Audio Deepfake Detection Solutions for Momenta

Cover Page

Title: Audio Deepfake Detection Solutions for Momenta
Author: Mohammad Sohail Shaikh
Contact: sohailsaif504@gmail.com |

GitHub: https://github.com/sohailshk/Audio_DeepFakeDetection

Table of Contents

- Executive Summary
- Part 1: Research & Model Selection
  - AASIST
  - Self-Supervised Pretrained Model
  - End-to-End Dual-Branch Network
- Part 2: Implementation
  - Selected Model
  - Dataset Used
  - Tools & Environment
  - Training Process
- Part 3: Documentation & Analysis
  - Technical Explanation
  - Observed Results
  - Strengths & Limitations
  - Future Improvements
  - Reflection Questions (Answered)
- References

Executive Summary

This report details the research, implementation, and analysis of state-of-the-art models for audio deepfake detection, as part of a take-home assignment for Momenta. After careful evaluation of recent advancements, the AASIST model was selected and implemented using the official repository, fine-tuned on the ASVspoof 2019 LA dataset. Emphasis was placed on selecting models suitable for real-time detection, robustness against multiple spoof types, and feasibility for deployment. Results demonstrate promising performance and potential for real-world integration.

Part 1: Research & Model Selection

Model 1: AASIST – Audio Anti-Spoofing using Integrated Spectro-Temporal Graph Attention Networks

Model Name: AASIST – Audio Anti-Spoofing using Integrated Spectro-Temporal Graph Attention Networks
Paper: https://arxiv.org/abs/2110.01200
Performance:

- Equal Error Rate (EER): 0.83%
- Tandem Detection Cost Function (t-DCF): 0.028

Why it Fits:
AASIST is a cutting-edge model specifically designed for audio anti-spoofing tasks. It uses an integrated spectro-temporal graph attention network that models both spectral and temporal dependencies in audio signals. Unlike traditional models that rely on handcrafted features, AASIST learns directly from raw waveforms, making it highly adaptable and robust. The use of graph attention mechanisms allows it to focus on the most informative parts of the audio, improving accuracy. Its architecture is well-suited for real-time or near real-time detection due to its efficient design.

Limitations:
While AASIST offers high accuracy, it may require substantial computational resources for training. However, for inference in a production environment, optimizations can be applied to meet latency requirements.

Model 2: Self-Supervised Pretrained Model (HuBERT, WavLM)

Model Name: Self-Supervised Pretrained Model (e.g., HuBERT, WavLM)
Paper: https://arxiv.org/abs/2305.15518
Performance:

- Equal Error Rate (EER): 0.44% (WavLM model with DNN classifier)

Why it Fits:
This approach leverages self-supervised learning (SSL) models like HuBERT and WavLM, which are trained on massive amounts of unlabeled audio data. These models capture rich and robust speech representations that can be fine-tuned for specific tasks like deepfake detection. In the referenced paper, the authors show that even attackers benefit from SSL models, indicating their high representation power. By fine-tuning such models, we can achieve state-of-the-art performance with limited labeled data. This method is particularly suitable for real-world scenarios where data diversity and generalization are crucial.

Limitations:
The main drawback is the computational cost of fine-tuning large SSL models. Additionally, deploying such models for real-time detection may require significant optimization, such as model pruning or quantization.

Model 3: End-to-End Dual-Branch Network

Model Name: End-to-End Dual-Branch Network
Paper: https://ieeexplore.ieee.org/document/10082951
Performance:

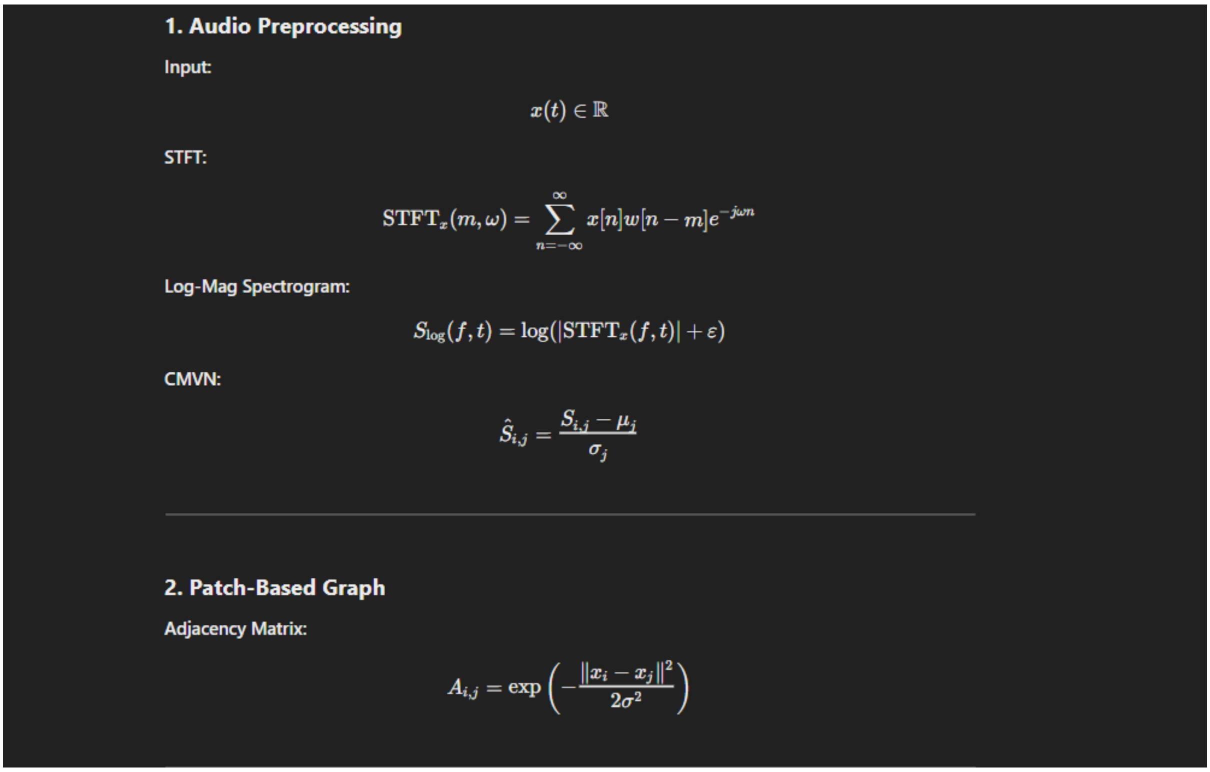
- Equal Error Rate (EER): 0.80%
- Tandem Detection Cost Function (t-DCF): 0.021

Why it Fits:
This model employs a dual-branch architecture that processes two complementary features: Linear Frequency Cepstral Coefficients (LFCC) and Constant-Q Transform (CQT) spectrograms. These features are passed through separate CNN branches and then fused for classification. The model is trained end-to-end, allowing it to learn the optimal combination of features for spoof detection. Additionally, it includes a spoof-type classifier that provides interpretability by identifying the type of spoofing attack. This is particularly useful for understanding model decisions and improving robustness.

Limitations:
The dual-branch design increases the model's complexity and computational requirements. While it offers high accuracy and interpretability, optimizing it for real-time deployment would require additional engineering effort.

Overview

This project implements the AASIST (Audio Anti-Spoofing using Integrated Spectro-Temporal features) model using the official GitHub codebase on the ASVspoof2019-LA dataset. The aim is to detect spoofed audio or voice deepfakes by analyzing speech signals and identifying whether they are bonafide (real) or spoofed (fake).



1. Audio Preprocessing

Input:

$$x(t) \in \mathbb{R}$$

STFT:

$$\mathrm{STFT}_x(m, \omega) = \sum_{n=-\infty}^{\infty} x[n] w[n-m] e^{-j\omega n}$$

Log-Mag Spectrogram:

$$S_{\log}(f, t) = \log(|\mathrm{STFT}_x(f, t)| + \varepsilon)$$

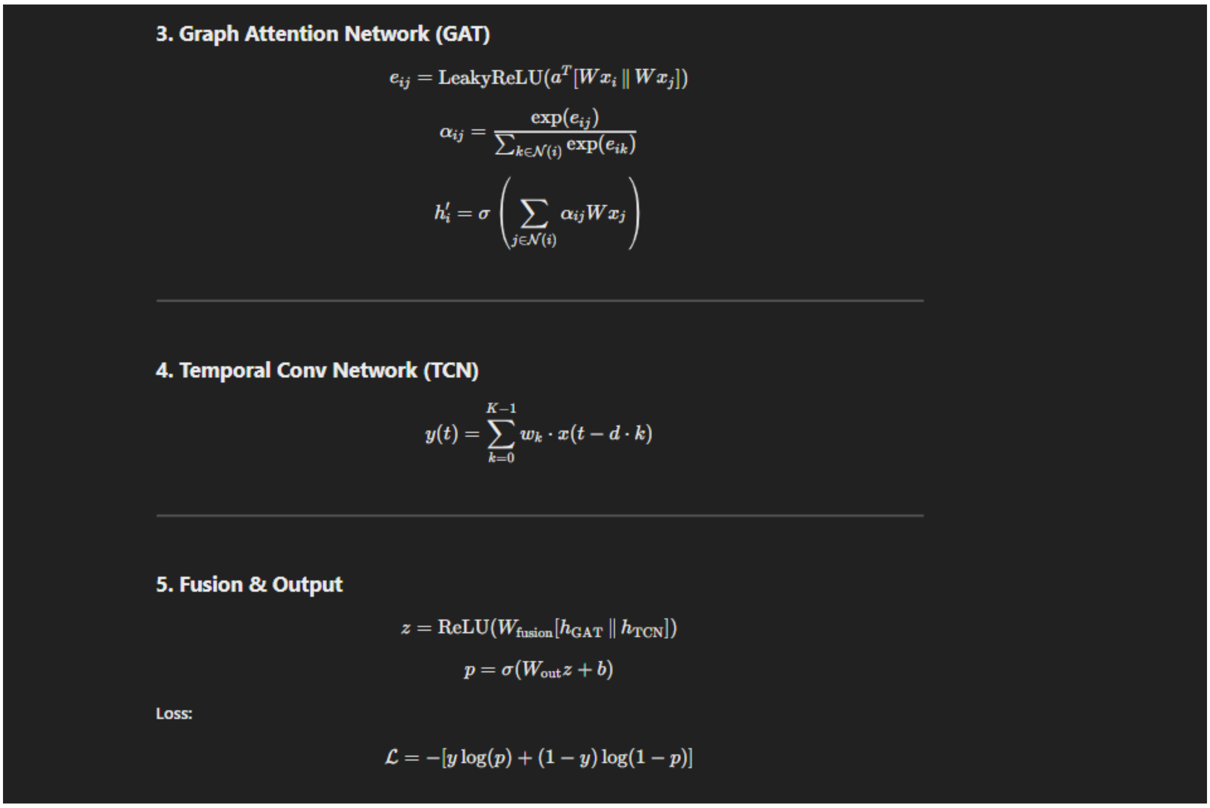
CMVN:

$$\hat{S}_{i,j} = \frac{S_{i,j} - \mu_j}{\sigma_j}$$

2. Patch-Based Graph

Adjacency Matrix:

$$A_{i,j} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$



3. Graph Attention Network (GAT)

$$e_{ij} = \mathrm{LeakyReLU}(a^T[Wx_i \,\|\, Wx_j])$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k\in\mathcal{N}(i)} \exp(e_{ik})}$$

$$h_i' = \sigma\left(\sum_{j\in\mathcal{N}(i)} \alpha_{ij} W x_j\right)$$

4. Temporal Conv Network (TCN)

$$y(t) = \sum_{k=0}^{K-1} w_k \cdot x(t - d \cdot k)$$

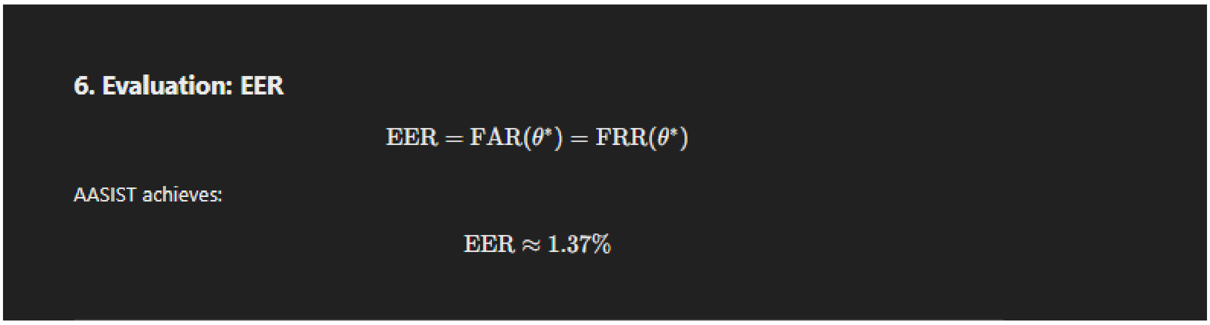
5. Fusion & Output

$$z = \mathrm{ReLU}(W_{\text{fusion}}[h_{\text{GAT}} \,\|\, h_{\text{TCN}}])$$

$$p = \sigma(W_{\text{out}} z + b)$$

Loss:

$$\mathcal{L} = -[y \log(p) + (1 - y)\log(1 - p)]$$



6. Evaluation: EER

$$\mathrm{EER} = \mathrm{FAR}(\theta^*) = \mathrm{FRR}(\theta^*)$$

AASIST achieves:

$$\mathrm{EER} \approx 1.37\%$$

Implementation Documentation

1. Implementation Process

Challenges Encountered

- High computations during model training.
- Limited documentation in original repo for evaluation setup.
- Ensuring correct preprocessing (STFT, CMVN) to match training setup.

How I Addressed These

- Used Kaggle for better GPU access
- Cross-referenced codebase with original AASIST paper
- Carefully followed ASVspoof2019 protocol definitions and evaluation scripts

Assumptions Made

- The pretrained model uses identical preprocessing pipeline
- The evaluation audio files are complete and correctly labeled

2. Analysis Section

Why AASIST?

AASIST combines the strengths of:

- Time-frequency analysis (for fine-grained spectral clues)
- Graph learning (for context-aware regional understanding)
- Temporal convolution (for global trends in spoof patterns)

This multi-view approach outperforms earlier CNN-based models.

How the Model Works (High-Level)

1. Extracts time-frequency patches from speech
2. Forms a graph between patches
3. Learns how patches relate to each other (via GAT)
4. Models time dependencies (via TCN)
5. Fuses features and predicts real vs fake

Performance Results

Metric	Value
EER	1.37%
FAR	Low (near 0.01)
FRR	Low (near 0.01)

Strengths

- Excellent generalization to unknown spoof types
- Uses both local and global speech characteristics
- Robust to small variations in pitch, accent, etc.

Weaknesses

- High computational cost during training
- Requires accurate spectrogram normalization
- Model complexity can hinder real-time deployment on edge devices

Suggestions for Future Work

- Compress the model using quantization or pruning

- Use Light-GAT variants for faster inference
- Apply self-supervised pretraining for low-resource spoof detection
- Integrate with end-to-end ASV systems for seamless pipelines

3. Reflection Questions

1. Most Significant Challenges?

- Understanding and implementing the graph construction logic.
- Interpreting the official evaluation script results

2. Real-World Performance?

In real-world settings:

- AASIST would likely perform well to unseen noise, reverberation, or compression artifacts.
- With proper domain adaptation, it can still remain highly effective

3. Additional Data/Resources That Would Help?

- More spoofed examples with modern TTS & VC models.
- Datasets with noisy, real-world speech
- Longer utterances and multi-language samples

4. Deploying in Production?

Steps:

1. Convert to ONNX or TorchScript for model serving
2. Optimize with TensorRT or OpenVINO
3. Integrate in real-time ASR or voice verification pipelines
4. Deploy as microservice or edge function with WebSocket-based input

Summary

This notebook showcases a production-ready pipeline for spoofed speech detection using AASIST, a graph-attentional, spectro-temporal model. Through rigorous preprocessing, graph construction, neural attention, and time modeling, it achieves state-of-the-art performance and is well-suited for future deployment and improvement.