AASIST Audio Spoof Detection Test Report

Take Home Assessment - QA Engineer (Core AI)

Introduction

The purpose of this assessment is to evaluate the performance of the **AASIST** model in detecting synthetic (spoofed) audio generated by modern Text-to-Speech (TTS) services.

For this assessment, I utilized three TTS platforms and random data from the ASVSpoof19 Evaluation set:

PlayHT, **ElevenLabs**, **Amazon Polly**

The AASIST model is evaluated on audio samples generated by these platforms, converted to FLAC format, and processed for spoof detection.

Methodology

1. Dataset Loading and Conversion

Initially, the project aimed to handle multiple audio formats, but for consistency and to ensure compatibility with the AASIST model, all audio files generated from TTS platforms were converted into **FLAC** format. This process involved a custom pre-processing script that converted MP3 and WAV files to FLAC. The audio files were also downsampled to match the input size required by the AASIST model, ensuring each file had a uniform length of 64600 samples (~4 seconds of audio).

---

2. Synthetic Audio Generation

Synthetic audio samples were generated using three prominent Text-to-Speech (TTS) platforms: PlayHT, ElevenLabs, and Amazon Polly. Additionally, I included samples from the ASVspoof19 eval dataset to compare performance on a known benchmark dataset.

Step 1: Selection of Synthetic Audio Generation Platforms

1. **PlayHT**:
    - PlayHT 2.0 offers state-of-the-art generative voice models capable of producing incredibly realistic human-like voices, which are crucial for generating authentic audio samples. These voices are trained on over a million hours of conversational speech,

ensuring natural intonations, emotions, and accents, perfect for datasets that aim to mimic real-world scenarios
    - PlayHT allows you to control the emotional tone and style of the generated speech, adding depth to your dataset by producing different speaking styles.
2. **ElevenLabs:**
    - ElevenLabs has been gaining popularity for its cutting-edge AI voice synthesis technology, and one of the recent headlines was its collaboration with DeepReel to create an AI version of Fernando Alonso.
    - **Widespread Use in Media and Content Creation**:Its ability to replicate emotions and speaking styles helps in producing highly interactive and contextually rich content
3. **Amazon Polly:**
    - **Multiple Speaker Styles**: Amazon Polly provides unique speaker styles such as **newscaster** and **whisper**, which are available in its standard engine (free version).The newscaster voice is designed to sound authoritative and polished, mimicking the style of broadcast journalism, while the whisper voice offers a more intimate and soft-spoken delivery.
    - Another reason for using Amazon Polly is that I already have a collection of samples generated through Polly's various engines and styles for my previous work on an **audio deepfake detection dataset**
        i. Specifically, I found it interesting to test the **newscaster** and **whisper** voices, which are generated using Polly's standard engine rather than the more advanced neural engine. These samples are not as natural-sounding as neural voices, offering a valuable opportunity to assess whether the model is successful in detecting these
    - Polly's TTS engine is widely used across various industries for applications like customer service bots, e-learning platforms, and accessibility features which is another reason to test the robustness of these models.

Step 2: Creation of the Sample Set

The final sample set was distributed as follows:

- **PlayHT**: 5 samples
- **ElevenLabs**: 5 samples
- **Amazon Polly**: 5 samples
- **ASVspoof19 eval dataset**: 5 samples

Each Platform provided different settings

- **PlayHT**: Speed, Stability, Similarity, Intensity and styles
- **ElevenLabs**:Multiple engines, voice conversion, Stability, Similarity, Style exaggeration
- **Amazon Polly**: Newscaster and whisper styles, Neural and Standard Engines

Step 3: Prompts

For consistency across platforms, The following prompts were used to generate the synthetic audio samples (2-5 are a few of the common sentences in the VCTK dataset):

1. "Let's see how realistic this sounds."
2. "Some have accepted it as a miracle without physical explanation."
3. "Ask her to bring these things with her from the store."
4. "People look, but no one ever finds it."
5. "There is, according to legend, a boiling pot of gold at one end."

Metadata for Samples

The metadata for each sample, including file names, prompts, settings, and the spoofing scores was recorded.

---

3. Custom Inference Script

The main challenge in the implementation was adapting the AASIST model's training script to function as an **inference-only** script. The original script included unnecessary training-related processes that were not relevant to this task. Here are the key modifications made:

- **Loading Pre-Trained Model**: The pre-trained AASIST model was loaded directly for inference.
- **Loading Custom Audio Dataset**: Instead of using the training and development loaders designed for the ASVspoof2019 dataset, the script was modified to load custom audio files from the test directory, This involved tweaking the dataset loader (from data_utils.py) to handle the new audio files.
- **Downsampling and Padding**: The audio files from TTS systems had varying sampling rates. To ensure compatibility with the AASIST model, the audio files were downsampled to the required input format (16 kHz, 64600 samples) and padded or cropped accordingly. This ensured that the audio data fed into the model was of the correct length, matching the expected input size.

Step-by-Step Inference Process:

1. **Loading the Custom Audio Files**:
    - The dataset loading process was adapted to load the custom audio files.
2. **Preprocessing and Downsampling**:
    - All input files were downsampled to **16 kHz** to match the model's requirements. Additionally, each audio file was padded or cropped to ensure that the input length was exactly 64600 samples (~4 seconds), which is the expected length by the AASIST model.
3. **Inference**:

- Once the custom dataset was loaded and preprocessed, the pre-trained AASIST model was used to infer whether each audio sample was spoofed or genuine based on the second logit output from the model.

---

4. Evaluation and Results

The AASIST model was evaluated using a set of 20 audio samples generated across three different Text-to-Speech (TTS) platforms, alongside samples from the ASVspoof19 eval dataset. Each sample was converted to FLAC format, downsampled, and processed through the model to determine whether the model classified the sample as genuine or spoofed.

Evaluation Metrics

Logits and Spoofing Scores

The AASIST model produces two logits for each audio file:

1. **Logit 1**: Corresponds to the **bonafide** class (genuine audio).
2. **Logit 2**: Corresponds to the **spoofed** class (fake audio).

In the code, we capture **Logit 2** (`batch_out[:, 1]`) as the spoofing score. This score reflects the model's confidence that the input audio is spoofed:

- **Higher positive values** indicate a higher likelihood of the audio being spoofed.
- **Negative or low values** indicate that the model considers the audio bonafide (genuine).

Results Breakdown

Sample	Score	Classification
'Sample_01'	3.9994867	Spoofed
'Sample_02'	3.5255578	Spoofed
'Sample_03'	-4.2308464	Misclassified(Genuine)
'Sample_04'	-4.4883237	Misclassified(Genuine)
'Sample_05'	-4.2774663	Misclassified(Genuine)
'Sample_06'	0.47145495	Spoofed
'Sample_07'	1.1184292	Spoofed
'Sample_08'	-1.6315277	Misclassified(Genuine)
'Sample_09'	4.362828	Spoofed
'Sample_10'	4.607647	Spoofed

'Sample_11'	-4.2932224	Misclassified(Genuine)
'Sample_12'	-1.115079	Misclassified(Genuine)
'Sample_13'	-3.1623971	Misclassified(Genuine)
'Sample_14'	-3.8039267	Misclassified(Genuine)
'Sample_15'	-1.1865851	Misclassified(Genuine)
'LA_E_4311043'	3.8464766	Spoofed
'LA_E_8411060'	5.066361	Spoofed
'LA_E_8411845'	-2.7512786	Misclassified(Genuine)
'LA_E_9296375'	6.3667016	Spoofed
'LA_E_9297933'	5.616393	Spoofed

The following is a summary of the results:

1. PlayHT Results

- **Spoofed Samples Detected**: 2/5
- **Genuine Samples Misclassified**: 3/5

Out of 5 PlayHT samples, only two were correctly classified as spoofed, while three were misclassified as genuine, despite being generated by state-of-the-art TTS technology. This indicates that the AASIST model struggles with PlayHT-generated synthetic audio.

2. ElevenLabs Results

- **Spoofed Samples Detected**: 3/5
- **Genuine Samples Misclassified**: 2/5

Despite ElevenLabs' advanced voice synthesis, three out of five samples were correctly detected as spoofed by the AASIST model, showing that even high-quality synthetic voices can still exhibit detectable artifacts.

3. Amazon Polly Results

- **Spoofed Samples Detected**: 0/5
- **Genuine Samples Misclassified**: 5/5

All five samples generated from Amazon Polly were misclassified as genuine. This includes samples generated with the newscaster and whisper styles, highlighting a significant weakness in the AASIST model when it comes to these types of synthetic speech.

4. ASVspoof19 Results

1. **Spoofed Samples Detected**: 4/5
2. **Genuine Samples Misclassified**: 1/5

The model performed well on the ASVspoof19 samples which aligns with its intended design for detecting traditional forms of spoofed audio.

Observations

1. **Performance on Modern TTS**: The model struggled significantly with modern TTS systems, particularly Amazon Polly. Most of the samples from PlayHT and ElevenLabs were misclassified as genuine, even though they were synthetic.
    a. Out of the **five correctly classified spoofed samples** from the three TTS platforms, two were voice conversion samples, while the other two were tampered with using modifiable settings like stability, style exaggeration, intensity, etc. This suggests that altering these parameters or using voice conversion makes it easier for the AASIST model to detect synthetic elements.
2. **ASVspoof19 Accuracy**: The model performed better on the ASVspoof19 dataset, which was expected given that the AASIST model was trained on similar data.
3. **Spoofing Detection Limitations**: The consistent misclassification of newer TTS-generated audio indicates that the model has limitations in detecting advanced synthetic voices. These voices, which mimic human speech patterns more effectively, challenge traditional spoof detection models.
4. To improve spoof detection on modern TTS platforms, the AASIST model would likely need additional training on synthetic speech generated by platforms such as PlayHT, ElevenLabs, and Amazon Polly.

Model Limitations and Areas of Underperformance

While the AASIST model performed effectively across most synthetic speech detection tasks, there were certain advanced spoofing techniques, found in the ASVspoof19 dataset, where the model showed notable limitations.

The following are specific attacks where the model underperformed:

A08 (TTS Neural Waveform): This attack uses **neural waveform generation**, which produces synthetic speech that closely mimics the natural characteristics of human speech.

A16 (TTS Waveform Concatenation): This attack involves **waveform concatenation**, where synthetic speech is created by concatenating segments of recorded speech.

A17 (VC Waveform Filtering): This voice conversion (VC) attack utilizes **waveform filtering**, which subtly alters voice characteristics while retaining the original speaker's identity.

A18 (VC Vocoder): This attack leverages **vocoder-based voice conversion**, where the speech signal is transformed using a vocoder and then reconstructed, often to change speaker characteristics.

Leveraging Research Insights for Model Failure

This shows that the AASIST model could be manipulated to fail under specific conditions. Some of the findings from research into attacks such as A08 and A17 could have been leveraged to introduce targeted failures in the system but modern TTS systems get the job done.

Resources and References

https://elevenlabs.io/app/speech-synthesis/text-to-speech

https://play.ht/text-to-speech/

https://github.com/clovaai/aasist

https://github.com/Dharva12/Audio-Deepfake-Detection-Dataset/blob/main/VCTK-common-sentences.txt

https://www.tomsguide.com/ai/elevenlabs-creates-an-ai-version-of-fernando-alonso-and-you-can-talk-to-him

https://pytorch.org/audio/stable/index.html

https://datashare.ed.ac.uk/handle/10283/3336