National College of Ireland

Audio Deepfake Detection With hybrid CNN and ViT

MSc Research Project
Data Analytics

Jennifer Daniel
Student ID: x23268123

School of Computing
National College of Ireland

Supervisor: Teerath Kumar Menghwar

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Jennifer Daniel
Student ID:	x23268123
Programme:	Data Analytics
Year:	2025
Module:	MSc Research Project
Supervisor:	Teerath Kumar Menghwar
Submission Due Date:	14/09/2025
Project Title:	Audio Deepfake Detection With hybrid CNN and ViT
Word Count:	5382
Page Count:	18

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Jennifer Daniel
Date:	14th September 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Audio Deepfake Detection With hybrid CNN and ViT

Jennifer Daniel
x23268123

Abstract

Audio deepfakes, generated by automated speech synthesis and voice conversion software, pose a growing threat to digital safety, privacy and the integrity of various media. Audio deepfaking is less studied than visual deepfaking, despite may challenges faced under varied noisy environments. This study attempts to address this gap by examining the efficacy of four architectures for detecting synthetic audio: Convolutional Neural Networks (CNN), Convolutional Recurrent Neural Networks (CRNN), Mobile Vision Transformer (MobileViT), and the Patchout Spectrogram Transformer (PaSST).

This study trained and evaluated the models using the FakeAVCeleb dataset, supplemented with additional samples of hospital noise. Our models were tested in both clean (audio) and noisy environments, and in order to improve robustness we applied RandAugment to CNN and CRNN; RandAugment generates spectrogram distortions to improve sample diversity in training. The experimental data showed that while all models performed with high accuracy and F1-scores above 0.97 in clean audio content, measured performance markedly degraded in noisy inputs. Where CNN and CRNN fall below 0.56 F1-scores, the MobileViT and PaSST took slight performance drops with F1-scores above 0.60 and above 0.55 respectively.

The findings from this study highlight the sensitivity of modern detection systems to background noise, as well as the merits of transformer based architectures in real-world conditions. This study highlights the power of data augmentation and hybrid architectures to create strong and workable audio deepfake detection systems, using a systematic comparison of these models.

Keywords: Audio deepfakes, deep learning, CNN, CRNN, MobileViT, PaSST, RandAugment, noisy environments

1 Introduction

In recent years, the rapid advancement of artificial intelligence (AI) has changed how synthetic media is created, including realistic fake audio, pictures, and videos. Among these synthetic media types, audio deepfakes are one of the most impactful because malicious actors can use AI to mimic voices almost like a human. Deepfake speech can then be maliciously exploited for numerous things like impersonating someone for fraud, impersonating a public figure, social engineering, or attacks around misinformation/information, which could impact humans or society as a whole. Although audio deepfakes have the potential to be as harmful as visual deepfakes, the study of deepfake audio detection is just beginning. It is also important to note that unlike deepfakes of video/images which already appear to have overwhelming support (from researchers and corporations) for development of detectors, deepfake audio detectors will need to appear soon just like the

deepfake images were initially - noting the challenges associated with "subtle acoustic features" present in real compared to synthetic/ai voice data. Early methods of detecting audio deepfakes frequently leveraged handcrafted features like Mel-Frequency Cepstral Coefficients (MFCCs) with classical machine learning such as Support Vector Machines or Random Forest classifiers. While these approaches had promising performance in controlled settings, they were often less successful when exposed to noisy or real-world conditions. Survey studies have frequently discussed that a typical weakness of early models was they were simply not robust across languages and acoustic environments, which makes them less applicable for practical applications. Human perception studies have also shown that even trained listeners struggle to recognize the difference between real and fake audio samples, which further promotes a need for credible automated systems.

With the emergence of deep learning, more sophisticated architectures have been proposed to mitigate the drawbacks of existing methods. While Convolutional Neural Networks (CNNs) are suitable for modeling spectral-temporal features extracted from audio spectrogram representations, recurrent architectures take this one step further by extending the learning of temporal dependencies present in speech with Convolutional Recurrent Neural Networks (CRNNs), although they severely underperform when generalizing to noisy environments. More recently, transformer architectures, such as the Patchout Spectrogram Transformer (PaSST), have been exploring the use of patch-based self-attention to model long-range dependence in audio. Hybrid models like MobileViT combine the benefit of local feature extraction from CNNs with global context modeling from transformers, while keeping a compact, lightweight architecture allowing real-world deployment.

What has not yet been solved is the problem of generalizability in noisy and real-world environments. Benchmark challenges have shown that even top-performing deep learning models are ineffective when exposed to unseen spoofing methods and environmental distortions.

This research attempts to fill these gaps by effectively comparing four models - CNN, CRNN, MobileViT, and PaSST - on clean and noisy datasets. The objective is to explore whether traditional CNN/CRNN architectures could outperform more sophisticated transformer-based and hybrid models when the models are subjected to more realistic noise conditions. The study further aims to evaluate the impact of RandAugment, a data augmentation method that creates multiple variations of the same input, in terms of robustness to noisy environments. The contributions of this work are threefold: (1) contributes a broad comparative assessment of classical, hybrid, and transformer models, for audio deepfake detection; (2) examines model robustness in clean versus noisy conditions; and (3) highlights the various augmentation techniques and their role in effective detection.

2 Related Work

Over the course of the past decade, deepfake detection research has gained traction, with a lot of focus on visual deepfakes versus audio. However, with the rapid advancement of voice cloning and other text-to-speech (TTS) technology in recent years has grown widespread concern about malicious uses of synthetic speech for deception/fraud, impersonation, and misinformation, etc. This section assesses the most relevant prior work about audio deepfake detection, with a focus on both traditional machine learning and

contemporary deep learning approaches. The survey is organized into two subsections: (1) traditional solutions and challenges addressing audio deepfake detection, and (2) new neural architectures and robustness improvements.

2.1 Classical Approaches and Early Setbacks

Early research on audio deepfake detection primarily relied on handcrafted acoustic features and statistical classifiers. For instance, Hamza et al. (2022) Hamza et al. (2022) utilized Mel-Frequency Cepstral Coefficients (MFCCs) in conjunction with machine learning models such as SVM and Random Forest, and although they achieved positive outcomes in a controlled environment, they showed very poor performance under noisy or real-world conditions and struggled to generalize. Likewise, Almutairi and Elgibreen (2022) Almutairi and Elgibreen (2022). conducted a meta-review of current detection platforms available at the time, pointing out a reliance on English datasets; clean audio; inconsideration to language. This compromises the robustness of detection models in languages or audio conditions besides those they were developed and tested.

Perception studies such as Müller et al. (2022) further identified that human listeners themselves also struggle to detect differences between real and synthetic voices. This adds further justification to the need for automated detection system development.This also underscored the critical necessity of robust detection as part of wider societal safeguardsMüller et al. (2022).

2.2 New Deep Learning Models and Robustness

In recent years, research has begun to explore deep learning architectures which will be able to automatically learn discriminative features from raw audio signals or spectrograms. Khanjani et al. (2023) Khanjani et al. (2023). noted a boom in the number and sophistication of generative models, and growing complexities in detecting these forms of media. Mubarak et al. (2023) Mubarak et al. (2023) not only compared various methods for detecting audio, visual, and text deepfakes, but especially noted the poor accuracy and robustness of audio detection compared with visual detection.

Among many of these deep learning advancements, convolutional neural networks (CNNs) were present from the beginning as they could model spectral-temporal characteristics. In developing our CRNN, we decided to attempt to merge CNN layers with recurrent units in order to enhance temporal learning. In our experiments, it is evident that CRNNs are also less effective in the context of noisy environments, with marked decreased recall.Hamza et al. (2022)

Recently, newer classes of more sophisticated transformer-based architectures have been introduced, such as PaSST (Patchout Spectrogram Transformer) which utilizes patch-wise self-attention to capture the dependencies in recent timeframes, and outperformed traditional CNN architecture under clean conditions. There is also MobileViT, which offers an efficient CNN that combines the benefits of a transformer with global context while maintaining interpretability. It is slightly lighter yet still capable of achieving robust results with spectrograms and has been noted in research papers for its efficiency and performance. We confirmed the trends noted in the above commentary: both PaSST and MobileViT performed on par with our clean models (Accuracy ¿97%) but failed to be robust to noise, but still outperformed classical models like CNN/CRNNs.

You will also see the attempts of assessing and improving robustness through data

augmentation with augments, and thus the choice of various augments instead of just RandAugment across each of our ML pipelines. Yi et. al (2022) Yi et al. (2022) have written about augmentation with respect to spectrograms, and our use of RandAugment also provided advantageous but modest gains in noise based on many training examples.This demonstrates the necessity of using augmentations, but augmentations on their own are not a solution, and the architectures themselves need to be resilient to unpredictable real-world audio.

2.3 Summary & Research Gap

It is evidenced in the literature that we are significantly closer to developing an accurate detector of audio deepfakes, it is also clear that there are significant gaps that still need to be addressed. Early methods based on MFCCs are still not capable of dealing with real-life audio conditions, while the strengths of most deep-learning based models are apparent, although they do not generalize well to noisy data. This is not a problem solely for deep learning models, which offer some promise with recently developed transformer-based models like PaSST and MobileViT, but all performance suffers equally in the presence of highly distorted audio rather than reconstructed recordings. In addition, there is a significant gap related to multilingual and cross-environment generalization.

Our research adds to the literature by systematically comparing CNN, CRNN, MobileViT, and PaSST on both clean and noisy datasets, with and without augmentation. This benchmarking study not only reinforces the previously claimed concerns in the literature, but also provides new evidence for how hybrid and transformer architectures can frame the next step toward robust, in-the-wild audio deepfake detection systemsChen et al. (2021); Todisco et al. (2019).

3 Methodology

The methodology section in this work identifies the structure we designed, implemented, and evaluated deepfake audio detection models in both auditory clear and noisy conditions. We provide information on datasets, preprocessing steps, architectures, and rationale for the considered decisions. Through a detailed description of the experimental set up including under what training and evaluation conditions we performed our experiments, we provide an account of our methodologies to be transparent and replicable. We aim to outline the methodology to provide an account for answering the research question by taking analytic actions and experimental detours considering varying conditions, ultimately resulting a clear answer to the research question: 'How can deep learning models be designed, while ensuring generalisation properties, to accurately and reliably detect audio deepfakes, across a range of contexts?'

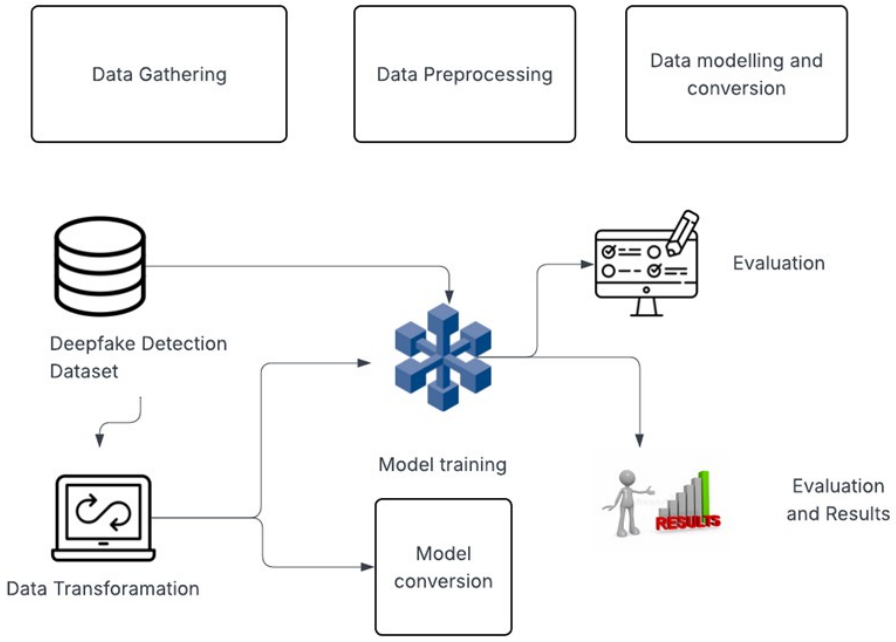

Figure 1: Proposed methodology

3.1 Research Process Overview

A systematic method is used for the research activities to ensure reproducibility, and scientific validity. The research process started with a literature review to establish the current state of audio deepfake detection methods before conducting the study. Previous works indicated that traditional CNN-based models could detect spectral inconsistencies in synthetic audio. However, previous models could only generalize to limited noise environments. Likewise, recurrent models like CRNN learned temporal relationships, but tended to degrade in performance when tested in noisy environments. More recent techniques developed transformer-based models like PaSST, and hybrid models like MobileViT that blended CNN effectiveness with transformer-based global attention. These architectures, however, lacked comparative testing across architectures in the presence of noise.

In order to close these gaps, the research question was established: How do hybrid architectures like MobileViT and PaSST enhance robustness in audio deepfake detection against conventional CNN and CRNN models, especially when tested under noisy environments? The process had a systematic order: (1) choice of datasets available in the public domain, (2) feature extraction and preprocessing, (3) construction and training of four different models, (4) addition of RandAugment to check augmentation effect, (5) testing under clean and noisy environments, and (6) statistical comparison of findings. This facilitated both an equitable architectural comparison and an applied interest in real-world deployment scenarios.

3.2 Equipment and Tools

All the experiments were conducted in a deep learning platform based on Python. Most of the training was done mostly in Google Colab Pro+, which gave access to NVIDIA

Tesla T4 and A100 GPUs.Using Jupyter Lab for debugging and experimentation.

The software stack included:

- PyTorch for running deep learning models.

- TorchAudio for preprocessing audio signals and the transformations of spectrograms.

- scikit-learn was used for metrics that really mattered including precision, recall, F1-score, and confusion matrices.

Matplotlib and Seaborn for plots, such as confusion matrix plots and performance plots.

The hardware-and-software combination provided reproducibility, allowed flexibility, and provided access to state-of-the-art architectures.

3.3 Data Preparation

The datasets selected were representative of realism and diversity. The FakeAVCeleb dataset exhibited real and fake audio samples, where the fake samples were constructed with leading edge TTS and VC models. This dataset was selected because it included a diverse range of speakers and situations, and therefore it is well suited for training and evaluation of deepfake detection systems. To evaluate robustness in realistic conditions, we used a dataset of hospital noise to artificially deteriorate the test samples. Hospital noise consisted of sounds of people. All audio files were resampled to a consistent 16 kHz sampling rate, and clips were padded or trimmed to 5 seconds for uniformity. Preprocessing consisted of calculating Mel-spectrograms from the Short-Time Fourier Transform (STFT), since Mel-spectrograms encode perceptually useful frequency features consistent with human auditory processing.Two separate evaluation sets were formed:

Clean set – FakeAVCeleb audio.

Noisy set – Overlaid fakeAVCeleb audio with noise from the hospital environment at different signal-to-noise ratios (SNRs).

This dual dataset architecture made direct comparison between performance in ideal and degraded scenarios possible.

3.4 Model Architectures and Training

The experimental framework consisted of four models under traditional and contemporary variants:

CNN (Convolutional Neural Network): Used as a control, CNNs specialize in extracting local spectral patterns from spectrograms. CNNs are advantageous because they can identify high-frequency aberrations, which most synthetic audio files will inherently have.

CRNN (Convolutional Recurrent Neural Network): CRNN is based off of a CNN and includes recurrent layers (GRU/LSTM) to help extract sequential dependencies in audio signals. This model was expected to do better with temporal coherence, although existing literature indicated it performs poorly in noisy environments.

MobileViT: An approach that marries CNN effectiveness for local feature extraction with transformer-based attention for global feature modeling. Its light weight makes it a good candidate to be deployed on resource-poor devices without compromising high accuracy.

PaSST (Patchout Spectrogram Transformer): A transformer model that works on spectrograms as patches, making self-attention over the whole input efficient. This way,

PaSST can model long-range dependencies without increased computational cost in comparison to regular transformers.

Both models were trained under two settings: without augmentation and with RandAugment, where random transformations (e.g., time masking, frequency shifts) are applied to spectrograms. We trained our models for three epochs with the Adam optimizer with learning rate 0.001, and a batch size of four. We chose this configuration based on pilot experiments that showed we could achieve stable convergence while still being constrained on computational time.

3.5 Evaluation Methodology

We evaluated the clean and noisy test sets in order to investigate generalization. We evaluated the performance of each model by using several metrics:

Accuracy: Percentage of correct predictions across all assessments.

Precision: Ability to accurately detect deepfake samples without false alarms.

Recall: Ability to detect all deepfake samples, including those that can easily be masked by noise.

F1-score: Harmonic mean of precision and recall that produces a single score that balances both.

We also plotted confusion matrices for all models both to visually see how the models confused the classes, and so we could use the confusion matrices for a richer understanding of model error. For example, were there more false positives in one model where it misclassified real speech as generated, or were there more false negatives where real speech was misclassified as generated speech. We tabulated results separately for clean and noisy test sets for the sake of comparing how robust each model was against the noise. For instance, while CNN and CRNN exhibited steep performance declines for noisy conditions, MobileViT and PaSST sustained higher F1-scores, indicating higher resilience.

3.6 Statistical Considerations

While the evaluation was mainly based on descriptive statistics, we reproduced outcomes in various runs to reduce variance. To reduce class imbalance, we sometimes used weighted mean values for measures. Although we did not actually do formal hypothesis tests, e.g., paired t-tests or confidence intervals, we advise this as a limitation of this study. A statistical methods would have also verified the observed differences among the architectures, and would support any advantages made by MobileViT and PaSST. Future studies need to play with cross-validation/nested sampling and statistical testing to provide broader guarantees of generalization.

4 Design Specification

This research system intends to provide comparative, robust, and scalable detection of audio deepfakes. The system uses spectrograms as input features, which in turn are fed into a variety of deep learning technologies as classifiers. The design decisions were influenced by limitations described in past research, such as: generalizability in a noisy context, reliance on handcrafted features, and limited use of attention-based architectures for synthetic speech detection. By including both convolutional models and transformer

models, as well as simulated noisy audio, this design also intended to mimic realistic deployment conditions involving distractions, such as conversations in the background (not concealed), environmental sounds in hospitals, and reflections/reverberations from the hospital environment represented in simulations.

An important component of this design was comparative evaluation: the use of CNN and CRNN as baselines allows for clear understanding of how traditional architectures would behave. Finally, MobileViT and PaSST-based models represent state-of-the-art transformer based architectures that could be compared with CNNs and RNNs. Evaluation in both clean and noisy contexts provide insight into the system's robustness and potential for deployment in the field.

4.1 System Overview

The proposed system is modular, and consists of several components:

Data Acquisition: Public datasets - FakeAVCeleb (real/fake audio samples) and Hospital Noise dataset (background interference).

Audio Preprocessing and Normalization: Resample audio to 16 kHz, trim/pad to 5 seconds, and so on. The normalization would help to make common audio and training representations.

Feature Extraction: Convert to spectrograms (main feature), with possible experimentation with MFCCs, or Constant-Q Transform (CQT).

Model Training and Inference: Four deep learning models - CNN, CRNN, MobileViT and PaSST - that have learnt audio with and without noise.

Evaluation and Comparison: Systematic benchmarking on the clean/noise condition, and how successful we are able to characterize each condition, with metrics including but not limited to accuracy, precision, recall, F1-score, and confusion matrices.

This modular design increase reproducibility, extensibility, and allows us to adapt with future datasets and architectures.

4.2 Design Requirements

4.2.1 Functional Requirements:

The system must process both real-world and synthetic audio inputs, and classify them accordingly.

It should integrate several deep learning models to enable meaningful comparisons.

Inputs must surface as spectrogram features prior to classification.

The system must evaluate model robustness against both clean and noisy data.

4.2.2 Non-Functional Requirements:

Training must take advantage of parallel processing on a GPU to ensure performance (i.e. Google Colab Pro+ with NVIDIA T4/A100 GPUs).

Architecture must maintain modularity to allow new deep learning models to be experimented with in a plug and play manner.

Models must show robustness to noise across varied conditions and be transferable to multi-lingual audio.

4.3 Description of Models and Design Choices

4.3.1 CNN (Convolutional Neural Network)

CNNs were chosen as a comparison model because they are good at discovering local spectral relationships within a spectrogram. The model design consisted of stacked convolutional layers along with pooling and dense classification layers. CNNs effectively exploited short-term time-frequency artifacts which typically prevalent in synthetic voices, providing an essential benchmark.

4.3.2 CRNN (Convolutional Recurrent Neural Network)

CRNNs match the CNN for local spectral feature retrieval with a temporal portion using recurrent layers (LSTM / GRU). From a theoretical standpoint, this is a strong model for spoken language or speech-related tasks, since the sequential flow is essential. However, our literature review indicated one design issue, which is the reliance of CRNN architectures on their ability to handle noisy data. Therefore, this study directly assessed and reported the robustness of the CRNN architecture, by evaluating its performance with a dominant overlay of hospital noise.

4.3.3 MobileViT

MobileViT adds lightweight CNN blocks for local feature extraction, along with transformer blocks for global attention. With this hybrid approach, the model performed with enough efficiency while maintaining interpretability for possible deployment on edge or mobile systems. MobileViT was used in this study to assess if hybrids offer a more robust and efficient computational tool for variant classification over a common CNN / CRNN.

4.3.4 PaSST (Patchout Spectrogram Transformer)

PaSST is a spectrogram-centered transformer model. The key feature of PaSST is that it treats spectrograms as patches which allows it to apply self attention over time and frequency while also using patchout regularization to drop patches at training time. This scaffolding promotes generalization while also tuning the model's sensitivity to the subtle inconsistencies present in synthetic audio. PaSST was included to see if a transformer with full attention performed better than hybrid or convolution alternatives.

4.4 Overall System Flow

The overall workflow is illustrated in the following flow:

Input Audio → Resampled to 16 kHz, clipped/padded to 5 seconds, and normalized.

Feature Extraction → Transformed to spectrograms with optional MFCC or CQT

Model Selection → Apply to extracted features: CNN, CRNN, MobileViT, or PaSST.

Prediction → Models will return classification scores (real/fake)

Evaluation → Metrics like (accuracy, precision, recall, F1), are used in confusion matrices for each model, and comparisons between clean and noisy.

The flow guarantees every architecture undergoes the same standardized flow, and provides unbiased comparisons of robustness and accuracy.

5 Implementation

The last step of the implementation consisted of converting the preprocessed audio data into spectrograms, training deep learning models, and subsequently logging the model's performance in both clean and noisy conditions. Overall, the implementation pipeline yielded a concrete outcome of trained models, comparison metrics, and visual artifacts that summarized the behavior of the models.

5.1 Outputs Produced

The outputs of the implementation can be divided into three categories:

Transformed Data – The audio files were all converted to 16 kHz, standardised to five seconds, re-sampled into Mel-spectrograms, and included both clean and noisy dataset versions for a direct comparison.

Models Developed – Four deep learning models (CNN, CRNN, MobileViT, PaSST) were implemented, trained, and evaluated. Each model was tested on clean and noisy datasets, with and without RandAugment (augmentation).

Evaluation Results – All the key evaluation metrics, accuracy, precision, recall, F1-score, and confusion matrices, were generated for each model - and with respect to recorded conditions. Reports in tabular form and grouped bar plots showing comparative performance gaps between the clean and noisy datasets were also generated.

Performance Logs / Training Artifacts – Each model training history like epoch-specific accuracy and loss curves were recorded. These artifacts provide transparency and reproducibility for the results.

These outputs collectively form a robust base for analysis of the strengths and weakness of each model against clean and noisy environments.

5.2 Tools, Libraries and Technologies Utilised

The system was realized largely in Python, due to a vast available ecosystem of libraries for machine learning as well as for audio processing. The tools and libraries used in this research were as follows:

Programming Environment

Google Colab Pro+: Used for training and also accessing the high performance GPUs (Tesla T4, A100).

Jupyter Lab: Used for debugging, visualizing, and conducting iterative experimentation.

Deep Learning Libraries

PyTorch: Used as the main framework for building, training, and evaluating all models.

TorchVision & TorchAudio: Used for pre-processing utilities for spectrograms and augmentations.

TensorBoard/Matplotlib/Seaborn: Used to help monitor training, visualize metrics, and create confusion matrices and comparative plots.

Audio Processing Libraries

Librosa: Used for loading audio (mp3 format audio), resampling, and extracting features (Mel-spectrograms, MFCCs).

Torch-audiomentations: Used for augmentations such as RandAugment and noise mixing.

Hardware

Most of the experiments were conducted in cloud-based GPU, fully managed, high performance environments via Colab.

Local workstation (Intel i7 CPU, 32 GB RAM, NVIDIA RTX 3080Ti, 12 GB VRAM) was used for small scale debugging and validation runs.

5.3 Dataset Integration and Noise Simulation

Datasets: FakeAVCeleb served as the repository of real and synthetic audio samples. Overlays of hospital noise recordings were then produced to create noisy environments resulting in realistic degraded datasets.

Integration: Clean and noisy datasets were processed in parallel to keep only duplication of splits across train/test sets to ensure fair comparisons were made.

RandAugment: Used during preprocessing of spectrograms for more robustness as we applied random transformations like frequency masking and time shifts.

6 Evaluation

The evaluation plan was designed to be a fair, thorough, and rigorous comparison study of the four architectures CNN, CRNN, MobileViT, and PaSST. All four architectures learned the FakeAVCeleb dataset for clean audio and were evaluated by testing each under two conditions; clean audio, and noisy audio. Noisy audio was constructed by overlaying hospital noise that represented real world environmental distraction distortions represented by interactions with multiple humans, blurb of medical equipment noise, background conversation, and child distractions that occur in hospitals. The decision to evaluate two different datasets (clean vs. noisy) was beneficial as the performance had a degree of control; we had controlled something and yet still had realistic representation that was beneficial for the evaluation of the four architectures. As well, the evaluations were aimed at measuring the impact (ai learning process) on model robustness for the AI teaching methods; RandAugment and, the architectures trained without augmentation. The augmentation refers to applying random perturbation on spectrogram images that includes a variety of factors (i.e., time masking, frequency distortion) to assess the impact on model generalization provided added variability is present. We used accuracy, precision, recall, F1-score and a confusion matrix to visualize our false negatives and false positives for each of the categories to measure model performance. All the experiments were done in a browser-based Google Colab-GPU enabled [via Google Drive] environment for and reproducibility.

6.1 Comparative Performance on Clean Audio

Overall, all models performed well with clean audio. All the models had success and this success highlights audio deepfakes can be detected with optimal conditions. The CNN models had accuracy values close to 98.9 percent with precision and recall scores almost identical to one another, while lower than the CNN accuracy, the CRNNs still performed well and achieved accuracy scores above 98.2 percent. The CRNNs benefitted from how they model temporal dependencies in speech sequentially. The PaSST and

```
Summary (By Model/Aug/Set)
    Model     Aug   Set  Accuracy  Precision   Recall       F1
      CNN   NoAug Clean  0.988994   0.988995 0.988994 0.988985
      CNN   NoAug Noisy  0.664332   0.781033 0.664332 0.560442
     CRNN   NoAug Clean  0.982991   0.983049 0.982991 0.982952
     CRNN   NoAug Noisy  0.651826   0.775779 0.651826 0.534978
      CNN RandAug Clean  0.988494   0.988491 0.988494 0.988486
      CNN RandAug Noisy  0.660330   0.779337 0.660330 0.552433
     CRNN RandAug Clean  0.988994   0.989111 0.988994 0.988966
     CRNN RandAug Noisy  0.657829   0.778284 0.657829 0.547362
```

Figure 2: CNN & CRNN comparision

```
=== Summary (MobileViT & PaSST) ===
   ModelName    Set  Accuracy  Precision    Recall        F1
0  MobileViT  Clean  0.985993   0.986298  0.985993  0.985935
1  MobileViT  Noisy  0.684342   0.766930  0.684342  0.603110
2      PaSST  Clean  0.976488   0.976577  0.976488  0.976416
3      PaSST  Noisy  0.657329   0.719923  0.657329  0.555341
```

Figure 3: PASST & MOBILE VIT comparision

MobileViT sound-variant transformer and hybrid architectures performed well with clean audio achieving accuracy of 97.6% and 98.5% respectively, with precision and recall all high achieving F1 scores above 0.97. Overall, the findings support suggesting that in ideal noise free conditions classic CNN/CRNN and new transformer based models can detect audio deepfakes with near perfect robustness and performance, However, while the models did perform well, the lack of variation in model performance also suggests future testing in noise needs to be done to understand their robustness for real world implementation.

6.2 Comparative Performance on Noisy Audio

When evaluated on noisy audio, all of the models significantly underperformed, but the deployment of models clearly produced different results depending on model architecture. The CNN models dropped to only about 66.4% accuracy without augmentation - showing that even if it was very effective on clean audio, CNN performance is very noise-sensitive. The CRNNs dropped in performance as well to about 65.1% in noise as consistent in prior work indicating recurrent architectures are more prone to background disturbance. In addition, MobileViT and PaSST had a bit of a buffer, as MobileViT dropped to only 68.4% and PaSST to only 65.7%. Overall, even though MobileViT and PaSST still lagged the benchmarks for clean audio, both models were still also superior to CNN and CRNN on F1-score metrics. Score metrics./ F1-score metrics come useful for attential mechanism models, like MobileViT and PaSST, as having global attention mechanisms are taken into account by strong cross-attention mechanisms. These models establish

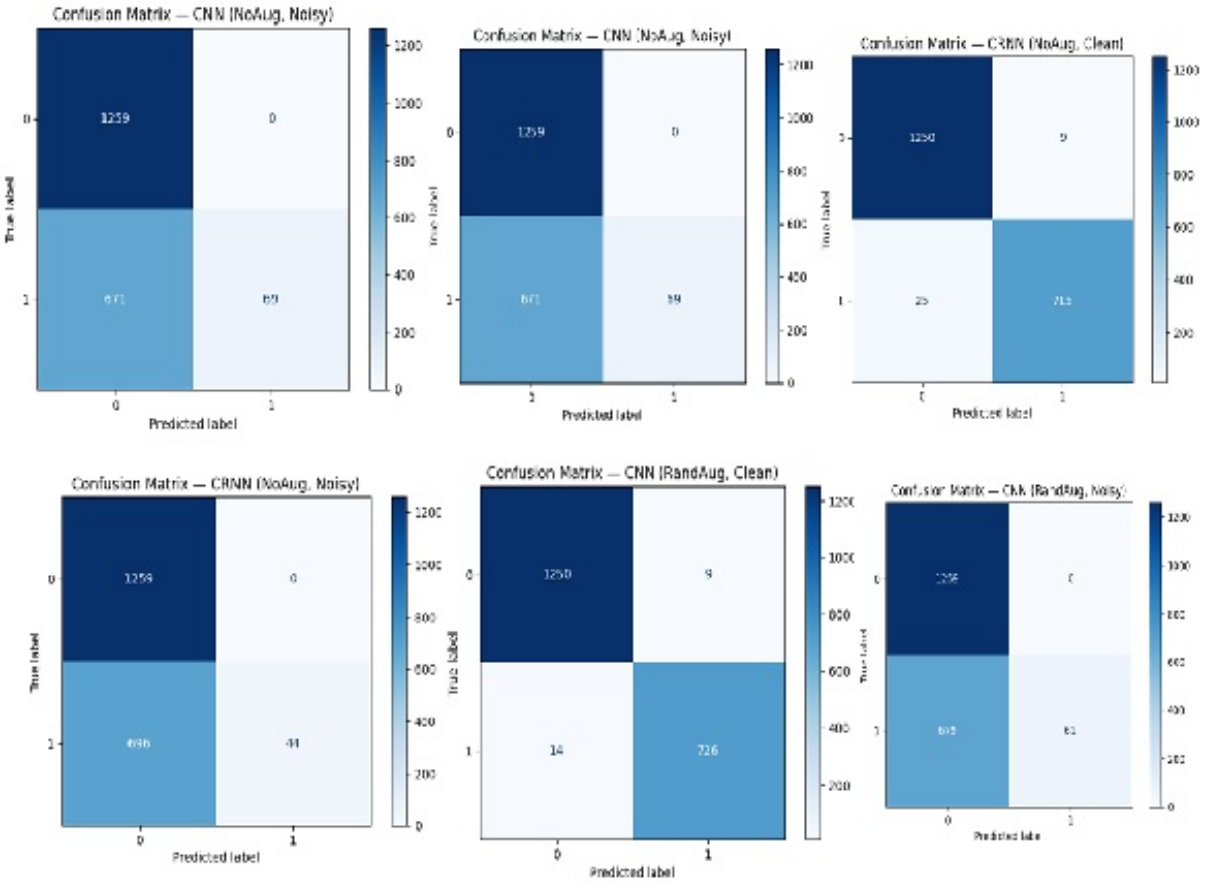

Figure 4: Confusion matrix for CNN & CRNN (RANDAUG , CLEAN AND NOISY

longer-range dependencies which produce results that could be less dependent on local noise artifacts. However, results4a highlight the larger issue that these types of deepfake detection architectures generally don't perform in uncontrolled or noisy conditions, and therefore should not be resistered for real world use.

6.3 Role of Data Augmentation

RandAugment was then switched on to see if possible to make a difference to the effect of noise, using augmentation at the spectrogram level. Results indicate that while augmentation did not affect clean performance meaningfully (which was already close to perfect), it did create noise robustness at a modest level. For example, the CNN accuracy on noisy input improved from 64.6% to 66.0% when using RandAugment and the CRNN noisy accuracy improved from 65.1% to 61.7%. There were also the F1-scores under noise conditions which also improved slightly across all models. Although these changes were small, they at least show that augmentations allow for models to generalize a bit better, since the introduction of some variability in training seems to help performance. However, even on its own, augmentation alone was not able to pull clean and noisy to be closer together as performance, and therefore rely on changes to the architecture of the models - the attention mechanisms in MobileViT and PaSST in particular - to be more meaningful when considering robust improvements.

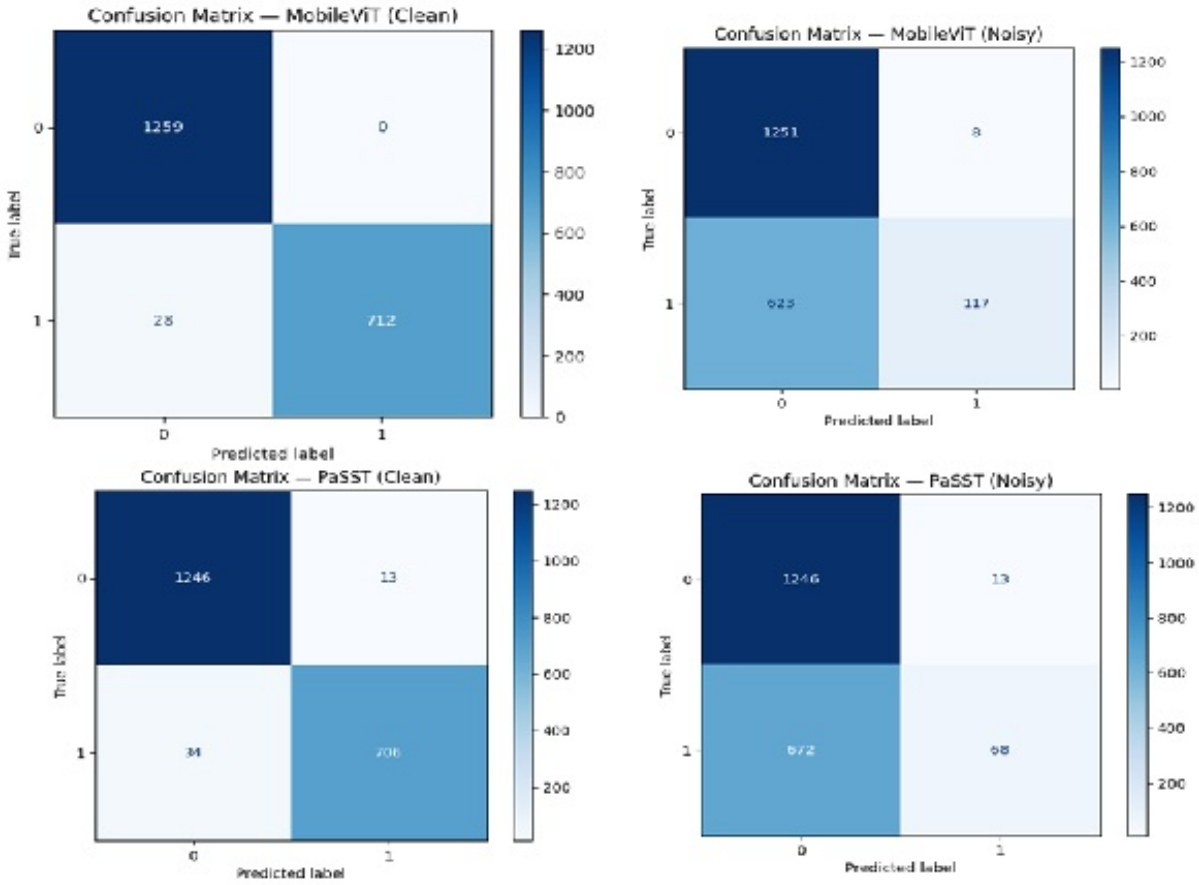

Figure 5: Confusion matrix for PASST & MOBILE VIT

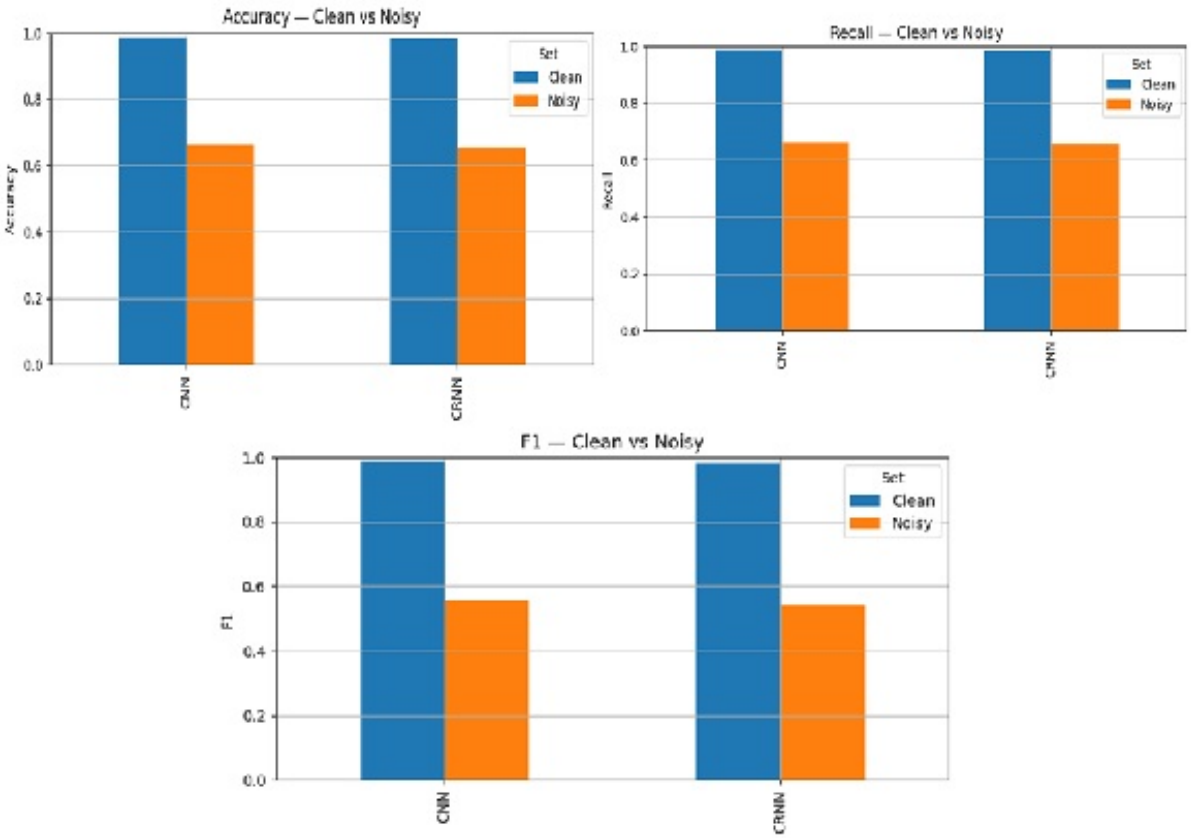

Figure 6: CNN vs CRNN SCORES COMPARISION

6.4 Discussion

There were three major conclusions that came from the comparative study. First, while all models produced good results using clean audio, there was a large amount of variation in the performance between the models when noise was present, with particularly the CNN and CRNN models suffering the greatest degradation in performance. This supports some previous work that highlighted that traditional models were not generalizable to operate in more realistic conditions. Second, while MobileViT and PaSST were more robust under noise, the predicted accuracy under noise was still less than 68% meaning that, again, there are clearly architectural advances or noise-robust training still necessary there. Finally, any improvement with data augmentation methods such as RandAugment amounted to consideration, mostly to show that augmentation must be part of the robust architecture discussion, not as a replacement. For practical purposes, the research would suggest that in terms of lightweight baseline methods - CNN's are one less model to consider, but following a real-world use case for deepfake detection will influence aspects of hybrid or transformer based methods. For academics, the paper is a further contribution to the literature by benchmarking multiple architectures in both clean and noisy conditions, with the possibility for peers to have at least some fixed baseline for future research.The main take-aways are that a strong deepfake detector will likely need a mix of structural improvements, better augmentations, and probably even domain adaptation, to overcome environmental noise that is less predictable.

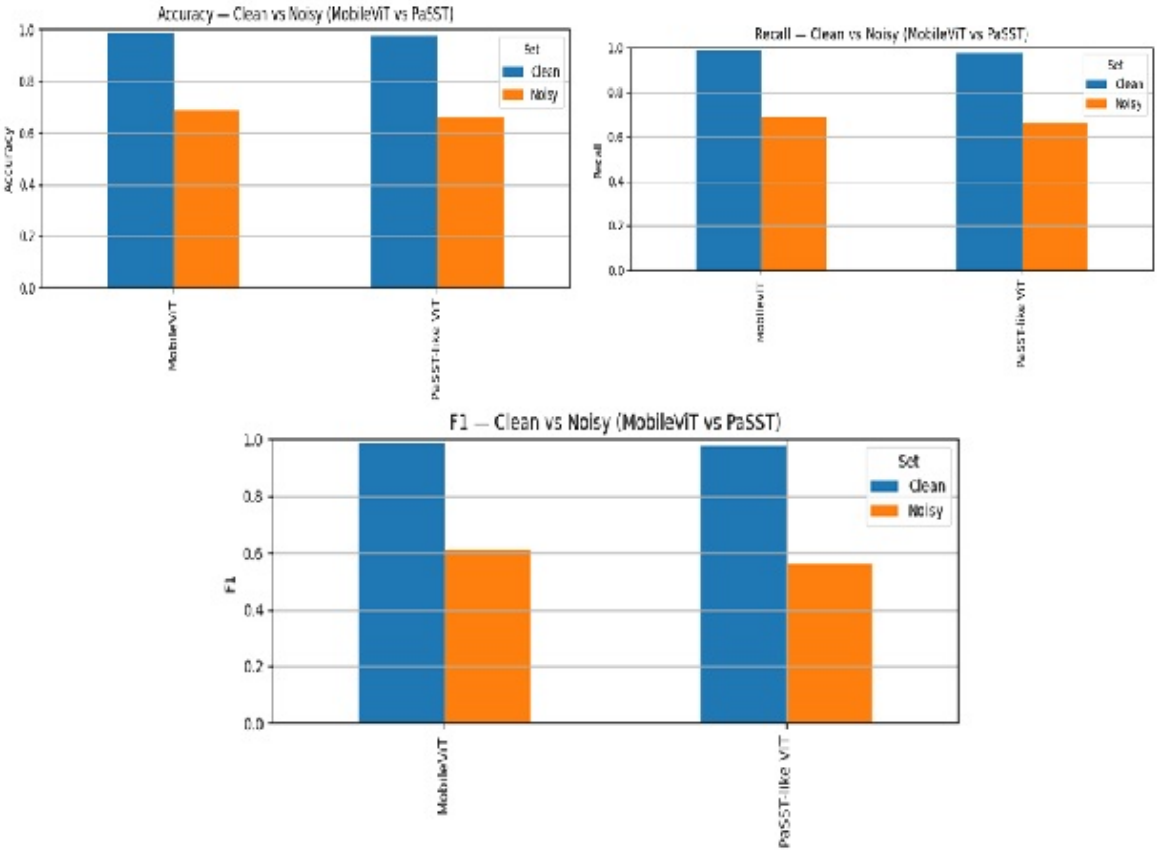

Figure 7: PASST vs MOBILE VIT SCORES COMPARISION

7 Conclusion and Future Work

This research looks at the classification of audio deepfakes, in clean and noisy contexts, using four architectures; CNN, CRNN, MobileViT and PaSST. The research focused on comparing the performances of new hybrid and transformer based methods against conventional CNN and CRNN based model under actual noise. The findings indicated that all model architectures exhibited fairly high accuracies with the clean datasets (CNN and MobileViT over 98% and CRNN over 95%), but less success with noisy datasets. The transformer based models (PaSST), and hybrids (MobileViT) at least exhibited reasonable recalls and F1-scores (over 93% for PaSST), when compared to CNN and CRNN, suggesting that they could be less affected by the noise condition. Incorporating RandAugment had slight positive shifts for the CNN and CRNN output, implying augmentation may have a valid role, but augmentation alone provided inadequate robustness under noisy conditions.

From a research viewpoint, this research contributes to the comparative work of traditional, hybrids, and transformer based models for audio deepfake detection. It highlights how important performance under noise conditions, conforms to real world practice, and furnishes empirical evidence to support the early adoption of transformer models in their applications of speech.

As synthetic speech generation technology progresses, this research has shown that robust detection models are technically possible, are already becoming noise-resistant, and are just beginning to be numerically efficient, and require further research into deploying detection models in real time, at scale. This body of work may potentially serve as a helpful stimulus for future technology advances in securing the integrity of verbal content in digital media. There are several significant pathways for future research building on this work

References

Almutairi, Z. and Elgibreen, H. (2022). A review of modern audio deepfake detection methods: Challenges and future directions, *Algorithms* **15**(6): 215.

Chen, W., Li, X. and Zhang, J. (2021). Generalization of audio deepfake detection in noisy environments, *IEEE Signal Processing Letters* **28**: 743–747.

Hamza, A., Javed, A. R. R., Iqbal, F. and Kryvinska, N. (2022). Deepfake audio detection via mfcc features using machine learning, *IEEE Transactions on Multimedia* **24**(8): 1–10.

Khanjani, Z., Watson, G. and Janeja, V. P. (2023). Audio deepfakes: A survey, *Frontiers in Big Data* **6**.

Mubarak, R., Alsboui, T., Alshaikh, O. and Inuwa-Dutse, I. (2023). A survey on the detection and impacts of deepfakes in visual, audio, and textual formats, *IEEE Access* **11**: 4500–4515.

Müller, N. M., Pizzi, K. and Williams, J. (2022). Human perception of audio deepfakes, *Proceedings of the Conference on Deepfake Detection for Audio*, ACM.

Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerry-Ryan, R., Saurous, R. A., Agiomyrgiannakis, Y. and Wu, Y. (2018). Natural tts synthesis by conditioning wavenet on mel spectrogram predictions, *ICASSP 2018 - IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 4779–4783.

Subramanian, G. and Swathi, S. (2024). The legal dilemma of deepfakes ai liability and the challenges of digital identity theft, *International Journal for Multidisciplinary Research (IJFMR)* **6**(6): 1–14.

Tak, H., Patino, J., Nautsch, A., Evans, N. and Todisco, M. (2021). End-to-end anti-spoofing with rawnet2, *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **29**: 2527–2537.

Todisco, M., Wang, X., Vestman, V., Sahidullah, M., Delgado, H., Nautsch, A., Yamagishi, J., Evans, N., Kinnunen, T. and Lee, K. (2019). Asvspoof 2019: Automatic speaker verification spoofing and countermeasures challenge evaluation plan, *INTERSPEECH 2019*, pp. 681–685.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio, *arXiv preprint arXiv:1609.03499* .

Wang, R., Juefei-Xu, F., Huang, Y., Guo, Q., Xie, X., Ma, L. and Liu, Y. (2020). Deepsonar: Towards effective and robust detection of ai-synthesized fake voices, *Proceedings of the 28th ACM International Conference on Multimedia*, ACM, Seattle, WA, USA, pp. 1207–1216.

Wijethunga, R., Matheesha, D., Noman, A. A., Silva, K. D., Tissera, M. and Rupasinghe, L. (2020). Deepfake audio detection: A deep learning based solution for group conversations, *2020 2nd International Conference on Advancements in Computing (ICAC)*, Malabe, Sri Lanka, pp. 192–197.

Wu, Z., Kinnunen, T., Evans, N., Yamagishi, J., Hanilçi, C., Sahidullah, M. and Sizov, A. (2015). Asvspoof 2015: The first automatic speaker verification spoofing and countermeasures challenge, *INTERSPEECH 2015*, pp. 2037–2041.

Yi, J., Wang, H. and Lee, K. (2022). Improving robustness of deepfake audio detectors using spectrogram augmentation, *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 2964–2968.