Momenta - Audio Deepfake Detection Take-Home Assessment

Ashish Rana

PART 1: Research & Selection

Review the GitHub Repository

Review the following GitHub repository, which contains a curated collection of papers and resources on audio deepfake detection: Audio Deepfake Detection GitHub Repository.

Repository Overview

Purpose: This repository collects state-of-the-art (SOTA) methods for detecting synthetic/deepfake audio generated by Text-to-Speech (TTS), Voice Conversion (VC), or other speech synthesis models.

Main Categories:

- Research Papers
- Datasets
- Detection Methods

Key Contents

Research Papers: The repository lists papers from top publishers like IEEE and others, covering diverse directions such as:

- Detection of synthetic speech (e.g., ASVspoof challenges).
- Generalized deepfake audio detection.

Datasets: The repository contains datasets such as:

- AVSspoof
- SpoofCeleb

Detection Methods:

- Traditional Approaches: LFCC, Spectrogram Analysis

- Deep Learning: CNN, LSTM, Transformer
- End-to-End Models: ResNet-based classifiers

Conclusion

This repository is a valuable resource for anyone working on audio deepfake detection, especially for literature review purposes, and is useful for surveying SOTA methods.

PART 2: Model Selection

Identifying Promising Models for Our Use Case

For the task of detecting AI-generated human speech, especially in real-time or near-real-time detection scenarios, I have identified the following models as most promising:

I will use Equal-Error-Rate (EER) and Tandem Detection Cost Function (t-DCF) as evaluation metrics for model selection.

EER (Equal-Error-Rate)

: This metric measures the threshold where the False Acceptance (FA) and False Rejection (FR) rates are equal. It helps evaluate how well a system distinguishes between real and AI-generated speech.

t-DCF (Tandem Detection Cost Function)

: This metric combines spoof detection with speaker verification error. It is critical for real-world deployment, where both imposter speech and identity matter.

Model 1: Dual-Branch Network

Cite the original paper: End-to-end dual-branch network towards synthetic speech detection.

Key Technical Innovation: It has a dual-branch architecture, where Branch-1 focuses on Linear Frequency Cepstral Coefficients (LFCC), and Branch-2 focuses on Constant-Q Transform (CQT). LFCC detects unnatural glitches in synthetic speech, while CQT detects unnatural pitch.

Reported Performance Metrics:

- EER: LA: 0.80
- t-DCF: LA: 0.021

Why this approach is promising for our needs: The dual-branch network effectively detects synthetic speech and voice cloning. It is a lightweight variant that can reduce inference time and is robust to background interference due to multi-scale feature learning.

Limitations: It has not been tested on PA (replay) scenarios and requires large-scale training data. The dual-branch architecture increases the number of parameters, which may require quantization for edge deployment.

Model 2: ResMax

Cite the original paper: ResMax: Detecting voice spoofing attacks with residual network and max feature map.

Key Technical Innovation: ResMax is a residual network that avoids training problems by using skip connections, combined with a max feature map (MFM) that highlights the most important features in audio to identify fakes.

Reported Performance Metrics:

- EER: LA: 2.19; PA: 0.37
- t-DCF: LA: 0.060; PA: 0.009

Why this approach is promising for our needs: MFM is effective at spotting neural vocoder artifacts, and with quantization, it achieves low latency on GPUs.

Limitations: Edge deployment requires pruning, and the model struggles with high-quality voice conversion.

Model 3: Voice Spoofing Countermeasure for Logical Access Attacks Detection

Key Technical Innovation: This model uses a Large Margin Cosine Loss Function to maximize the variance between genuine and spoofed classes while minimizing intra-class variance. It also incorporates FreqAugment, a layer that randomly masks adjacent frequency channels during training, improving generalization.

Reported Performance Metrics:

- EER: LA: 1.81
- t-DCF: LA: 0.052

Why this approach is promising for our needs: It is a robust, end-to-end deep learning framework for voice spoofing detection from a wide variety of unknown TTS and VC systems, offering high accuracy. It is also lightweight.

Limitations: The model has not been tested on PA (replay) scenarios.

PART 3: Documentation & Analysis

Challenges Encountered

While implementing the model, I did not face major challenges, as I used the provided GitHub repository. However, since they used MATLAB for LFCC extraction, I had to implement this in PyTorch. Additionally, I had to develop my own data generator, where I passed audio signals as inputs and extracted LFCC, CQT, labels, and fake labels (using torchaudio).

As I am new to this domain, I faced difficulties in understanding terminology such as EER, t-DCF, MEL Spectrogram, etc.

How I Addressed These Challenges

To resolve these issues, I sought help from StackOverflow, DeepSeek, ChatGPT, and other online resources. I could have implemented EER using the GitHub repository, but I decided not to proceed with that approach. Instead, I used F1 score as the evaluation metric.

Assumptions

I have not made any assumptions but have balanced the dataset by taking an equal number of bonafide and spoof examples. I also considered the system$_id(e.g., A_01, A_02, etc.)forthespoofcla$

Why I Selected This Particular Model for Implementation

I selected the dual-branch network because it outperforms other models on Multi-task Learning-based Forgery Detection, especially in LA, according to the given GitHub repository. This model not only detects synthetic speech but also focuses on common features across different types of synthetic speech using multitask learning. I used the ASVspoof 2019 dataset, which was also tested in the original paper.

How the Model Works (High-Level Technical Explanation)

A. Speech Preprocessing:

- **Input:** Raw audio.
- **Feature Extraction:**
  - Branch 1: Computes LFCC (Linear Frequency Cepstral Coefficients) to detect vocoder glitches.
  - Branch 2: Computes CQT (Constant-Q Transform) log power spectrum to detect pitch/harmonic anomalies.

B. Feature Extraction Module (FEM):

- **Architecture:** Modified ResNet18 with CBAM (Convolutional Block Attention Module).
- **CBAM:** Dynamically highlights important channels and spatial regions in the feature maps.
- **Channel Attention:** Identifies which frequency bands are most discriminative.
- **Spatial Attention:** Identifies where artifacts occur in time-frequency space.
- **Output:** High-level features from each branch.

C. Forgery Classification Module (FCM):

- **Task:** Binary classification (real vs. synthetic speech).
- **Process:** Combines features from both branches and uses a classifier (e.g., fully connected layers) to make the final decision.

D. Forgery Type Classification Module (FTCM):

- **Task:** Multi-class classification (identifying the spoofing method, e.g., TTS, VC).
- **Adversarial Training via GRL (Gradient Reversal Layer):**
  - GRL flips gradients during backpropagation for the FTCM.
  - Purpose: Forces the FEM to learn generic spoofing features (not specific to any attack type).

Observed Strengths and Weaknesses

- **Strengths:** It works well for directly injected synthetic speech.
- **Weaknesses:** The model has not been evaluated on recorded speech, which might be a limitation.

Suggestions

I suggest using MFCC (Mel Frequency Cepstral Coefficients) instead of LFCC, combined with CQT, in the dual-branch network for synthetic speech detection. MFCC captures the general spectral shape and low/mid-frequency characteristics of speech, while CQT focuses on capturing harmonic details and artifacts. Additionally, including more diverse datasets (in different languages) would improve the model's robustness.

Performance Results on Chosen Dataset

I used a balanced dataset for training and validation. Therefore, I evaluated the model's performance using F1 score and accuracy as metrics. The performance results on the validation data are as follows:

- **F1 Score of Validation Data:** 0.99
- **Accuracy of Validation Data:** 0.99

Deployment in Production Environment

To deploy the Dual-Branch Network in a production environment, the following approach is considered:

- **Model Optimization:** Quantization and pruning to reduce model size and improve inference time. Convert the model to ONNX or TensorFlow for deployment.
- **Edge vs Cloud Deployment:** For real-time detection, the model can be deployed on edge devices with hardware accelerators or in the cloud for scalability.
- **API Setup:** Develop a RESTful API for real-time audio classification, processing small chunks of audio continuously.
- **Monitoring and Maintenance:** Set up logging for performance metrics like EER and t-DCF, and monitor model drift to trigger retraining when necessary.
- **Security:** Encrypt data and ensure privacy policies for sensitive audio data.
- **Continuous Learning:** Periodically update the model with new data to stay resilient against emerging spoofing techniques.