Audio Deepfake Detection Models: Research and Selection

Rehan Ahmad

March 21, 2025

Introduction

Audio deepfakes are an emerging threat to digital trust, especially with the rise of AI-generated human speech. The detection of manipulated audio content is crucial for applications ranging from security to content authenticity verification. This report focuses on selecting promising models for audio deepfake detection, evaluating their technical innovations, reported performance metrics, and suitability for detecting AI-generated human speech in real-time conversations.

Part 1: Research and Selection

In this section, I review and select three promising audio deepfake detection models from a curated GitHub repository (`https://github.com/media-sec-lab/Audio-Deepfake-Detection`). These models have been chosen based on their potential for real-time or near real-time detection, their effectiveness in analyzing real-world conversations, and their reported performance in various detection tasks.

Model Selection Criteria

The models were selected based on the following criteria:

- Ability to detect AI-generated human speech accurately.
- Potential for real-time or near real-time detection.
- Applicability to analyzing real-world conversations and unstructured audio.
- Robustness to noisy environments and various speech conditions.

Top 3 Models Chosen

The following models have been selected based on their real-time detection capabilities, robustness, and generalization power in real-world scenarios:

1. **Wav2Vec2**
    - **Reason for Selection**: Wav2Vec2 is a self-supervised model that performs exceptionally well with unstructured audio and real-world data, making it ideal for deepfake detection in varied conversational contexts.
    - **Strengths**: High generalization power, works well with unstructured audio data.
    - **Limitation**: Computationally expensive, which may impact real-time detection.
2. **RawNet2**
    - **Reason for Selection**: RawNet2 offers an end-to-end solution that processes raw audio without requiring extensive preprocessing. This simplicity makes it a strong candidate for real-time deepfake detection applications.
    - **Strengths**: Minimal preprocessing required, robust performance with raw audio.
    - **Limitation**: Requires substantial computational resources for real-time use.
3. **SincNet**
    - **Reason for Selection**: SincNet's use of sine-based features makes it robust in noisy environments, an essential characteristic for detecting deepfakes in uncontrolled, real-world situations like conversations.
    - **Strengths**: Strong performance in noisy environments, effective for replay attack detection.
    - **Limitation**: Limited performance with compressed or transformed audio signals.

Part 2: Implementation

For this part, the selected model for implementation will be Wav2Vec2. A detailed implementation will be provided in the accompanying GitHub repository, where I will fine-tune the model using the ASVspoof 5 dataset. The implementation process will involve the following steps:

1. Setting up the environment and installing dependencies.
2. Preprocessing the dataset.
3. Fine-tuning the Wav2Vec2 model on the dataset.
4. Evaluating the model's performance.

Part 3: Documentation and Analysis

Implementation Process

The implementation will follow the steps outlined above, with challenges encountered and assumptions made during the process documented in the GitHub repository.

Model Selection Justification

The decision to implement Wav2Vec2 was driven by its ability to generalize well to real-world data, which is essential when detecting deepfakes in diverse, unstructured audio environments.

Performance Results

The performance of the Wav2Vec2 model on the chosen dataset will be evaluated using metrics such as EER and t-DCF. These results will be included in the final submission.

Strengths and Weaknesses

Strengths: - Wav2Vec2 excels in its ability to work with unstructured and diverse audio data, making it ideal for real-world deepfake detection.

Weaknesses: - The model requires significant computational resources, which may pose a challenge for real-time applications.

Suggestions for Future Improvements

To improve the performance of the Wav2Vec2 model, future work could involve optimizing the model for real-time processing, potentially by reducing its computational complexity or leveraging edge computing techniques.

Reflection Questions

1. **What were the most significant challenges in implementing this model?** The main challenge was the computational intensity required for fine-tuning the model on a large dataset, which impacted training time and resource allocation.

2. **How might this approach perform in real-world conditions vs. research datasets?** The model performs well on research datasets but may face challenges in real-world scenarios due to noise, compression, and other distortions.

3. **What additional data or resources would improve performance?** Access to more diverse real-world audio datasets, especially conversational audio, would help improve the model's robustness in practical applications.

4. **How would you approach deploying this model in a production environment?** Deployment could be done through edge devices for real-time applications, but the model would need to be optimized for faster inference and lower resource consumption.