      'Spectral Centroid Mean': spectral_centroid_mean,
      'Spectral Bandwidth Mean': spectral_bandwidth_mean,
      'RMS Mean': rms_mean,
      'Zero-Crossing Rate Mean': zero_crossing_rate_mean,
      'Spectral Contrast Mean': spectral_contrast_mean,
      'Pitch Mean': pitch_mean,
      'Pitch Confidence Mean': pitch_confidence_mean,
      'Mel Spectrogram Mean': mel_spectrogram_mean,
      'Mel Spectrogram Variance': mel_spectrogram_var,
      'Energy Mean': energy_mean,
      'Speech rate': speech_rate
    }

    audio_features_list.append(feature_row)

# Convert to a DataFrame
df = pd.DataFrame(audio_features_list)

# Save to Excel
output_path = 'cr_dataset.xlsx'
df.to_excel(output_path, index=False)

print(f"Features extracted and saved to {output_path}")
```

Imputation of missing values was done in case any errors in feature extraction of any specific data point occurred, which was initially handled by initially assigning it the value 0, and then later imputing it with the mean value of that feature across all data points.

$$\text{mean} = \overline{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

After processing, all audio clips were consolidated into a single CSV file, with each data point labelled as Female AI (F_AI), Male AI (M_AI), Female Real (F_REAL), or Male Real (M_REAL) for streamlined analysis and model training.

3. DEVELOPMENT METHODOLOGY

3.1. Development model

The methodology used for the development of this project integrates principles from the CRISP-DM (Cross-Industry Standard Process for Data Mining) framework along with software engineering best practices. Given its iterative nature and alignment with machine learning workflows, CRISP-DM was chosen as the overarching methodology. The six stages, Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, and Deployment, were tailored to fit the project's requirements. Feature engineering, including normalization, extraction, and scaling, played a critical role in data preparation, while transfer learning and neural networks were leveraged for model development and deployment.

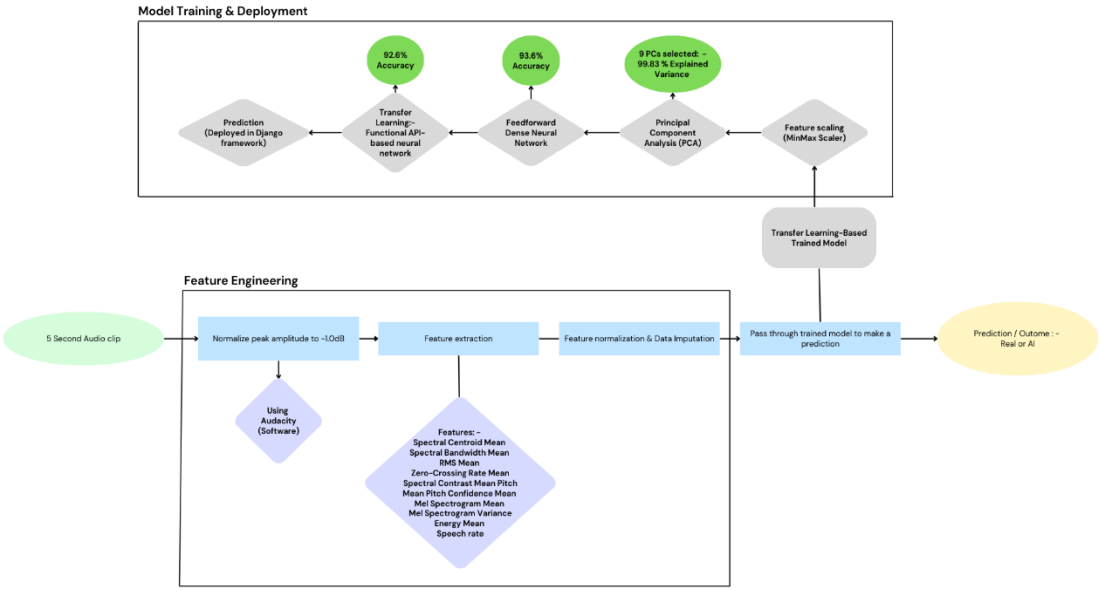

Figure 1 - End-to-End Audio Classification Pipeline: From Feature Engineering to Model Deployment

3.2. Version control and tools

To ensure reproducibility and collaborative tracking of progress:

- **Git** was used for version control, with code hosted on **GitHub** in a private repository.

- **Jupyter Notebooks** were used for prototyping and exploratory data analysis.

Python 3.9 was the primary programming language, utilizing key libraries:

- **scikit-learn** for machine learning, model evaluation, pre-processing, and cross-validation
- **xgboost, catboost, lightgbm** for gradient boosting models
- **TensorFlow and Keras** for deep learning, neural networks, and optimization
- **joblib** for model persistence and serialization
- **seaborn and matplotlib** for data visualization
- **pandas and numpy** for data manipulation and numerical computing
- **mpl_toolkits (3D plotting)** and **PCA (dimensionality reduction)** for data analysis
- **LabelEncoder and OneHotEncoder** for categorical feature encoding

- **Various ML models**, including:

  - **K-Nearest Neighbors (KNN)**
  - **Logistic Regression**
  - **Naïve Bayes (GaussianNB, BernoulliNB)**
  - **Random Forest, Gradient Boosting, Extra Trees**
  - **Linear Discriminant Analysis (LDA)**
  - **Decision Tree**
  - **Support Vector Machines (SVM)**

- **Deep Learning architectures**, including:

  - **Sequential and Functional API models**
  - **SimpleRNN for recurrent neural networks**
  - **Dense, Dropout, BatchNormalization, and Activation layers**
  - **Optimizers - Adam, SGD, RMSprop, Adagrad**

3.3. Data development

As detailed in Chapter 2, a custom dataset of 1,956 five-second audio clips was created. Each clip was carefully labelled with speaker origin (AI vs. real) and gender (male or

female), forming a balanced and diverse dataset across multiple accents. Normalization and pre-processing ensured consistent quality across samples, enabling effective feature extraction. Key audio features were extracted to capture pitch, spectral content, energy distribution, and temporal dynamics.

3.4. Model development and evaluation

To ensure technical robustness and academic depth in this project, a rigorous evaluation framework was adopted.

3.4.1. Feature Engineering

Normalization: Audio signals are normalized to a peak amplitude of -1.0 dB using Audacity software, which ensures uniform loudness across all samples, preventing discrepancies caused by varying recording levels.

Feature Normalization & Data Imputation: To ensure consistency, extracted features are normalized using MinMax Scaling, bringing values into a fixed range. Any missing values in the dataset are imputed using appropriate statistical techniques to maintain data integrity.

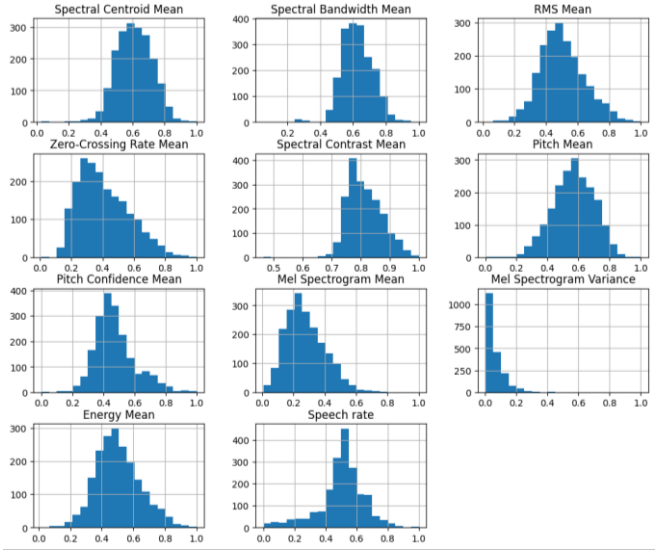

Figure 2 - Normalized Feature Distribution

Dimensionality Reduction: To eliminate redundancy and improve computational efficiency, Principal Component Analysis (PCA) is applied. 9 principal components (PCs) are selected, explaining 99.83% of the variance, ensuring minimal information loss.

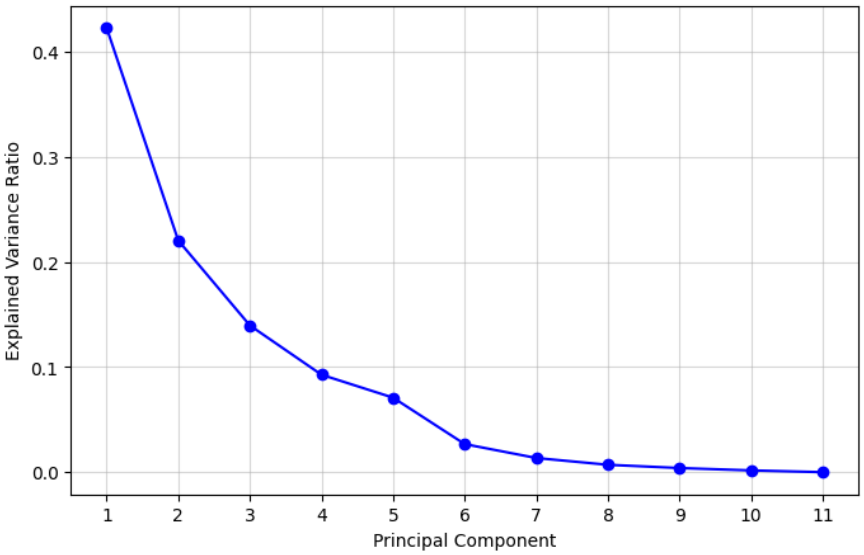

Figure 3 - Scree Plot showing Explained Variance at each PC

Feature Scaling: MinMax Scaler is used to normalize feature values within a defined range, ensuring uniform weightage across all features.

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

3.4.2. Model selection

Models tested included k-Nearest Neighbours (kNN), Naive Bayes (Bernoulli and Gaussian), Decision Trees, Logistic Regression, Linear Discriminant Analysis, Support Vector Machines (SVM), Random Forest, Gradient Boosting, XGBoost, CatBoost, LightGBM, and Extra Trees. Deep learning models tested were Recurrent Neural Networks (RNN), Feedforward Neural Networks (FFNN), and a Transfer Learning Functional API-Based Neural Network.

Each model was trained and evaluated based on average accuracy, precision, recall, F1-score across all train-test splits and confusion matrices as well as their ability to adapt the most effective approach for the task.

3.4.3. Transfer Learning

Given dataset limitations, a transfer learning-based FFNN was developed using pre-trained weights and fine-tuned on the custom dataset. This approach significantly improved performance and generalizability as discussed in chapter 4.

3.5. Model Performance Evaluation (Testing and Validation)

The PCA-transformed dataset with normalized features was run through multiple machine learning models to determine the best-performing approach.

Models tested included k-Nearest Neighbours (kNN), Naive Bayes (Bernoulli and Gaussian), Decision Trees, Logistic Regression, Linear Discriminant Analysis, Support Vector Machines (SVM), Random Forest, Gradient Boosting, XGBoost, CatBoost, LightGBM, and Extra Trees. Deep learning models tested were Recurrent Neural Networks (RNN), Feedforward Neural Networks (FFNN), and a Transfer Learning Functional API-Based Neural Network. These models were evaluated to identify the most effective approach for the task.

Testing was performed at multiple levels:

- **Comparative Testing** in which all 15 models were evaluated using a standardized set of performance metrics, including accuracy, precision, recall, and F1-score. This allowed for a fair comparison of different algorithms in terms of their ability to classify AI-generated and real voices. Feature importance, computational efficiency, and model complexity were also considered to determine the best-performing approach.
- **Real-world Testing** using voice samples post-deployment. Beyond controlled experiments, the deployed model was tested on real-world voice samples to assess its performance in practical scenarios. The unseen audio clips were run through the deployed model via the app, which predicted the most likely class along with its confidence percentage. This provided valuable insights into how well the model had generalized, offering a clear indication of its effectiveness in distinguishing AI-generated voices from real ones in practical applications.

3.5.1. Correlation Matrix

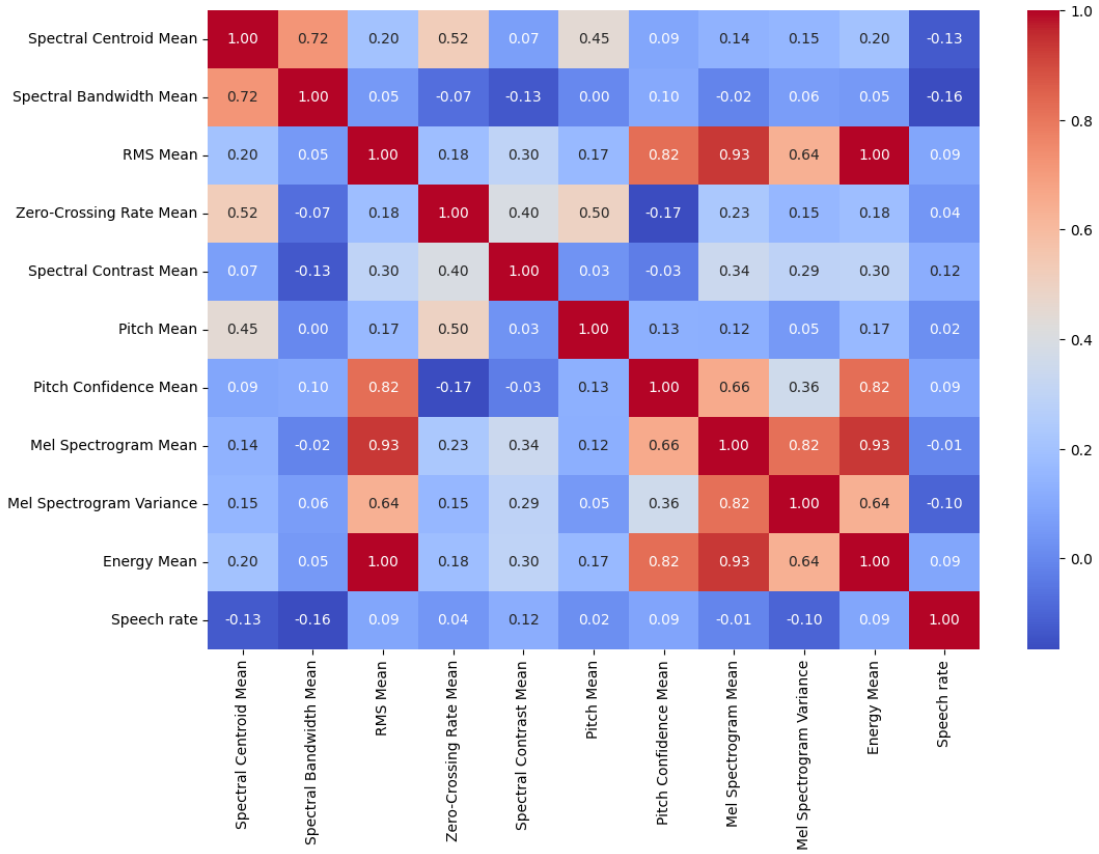

Figure 4 - Feature Correlation Matrix

The high correlations in the matrix reflect the intrinsic relationships between these audio features, which all measure different aspects of the energy, pitch, and frequency content of sound.

As can be seen in detail in Table 3, Features like spectral centroid, spectral bandwidth, RMS mean, and Mel spectrogram features are all closely related because they capture aspects of the sound's energy and frequency distribution. The features related to energy (such as RMS mean and energy mean) are often highly correlated with each other, as they describe similar aspects of loudness or intensity.

Similarly, pitch confidence is often related to features that describe the spectral characteristics of the signal, as clearer pitch patterns typically occur in more structured and energetic sounds.

These relationships emerge from the way different aspects of sound, such as loudness, frequency distribution, and spectral structure, interact to shape the

overall perception of an audio signal. Understanding these correlations provides valuable insights into the data's behaviour and plays a crucial role in guiding the next steps toward Principal Component Analysis (PCA).

.

Table 3 - correlation between various audio features, their explanations, and the reasons behind these correlations

Feature Pair	**Correlation**	**Explanation**	**Reason Behind Correlation**
Spectral Bandwidth and Spectral Centroid	High	Both are related to the distribution of energy across the frequency spectrum. Spectral centroid indicates the 'center of mass' of the spectrum, while spectral bandwidth describes the spread of the frequencies.	A signal with high spectral centroid often has more energy in higher frequencies, leading to a higher spectral bandwidth as well. Both features are influenced by the amount of high-frequency content in the signal.
RMS Mean and Energy Mean	High	RMS reflects overall energy in the time domain, and energy mean reflects the average energy across time.	Both are measures of the overall energy content in a signal, so they are highly correlated. A higher RMS generally indicates a louder or more energetic signal, which corresponds to a higher energy mean.
RMS Mean and Mel Spectrogram Variance	High	RMS measures the overall energy, and Mel spectrogram variance measures fluctuations in the energy distribution across frequency bands.	Signals with higher overall energy (RMS) often exhibit more variability in frequency distribution, leading to a higher Mel spectrogram variance.
RMS Mean and Mel Spectrogram Mean	High	RMS reflects overall energy, while Mel spectrogram mean reflects the average energy across different frequency bands.	Higher energy levels are reflected both in the RMS value and the average energy across frequency bands in the Mel spectrogram, leading to a high correlation

			between these features.
RMS Mean and Pitch Confidence Mean	High	RMS measures the energy of the signal, and pitch confidence reflects how clearly pitch can be detected in the signal.	Higher energy signals often have clearer and more defined pitch, making pitch detection more reliable, leading to a high correlation with RMS.
Energy Mean and Mel Spectrogram Mean	High	Both describe aspects of the overall energy of the signal, with energy mean reflecting total energy and Mel spectrogram mean reflecting the energy distribution across frequency bands.	Signals with higher total energy typically have higher energy in their Mel spectrogram as well, leading to a strong correlation.
Mel Spectrogram Variance and Mel Spectrogram Mean	High	Mel spectrogram variance captures fluctuations in frequency energy distribution, while the mean captures the average energy across frequency bands.	Consistent energy distribution in the Mel spectrogram (less variation) often results in both a high mean and lower variance. Thus, both features tend to move in similar directions.
Zero Crossing Rate Mean and Pitch Mean	Moderate (50-70%)	Zero crossing rate measures how often the signal crosses zero amplitude, while pitch mean reflects the average pitch of the signal.	High zero crossing rates often correlate with higher frequency content, which can also increase the spectral centroid and affect pitch detection. However, the relationship is weaker because zero-crossing rate also depends on sound texture, not just pitch.
Zero Crossing Rate Mean and Spectral Centroid Mean	Moderate (50-70%)	Zero crossing rate indicates percussiveness or noisiness, while spectral centroid indicates where the	Higher zero-crossing rates, often linked to noisier sounds, can coincide with higher spectral centroid values as both can

		energy is concentrated in the frequency spectrum.	arise from signals with more high-frequency content. However, the correlation is not perfect due to other factors influencing both measures.
Zero Crossing Rate Mean and Pitch Confidence Mean	Moderate (50-70%)	Zero crossing rate measures noisiness or percussiveness of a sound, while pitch confidence measures how confidently pitch is detected.	In tonal sounds, zero crossing rate is often low because pitch is clear, leading to higher pitch confidence. Noisy or percussive sounds with higher zero-crossing rates may have lower pitch confidence.

3.5.2. Principal Component Analysis

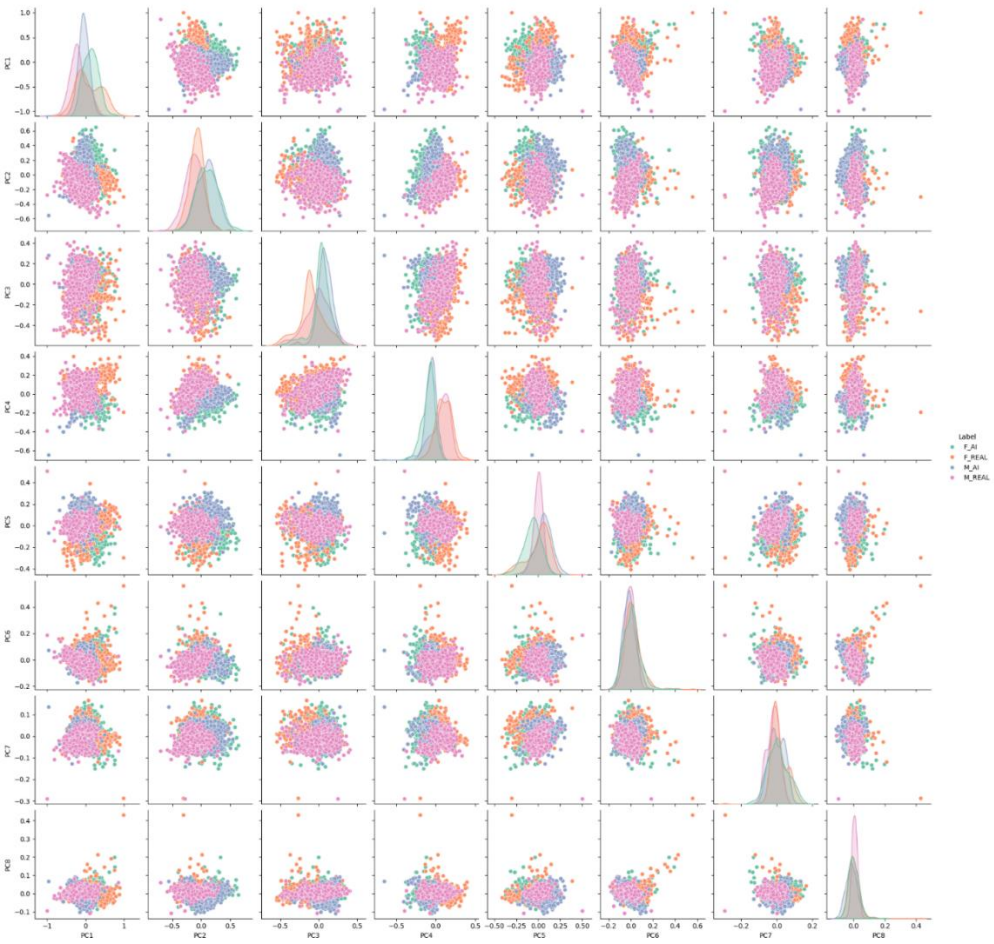

Figure 5 - Pairwise Scatter Plot of Each Principal Component

After deriving valuable insights from the feature correlation matrix, Principal Component Analysis (PCA) was applied to reduce dimensionality while preserving the most informative features.

As can be seen in Figure 4, the pairwise scatterplot reveals very interesting clear distinctions among the four classes, F_AI, F_REAL, M_AI, and M_REAL.

More notably, there is a strong separation between AI and REAL voices overall, with minimal overlap, particularly evident in PC2 vs. PC4 and PC2 vs. PC1.

The distinction becomes even more pronounced at the class level, as seen in PC5 vs. PC2.

This visualization confirms that AI and real voices have distinct feature distributions, reinforcing the model's ability to differentiate between them. Moreover, certain principal components contribute more to class separation, which can be used for further optimization in classification tasks.

4. RESULTS

The performance of various machine learning and deep learning models was evaluated based on average accuracy across all train-test splits, in detecting whether a given audio clip is synthetic or that of Real human with further distinguishing between male and female distinctions, as depicted in Figure 6. The results indicate that traditional machine learning models, such as k-Nearest Neighbours (kNN) and Naïve Bayes (Bernoulli and Gaussian variants), exhibited lower performance, with kNN achieving the lowest accuracy (below 40%). However, tree-based models like Decision Trees, Random Forest, Gradient Boosting, XGBoost, CatBoost, LightGBM, and Extra Trees demonstrated strong predictive capabilities, reaching accuracy values between 75% and 90%.

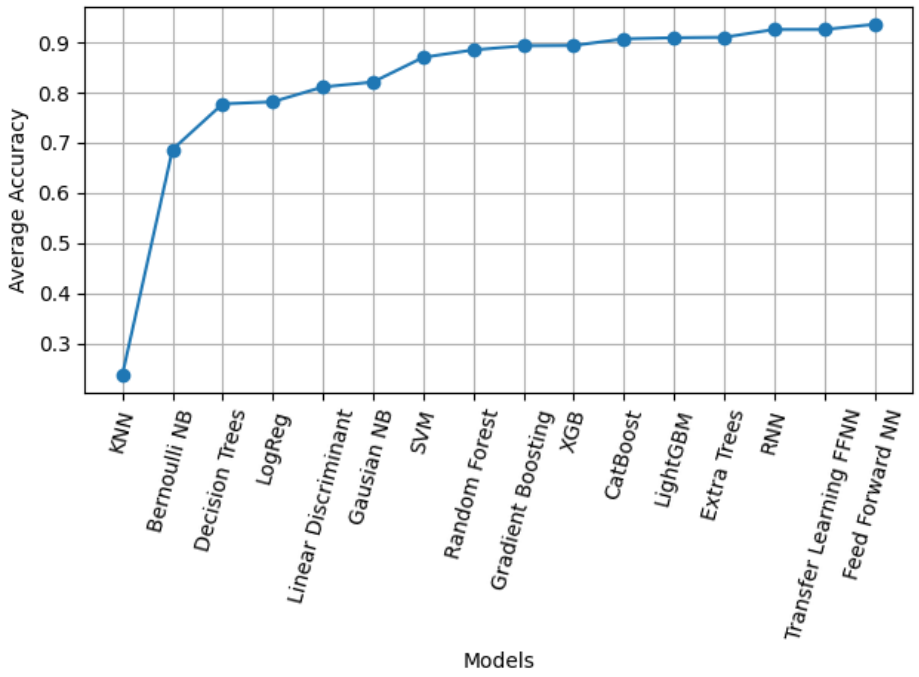

Figure 6 - Model Performance Comparison (Accuracy)

Among deep learning models, Feedforward Neural Networks (FFNN) and Recurrent Neural Networks (RNN) outperformed most traditional models, achieving accuracies above 90%. However, based on the experimental results, Transfer Learning FFNN would be the most suitable model for this small dataset due to its ability to generalize well with limited data, faster training time, and superior accuracy compared to other methods.

While traditional ensemble models and deep learning architectures performed well, the transfer learning approach by nature would be very efficient, making it the optimal choice for this task. Its ability to use pre-trained knowledge allows it to achieve accuracy with minimal data and computational resources, ensuring robust performance even in limited-data scenarios.

For this classification task, the Transfer Learning Neural Network, pre-trained on a Feed Forward Neural Network architecture is selected as the final model due to its adaptive nature. While traditional ensemble models and deep learning architectures performed well, the transfer learning approach by nature would be very efficient, making it the optimal choice for this task. Its ability to use pre-trained knowledge allows it to achieve accuracy with minimal data and computational resources, ensuring robust performance even in limited-data scenarios.

4.1. Why Transfer Learning FFNN

Effective Feature Representation for Speech Data

The dataset contains a rich set of acoustic, prosodic, and speech-related features, including spectral centroid, bandwidth, energy, pitch, Mel spectrogram statistics, and speech rate. Transfer learning enables the model to use audio representations that were pre-trained on a small section of the dataset on a Feed Forward NN, allowing it to better understand these complex features and extract meaningful patterns more effectively than models trained from scratch.

Superior Generalization Despite Small Data Size

While 1,956 audio clips provide a solid foundation, deep learning models typically require much larger datasets to generalize effectively. Transfer learning mitigates this issue by utilizing pre-trained feature extractors, reducing the likelihood of overfitting while improving classification performance.

Reduced Training Time and Computational Efficiency

Training a deep learning model from scratch on a dataset of this size would require substantial computational resources and time to achieve optimal performance. The Transfer Learning FFNN, however, uses pre-trained weights and fine-tunes only a subset of layers, leading to faster convergence and lower computational cost compared to fully training an FFNN or RNN.

Higher Accuracy Compared to Traditional Models

Traditional models like Logistic Regression, Decision Trees, and ensemble methods (Random Forest, XGBoost, CatBoost, and Extra Trees) performed well but were ultimately outperformed by deep learning models. This suggests that the high-dimensional and complex nature of speech features is better captured by deep neural networks rather than manually designed decision trees or linear classifiers.

Ability to Capture Temporal and Frequency Variations

Speech classification involves both temporal and spectral features, which means the model must capture variations over time and frequency. While RNNs are known for handling sequential data, the Transfer Learning FFNN outperformed RNN in this case, due to its ability to extract high-level, pre-trained representations that generalize well to this dataset.

4.2. Architecture of the Transfer Learning FFNN and Its Strengths

The Transfer Learning FFNN follows a two-stage approach, where an initial FFNN model is trained on the dataset, and its last hidden layer (layer before the output layer) is extracted and used as a feature extractor for the final transfer learning model.

4.2.1. Base Model (Initial FFNN):

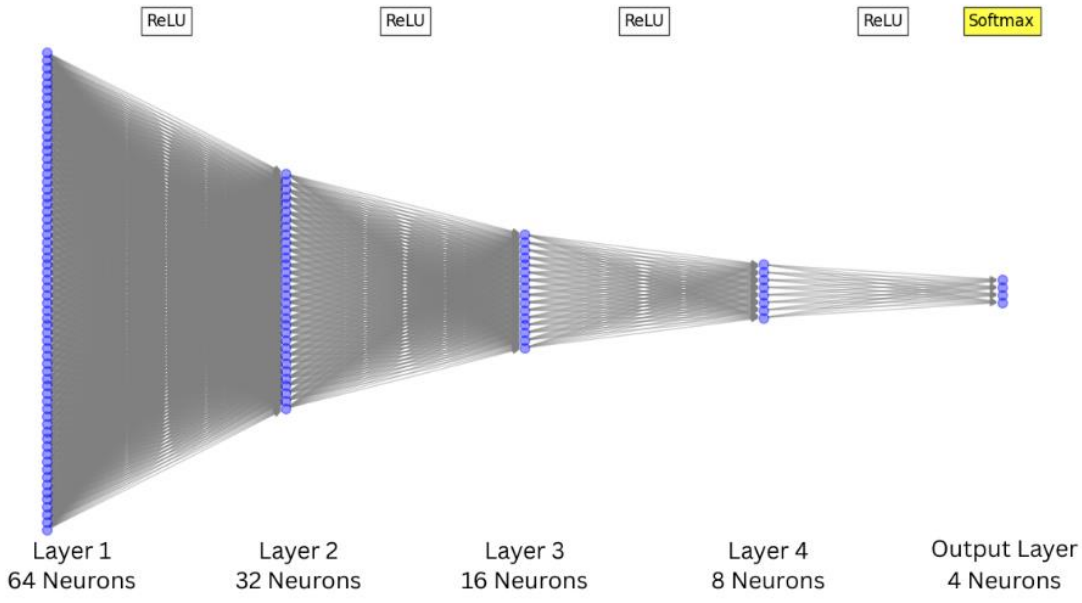

Figure 7 - Feed Forward Neural Network Visualization

Optimization Strategy

Several optimization algorithms were evaluated to determine the most effective approach for training the model. The optimizers tested included:

- **Adam** (Adaptive Moment Estimation)
- **SGD** (Stochastic Gradient Descent)
- **RMSprop** (Root Mean Square Propagation)
- **Adagrad** (Adaptive Gradient Algorithm)

After extensive hyperparameter tuning, the Adam optimizer with a learning rate of 0.005 proved to be the most optimal for our dataset. The categorical cross-entropy loss function is used to evaluate the model's performance, making it suitable for multi-class classification tasks.

Procedure

The model is trained for 150 epochs using a batch size of 64. This means that in each epoch, the audio dataset is divided into mini-batches of 64 samples, and the model updates its parameters after processing each batch. The training process ensures efficient gradient updates while balancing computational efficiency and model generalization.

Model Architecture

The base model consists of multiple fully connected layers with batch normalization, ReLU activations, and dropout for regularization, designed to optimize performance for multi-class classification.

The **First Dense Layer** consists of 64 neurons, followed by batch normalization, ReLU activation, and dropout (0.2) to prevent overfitting.

$$ReLU(x) = max(0, x)$$

The **Second Dense Layer** consists of 32 neurons, batch normalization, ReLU activation, and dropout (0.2).

The **Third Dense Layer** consists of 16 neurons with batch normalization and ReLU activation.

The **Fourth Dense Layer** consists of 9 neurons with batch normalization and ReLU activation.

The **Output Layer** consists of 4 neurons with SoftMax activation for multi-class classification.

$$Softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

4.2.2. Transfer Learning Model:

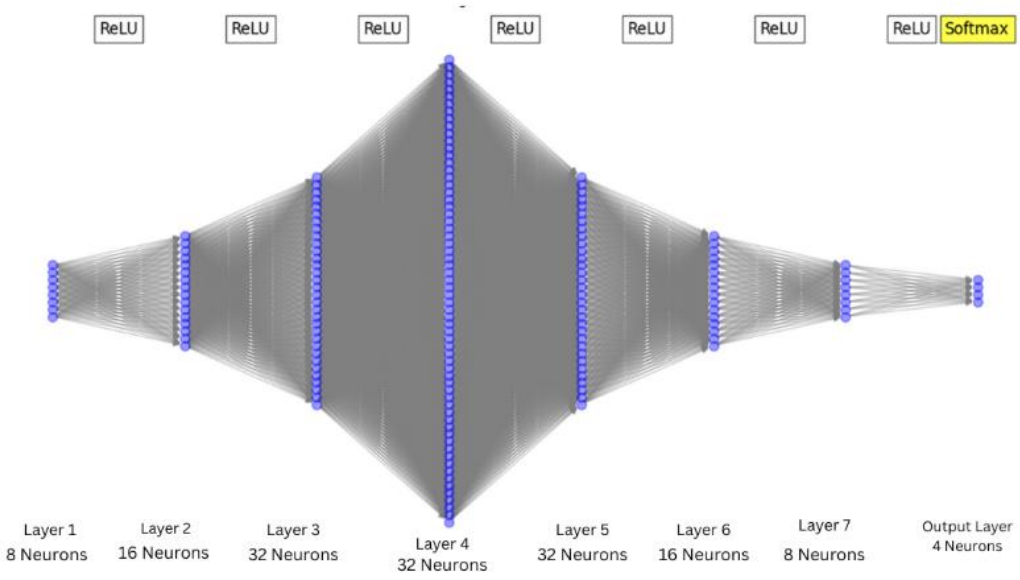

Figure 8 - Transfer Learning Model visualization

Optimization Strategy

Several optimization algorithms were evaluated to determine the most effective approach for training the model. The optimizers tested included:

- **Adam** (Adaptive Moment Estimation)
- **SGD** (Stochastic Gradient Descent)
- **RMSprop** (Root Mean Square Propagation)
- **Adagrad** (Adaptive Gradient Algorithm)

After extensive hyperparameter tuning, the Adam optimizer with a learning rate of 0.001 was selected for the final training process. The categorical cross-entropy loss function is used to evaluate the model's performance, making it suitable for multi-class classification tasks.

Procedure

The model is trained for 150 epochs using a batch size of 64 in which in each epoch, the audio dataset is divided into mini-batches of 64 samples, and the model updates its parameters after processing each batch.

Freezing the base model layers ensures that pre-trained representations are retained, while the newly added layers adapt to the dataset. The training process optimizes feature extraction and classification performance.

Model Architecture

The model uses a feature extraction approach, where the last hidden layer of the base model Feedforward Neural Network (FFNN) is used as a feature extractor, preserving learned representations. A new FFNN is built on top of this extractor with additional layers for classification.

As shown in Figure 8, the newly added layers consist of 7 Dense layers with batch normalization, ReLU activation, and dropout in the first 3 of layers for regularization, and an output layer consisting of 4 neurons using SoftMax activation for classification into four the 4 output classes.

This architecture ensures that the base model retains its pre-trained knowledge while the new layers refine this classification task with limited amount of data points in the audio dataset.

4.3. Why This Architecture Performs Well

4.3.1. Pre-trained Feature Extraction

The last hidden layer of the base FFNN preserves essential information extracted from the speech features. Instead of learning from scratch, the model leverages previously learned representations, making training more efficient.

4.3.2. Regularization Techniques for Stability

Batch normalization is essential for stabilizing and accelerating the training of deep neural networks. It normalizes the activations of each layer by adjusting and scaling them, leading to several benefits:

Improved Convergence Speed: Normalizing the inputs to each layer helps keep activations within a controlled range, allowing the model to train faster.

Reduced Internal Covariate Shift: BN mitigates the issue where the distribution of inputs to each layer changes during training, improving overall stability.

Better Generalization: By making training more stable, BN indirectly reduces overfitting, leading to better performance on unseen data.

Dropout is a regularization technique where random neurons are "dropped" (set to zero) during training, preventing reliance on specific neurons. The key advantages include:

Prevents Overfitting: By randomly deactivating neurons, dropout forces the model to learn redundant and diverse representations, improving generalization.

Encourages Robust Feature Learning: The network learns to distribute weights more evenly, making it less dependent on any single neuron.

Works Well with Batch Normalization: While BN stabilizes training, dropout adds an extra layer of robustness by preventing co-adaptation among neurons.

4.3.3. Efficient Gradient Optimization

The Adam optimizer is well-suited for this architecture, ensuring adaptive learning rates that converge faster than standard gradient descent.

Adam Optimizer Formula

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$
$$\theta_t = \theta_{t-1} - \frac{\alpha \hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon}$$

4.3.4. Higher Classification Accuracy with Minimal Data

The transfer learning approach allows the model to generalize well even with limited training samples. It outperforms traditional machine learning models that struggle with high-dimensional feature spaces and effectively captures both spectral and temporal variations in speech, making it highly suitable for complex audio classification tasks.

4.4. Deployment in Django-based Web Application

To make the model accessible and user-friendly, deployment of the best-performing model was done in a Django-based web application.

The web application allows users to upload an audio clip and receive real-time predictions on whether the clip is AI-generated or real.

A user-friendly interface was developed using HTML, CSS, and JavaScript to ensure ease of use for non-technical users.

The Django application integrates the machine learning model with a REST API, allowing for seamless interaction between the front-end and back-end. The deployment process included setting up a virtual environment, ensuring proper model serialization using joblib, and testing the application to guarantee smooth deployment.

The web application provided a convenient platform for potential end-users to interact with the deepfake detection model, offering a practical use case for the research findings.

5. PROFESSIONAL CONDUCT

Professional and ethical conduct played a central role throughout the development of this project. All phases, from data collection and model development to deployment and evaluation, were guided by best practices in project management, data ethics, and academic integrity. This chapter outlines how the project was conducted responsibly, adhering to established professional standards and institutional expectations.

5.1. Project management and scheduling

A detailed project plan was developed at the outset, using **Gantt chart** to define clear milestones, deliverables, and timelines. Tasks were divided into phases:

1. Proposal & Research
2. Dataset Creation
3. Model Development
4. Evaluation & Optimization
5. Web App Deployment
6. Documentation & Final Reporting

Regular progress was monitored using weekly logs and checklists to track task completion. Adjustments were made when necessary to maintain momentum, especially when data collection took longer than anticipated.

The main activities and milestones are depicted in Gannt chart (Figure 9).

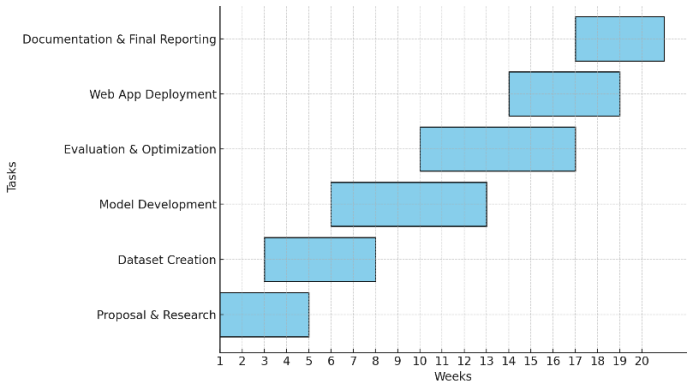

Figure 9 - Gantt Chart

5.2. Data Management and Logging

All data was stored in a structured directory system with proper versioning and backups. A secure local environment was used for raw audio files, with processed data stored in standardized formats (e.g., .wav, .csv for feature matrices).

A detailed project log was maintained, recording:

- Weekly activities
- Feature extraction processes
- Model experiments and results
- Technical issues and resolutions
- Supervisor feedback and implementation steps

This ensured traceability, transparency, and easy reproducibility of all development stages.

5.2.1. Data Collection Ethics

All audio data was sourced from publicly available content or licensed platforms (e.g., podcasts, speeches, Canva TTS), ensuring compliance with usage rights. No personal or sensitive data was used. Speakers were not identifiable, and clips were trimmed to short, decontextualized segments, reducing privacy risks.

5.2.2. AI Ethics

As the project centres on deepfake detection, ethical implications were taken seriously by (1) addressing misuse of synthetic voices (cybercrime, misinformation), (2) promoting transparency and trust in digital media, (3) mitigating biases in detection by including diverse accents and genders

5.2.3. Legal and IP Considerations

The dataset was created from scratch, ensuring no third-party proprietary datasets were used. Tools and libraries used were open-source and properly cited. The final code and model were developed in accordance with MIT License. (Initiative, 2021)

5.2.4. EDI (Equality, Diversity & Inclusion)

Efforts were made to ensure gender and accent diversity in the dataset, which supports fairness and inclusivity in model training. Recognizing underrepresentation in typical

datasets, this project aimed to include global English accents (e.g., African, Indian) that are often overlooked in mainstream voice AI models.

5.2.5. Environmental Considerations

Lightweight models were prioritized to reduce training time and energy use. Experiments were run locally rather than on power-intensive cloud GPUs. The use of transfer learning reduced computational cost by avoiding full model retraining.

5.3. Professional Codes of Conduct

The project adheres to established professional codes of conduct from BCS (British Computer Society) Code of Conduct to ensure ethical, legal, and responsible AI research practices. Given the sensitivity of deepfake detection technology and its potential societal impact, adherence to these standards was crucial.

The primary goal was to detect deepfake audio, helping to prevent its misuse in fraudulent activities and misinformation.

Research methodologies were applied with professionalism to ensure the results were reliable, reproducible, and free from bias.

The project complied with data protection regulations, such as GDPR, to ensure the ethical handling of the dataset.

Efforts were made to minimize the environmental impact of the project by optimizing computational resources and utilizing cloud-based services efficiently such as Google Colab.

The research was carried out with integrity, aiming to contribute positively to society by addressing the risks associated with AI-generated content.

5.4. Activities

A structured approach was followed in this to ensure efficient data collection, feature extraction, model evaluation, and analysis of results. The major activities were carried out in a step-by-step manner, as detailed below:

5.4.1. Dataset Acquisition and Pre-processing:

The dataset used for this project consists of 1,956 audio clips, equally distributed across four classes (Female AI, Male AI, Female Real, Male Real) to avoid biases that could affect the model performance.

The audio files were pre-processed using Audacity for normalization to a consistent peak amplitude of -1.0 dB, ensuring uniform loudness across all samples.

The audio clips were then converted into features, including spectral, prosodic, and other speech-related characteristics. This step involved using libraries such as Librosa for feature extraction.

Missing values in the dataset were handled through imputation techniques, and MinMax Scaling was applied to normalize the feature values, ensuring they were within a standard range for input into machine learning models.

5.4.2. Feature Engineering

The extracted features included spectral, prosodic, and other speech-related which were key to differentiating AI-generated voices from real human speech by distinguishing them on the basis of subtle differences in brightness, loudness, spread, timbre, power, frequency and speed of the audio clips.

Principal Component Analysis (PCA) was applied for dimensionality reduction, reducing the complexity of the feature set by selecting the 9 most significant principal components ensuring that the models were trained on the most relevant features and improved computational efficiency by reducing noise and redundancy.

5.4.3. Model Training and Evaluation

The project used a combination of 15 different machine learning and deep learning models to evaluate their performance in detecting deepfake AI audio. These included models such as k-Nearest Neighbours (kNN), Support Vector Machines (SVM), Random Forest, Gradient Boosting, as well as deep learning models like Recurrent Neural Networks (RNN) and Feedforward Neural Networks (FFNN).

Each model was trained on the pre-processed and PCA-transformed dataset, with hyperparameter tuning conducted to optimize performance.

Evaluation metrics such as accuracy, precision, recall, and F1-score were used to assess the models' performance in distinguishing between real and AI-generated speech.

5.4.4. Performance Comparison and Analysis

After training the models, performance metrics were compiled to compare each model's ability to accurately classify the audio clips.

Feature importance was also analyzed to determine which features (such as spectral centroid, pitch, or Mel spectrogram variance) contributed most to the model's ability to detect deepfakes.

A detailed analysis was conducted to identify the strengths and weaknesses of different models and to determine the most effective model for deepfake AI audio detection.

5.4.5. Documentation

Throughout the project, detailed notes were taken on each step, from data pre-processing to model evaluation. This included recording model configurations, hyperparameters, and experiment results for transparency.

The findings were documented in a comprehensive dissertation report, covering the methodology, data analysis, results, and conclusions. The final report was organized into logical sections, including an introduction, literature review, methodology, results, and conclusion with recommendations for future work.

5.5. Data Management

Efficient handling of data and research materials was critical to ensure consistency, reproducibility, and transparency in the project. The following data management strategies were employed:

5.5.1. Version Control

Cloud Services like GitHub, Google Drive and Google were used for version control of both the code and models. This ensured that any changes made to the project (whether code modifications or model adjustments) were tracked and could be rolled back if necessary. It also allowed for collaborative work, as any updates were easily accessible to others.

5.5.2. Project Logs

A project log was maintained throughout the duration of the project, documenting key actions taken, decisions made, and results from model training and hyperparameter tuning. This log helped track progress and allowed for effective troubleshooting in case of issues.

Specific logs were kept for each model's evaluation results, including training accuracy, validation performance, and final test accuracy.

5.5.3. Reference Management

Microsoft Word was used to manage all relevant literature and citations. This tool helped organize research papers, articles, and reports that were referenced throughout the dissertation. It also made citation management easier and ensured consistency in referencing throughout the report.

5.6. Deliverables

The project included several key deliverables that were finalized and submitted at various stages of the project. These deliverables were essential to ensure continuous progress and clear communication throughout the duration of the project.

Project Proposal - The project started with a detailed proposal outlining the objectives, scope, methodology, expected deliverables, and timeline. This proposal served as a roadmap for the project and ensured alignment with the project goals. It was submitted for approval at the initial stage.

Progress Report - A progress report was created to track the ongoing developments of the project. The report included updates on the current status of the dataset, feature engineering, model selection, and preliminary results. This report helped to identify any potential challenges early and enabled the supervisor to provide guidance on addressing them.

Weekly Sprints - To maintain focus and monitor steady progress, weekly sprint meetings were held with the project supervisor. These weekly sprints provided structured feedback and ensured that the project progressed efficiently and effectively.

Each sprint was dedicated to specific aspects of the project, such as:

- **Feature Engineering**: Refining and processing the dataset for optimal model input.
- **Model Selection and Evaluation**: Testing and comparing various machine learning and deep learning models to identify the most effective one for deepfake audio detection.
- **Documentation**: Ensuring detailed documentation of the methodology, results, and analyses was maintained throughout the project to support reproducibility and clarity.

6. DISCUSSION

Throughout the project, various challenges were encountered at different stages, from data preprocessing to model deployment. Each issue required careful analysis and iterative solutions to improve the overall system performance. The key challenges and their resolutions are detailed below.

6.1. Data Pre-processing Challenges

Audio Normalization Issues – Initially at the start of the data collection process, the audio clips in the dataset were not uniformly normalized, leading to inconsistencies in volume levels. This affected the extracted features, which in turn impacted model performance by introducing noise into the classification process. To address this, peak normalization was applied to all audio files using Audacity software, ensuring that all samples were at a consistent loudness level before feature extraction.

Feature Extraction Errors - Some features, particularly speech rate, were not extracted properly for certain audio clips, leading to missing values. In such cases, the feature value was defaulted to 0 and later imputed using the mean of other values in the column. This resulted in a loss of variance, which could impact the model's ability to distinguish between real and AI-generated voices. To minimize data loss, imputation techniques such as Mean Imputation were considered, but ultimately, stricter validation of extracted features was implemented to reduce missing values in future iterations.

Dimensionality Reduction Trade-offs - Principal Component Analysis (PCA) was applied to reduce feature dimensions while preserving variance. However, selecting the optimal number of principal components was difficult, removing too many could lead to loss of information, while keeping too many could introduce noise. A balance was struck by selecting 9 principal components, which explained 99.83% of the variance, ensuring minimal information loss while maintaining computational efficiency.

6.2. Model Training and Evaluation Challenges

Imbalanced Feature Importance - Some features contributed more significantly to classification than others, leading to potential biases in model training.

Various feature selection techniques, including Recursive Feature Elimination (RFE), correlation matrix and permutation importance analysis, were used to refine feature

importance and reduce redundancy. Using these techniques, a feature that stored the file size of the audio clip was eliminated as it didn't have much correlation with other features and did not contribute a lot to the explained variance in PCA.

Variability in Model Performance - Different models performed inconsistently across evaluation metrics. While deep learning models showed promising results, some simpler models (e.g., KNN) failed to capture the complexity of AI-generated speech.

Extensive hyperparameter tuning and ensemble techniques (such as Random Forest and XGBoost) were explored to improve overall classification accuracy.

Generalization Issues - The model sometimes struggled to classify unseen deepfake audio, indicating an overfitting problem on the training dataset.

Data augmentation techniques (such as considering speech rate and pitch variation) were tested to improve model robustness, though computational constraints limited the extent of augmentation applied.

6.3. Deployment Challenges

Incorrect Class Mapping in Deployment - After successfully training the model, it consistently produced incorrect predictions during the deployment phase. Upon investigation, it was discovered that the list of class labels in the deployment code was not in the same order as the output labels from the neural network, leading to incorrect class assignments.

The class label order was corrected by ensuring consistent indexing between the trained model and the deployed API, aligning both during inference.

6.4. Computational Constraints

Processing Large Audio Data - Training models on high-dimensional audio features required significant computational resources, particularly for audio files volume and normalization deep learning models.

A combination of feature selection, PCA, and cloud-based computing resources was used to optimize training time without sacrificing accuracy.

6.5. Ethical and Practical Considerations

Bias in AI Detection - The dataset was carefully created by gathering important key features, but it was noticed that some classes were overrepresented than the others. The model would then struggle with detecting deepfake voices generated using more advanced synthesis techniques and led to Incorrect classification of real voices as deepfake (false positives) or deepfake voices as real (false negatives).

To tackle this problem all the classes were amounted to the same quantity of data points, eliminating risk of biases.

7. CONCLUSIONS

The research presented in this dissertation aimed to develop a system capable of distinguishing between AI-generated voices and real human voices, with the additional challenge of gender classification. By building a custom dataset containing both synthetic and real audio samples, including various accents and speech types, the study contributed valuable insights to the field of deepfake audio detection. Through a series of experiments involving traditional machine learning models and deep learning architectures, the study evaluated multiple models to determine the best approach for this task.

7.1. Summary of Key Findings

One of the key findings of this research is that traditional machine learning models, such as k-Nearest Neighbours (kNN) and Naïve Bayes, were less effective at distinguishing between AI-generated and real human voices, with kNN performing particularly poorly with an accuracy below 40%. This highlights the limitations of classical methods in handling complex, high-dimensional audio data. In contrast, tree-based models such as Random Forest, XGBoost, and LightGBM performed significantly better, with accuracies ranging between 75% to 90%. These models utilized feature importance techniques to capture the complex, non-linear relationships within the audio data.

Among the deep learning models, Feedforward Neural Networks (FFNN) and Recurrent Neural Networks (RNN) performed better than the traditional models, achieving accuracies exceeding 90%. Notably, the Transfer Learning FFNN emerged as the most effective model, excelling in small-data scenarios and demonstrating the potential of leveraging pre-trained knowledge to improve model performance with limited resources. This approach significantly reduced training time and enhanced the model's ability to generalize, making it a promising solution for real-world deepfake detection.

The inclusion of gender classification in the task, as per the project's aim, provided an added layer of complexity and practical relevance. The Transfer Learning FFNN model successfully distinguished between male and female voices in addition to classifying AI-generated versus real voices, fulfilling the project's objective of providing a robust and scalable classification system.

7.2. Theoretical and Practical Contributions

From a theoretical standpoint, this dissertation contributes to the field by demonstrating how transfer learning can effectively address challenges in synthetic speech detection, particularly when data is limited. The research extends the understanding of deepfake detection and provides new insights into the effective use of pre-trained models for audio classification tasks. Practically, this work has important implications for industries involved in voice authentication, digital forensics, and cybersecurity. The findings highlight that the Transfer Learning FFNN model is not only computationally efficient but also highly accurate, making it suitable for deployment in real-world applications, such as detecting synthetic voices in media content.

The project also contributed to the growing body of knowledge on the differentiation between male and female voices in the context of synthetic speech detection. By incorporating this dimension, the study provides a more nuanced approach to voice classification, further enhancing the practical utility of deepfake detection systems.

7.3. Limitations of the Study

Despite the promising results, the study does have limitations. One of the key constraints is the size and diversity of the dataset. While the dataset used (1,956 audio clips) was substantial for the scope of this research, it may not fully capture the vast array of deepfake generation techniques, particularly those using newer, more advanced synthesis models. Future work should consider expanding the dataset to include a broader range of synthetic voices generated by diverse AI models. Additionally, the study focused on static audio features, which, although informative, do not account for the real-time detection challenges faced in live environments. Real-time detection remains an area for further investigation, along with the potential to explore more advanced deep learning models such as transformers for speech processing tasks.

7.4. Recommendations for Future Research

Future research could explore the incorporation of more advanced feature engineering techniques, particularly those that capture temporal dynamics and speaker-specific traits. Moreover, interdisciplinary collaborations between speech scientists, linguists, and machine learning researchers could lead to the development of more robust and effective detection frameworks. Investigating real-time deepfake detection on live

streaming platforms would also provide practical insights into how these systems can be deployed in dynamic environments.

Another recommendation is to explore lightweight, mobile-friendly deepfake detection models that can be deployed on devices for on-the-spot voice classification. This would expand the practical applications of the research, making it more accessible and usable in everyday scenarios.

7.5. Final Reflections

In conclusion, this dissertation provides a comprehensive exploration of deepfake audio detection, offering new theoretical insights and practical solutions to address the growing challenge of synthetic speech. By integrating transfer learning into the classification process, this research underscores the importance of leveraging pre-existing knowledge to overcome limitations posed by small datasets. The successful development of a model capable of classifying both the authenticity and gender of voices represents a significant step forward in the field of voice forensics.

While there are limitations that need to be addressed in future work, such as dataset size and real-time detection capabilities, the findings of this research have important implications for both academic research and real-world applications. As deepfake technologies continue to evolve, the need for accurate, scalable detection systems becomes ever more critical. This study serves as a foundation for future innovations in synthetic speech detection and underscores the importance of continued research and development in this area.

By bridging theoretical perspectives with practical applications, this dissertation contributes to the ongoing effort to combat the risks associated with deepfakes, offering a pathway for further advancements in both audio forensics and cybersecurity.

References

AlBadawy, E. A., Lyu, S., & Farid, H, 2019. *Detecting AI-Synthesized Speech Using Bispectral Analysis,* s.l.: s.n.

Alvarez, I., Reisslein, M., & Piamrat, K., 2020. *A Comprehensive Survey on E-Learning for Vision-Based Deepfake Detection.,* s.l.: s.n.

Anon., 2024. Spectral centroid.

Anon., 2025. Pitch (music).

Anon., 2025. Root mean square.

Anon., 2025. Zero-crossing rate.

Baevski, A., Zhou, H., Mohamed, A., & Auli, M., 2020. *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations.,* s.l.: s.n.

Bhagtani, K, 2025. *Speech forensics using machine learning. PhD dissertation,* s.l.: Purdue University.

Boyang Zhang, Jared Leitner, Sam Thornton, n.d. *Audio Recognition using Mel Spectrograms and Convolution Neural Networks,* s.l.: s.n.

Bradbury, J., n.d. Refining Pitch Analysis.

Guang Hua, Andrew Beng Jin Teoh, Haijian Zhang, 2021. *Towards End-to-End Synthetic Speech Detection,* s.l.: s.n.

Hainan Ren, Li Lin, Chun-Hao Liu, Xin Wang, Shu Hu, 2024. *Improving Generalization for AI-Synthesized Voice Detection,* s.l.: s.n.

Initiative, O. S., 2021. *MIT License - Allows open-source usage, modification, and distribution with attribution to the original authors..* s.l., Open Source Initiative.

Khalid, H., Tariq, S., Kim, M., & Woo, S. S, 2021. *FakeAVCeleb: A Novel Audio-Video Multimodal Deepfake Dataset.,* s.l.: s.n.

Kuiyuan Zhang, Zhongyun Hua, Yushu Zhang, Yifang Guo, and Tao Xiang, 2024. *Robust AI-Synthesized Speech Detection Using Feature Decomposition Learning and Synthesizer Feature Augmentation,* s.l.: s.n.

Librosa, 2025. Energy and RMSE.

Librosa, 2025. Spectral Centroid. *Spectral Features.*

Lundberg, S. M., & Lee, S.-I., 2017. *A Unified Approach to Interpreting Model Predictions.,* s.l.: s.n.

Migacz, S., 2017. *8-bit Inference with TensorRT.* s.l., s.n.

Nina LT So, Jacob A Edwards, Sarah MN Woolley, 2020. Auditory Selectivity for Spectral Contrast in Cortical Neurons and Behavior.

Sun, C., Jia, S., Hou, S., AlBadawy, E., & Lyu, S, 2023. *Exposing AI-Synthesized Human Voices Using Neural Vocoder Artifacts,* s.l.: s.n.

Xuan Hai, Xin Liu, Yuan Tan, Gang Liu, Song Li, Weina Niu, Rui Zhou, Xiaokang Zhou, 2024. *What's the Real: A Novel Design Philosophy for Robust AI-Synthesized Voice Detection,* s.l.: s.n.

Yamagishi, J., Wang, X., Todisco, M., Sahidullah, M., Patino, J., Nautsch, A., Liu, X., Lee, K. A., Kinnunen, T., Evans, N., & Delgado, H., 2021. *ASVspoof 2021: Accelerating Progress in Spoofed and Deepfake Speech Detection.,* s.l.: s.n.