Audio Processing And Indexing, 2023
PROJECT REPORT

Silicon vs Vocal cords: Classifying fake audios

May 23, 2025

Christos Tsirogiannis(S3226352),
Parthipan Ramakrishnan(S3447014),
Priya Prabhakar(S3292983),
Swati Soni(S3417522),
Leiden Institute of Advanced Computer Science
Leiden University

1 Abstract

In this study, we explore advanced audio classification techniques to differentiate between human and AI-generated audio, leveraging the ASVspoof2019 LA dataset. We investigate two primary approaches: utilizing raw audio input and employing extracted audio features. Employing Deep Neural Network (DNN) methods, including a Time-Domain Synthetic Speech Detection (TSSD) model and a novel CNN-LSTM hybrid model, we aim to achieve high accuracy in classifying audio as either 'bonafide' (genuine human speech) or 'spoof' (AI-generated). The CNN-LSTM model, integrating convolutional layers with Long Short-Term Memory units, emerges as the best-performing model, exhibiting remarkable accuracy. Additionally, we developed a Streamlit-based interactive web application, allowing real-time classification of audio files and showcasing the practical application of our research.

Project Webapp URL : `https://siliconvsvocalchords.streamlit.app/`

2 Introduction

The rapid advancement of artificial intelligence has brought groundbreaking developments across various fields, yet it also poses significant challenges, particularly in the realm of audio deepfakes. A notable instance of these challenges is a fraudulent transfer of $35 million, underscoring the urgent need to address the risks associated with AI-generated audio, including audio spoofing and voice cloning scams. Our study engages with these challenges, focusing on differentiating between human-spoken and AI-generated audio using the ASVspoof2019 LA dataset, specifically designed for developing countermeasures against audio spoofing.

The global impact of AI voice cloning scams has been profound. A 2023 global survey revealed that one in ten individuals reported being targeted, leading to significant financial losses and manipulative practices ranging from political discourse to fraudulent multimedia content. This issue of deepfakes extends beyond audio, affecting video, image, and text artifacts, all manipulated by artificial intelligence. The academic community's response to these challenges is reflected in the surge of deepfake-related publications, with a significant increase in articles from 60 in 2018 to 1,323 by the end of 2020.

In addressing these concerns, our project employs two distinct methodologies: one utilizing raw audio data, following the method proposed by Hua et al.[1], and another that leverages a feature extraction strategy. We explore the effectiveness of these approaches through the development of various deep neural network (DNN) models, particularly emphasizing a CNN-LSTM hybrid model. This approach not only adds to the academic understanding of audio authentication but also presents practical solutions, including the development of an interactive web application.

3 Previous Works

Significant studies in the field, such as Deep Voice 3 by Ping et al.[2] in 2018, the WaveNet model by Oord et al.[3] in 2016, and the ASVspoof 2017 dataset by Wu et al.[4], underscore the importance of this research. These studies represent major advancements in synthesizing quality speech and developing robust countermeasures against deepfakes. Our comprehensive approach, utilizing the ASVspoof2019 LA dataset for training, highlights the complexities and potential solutions in the battle against audio deepfakes, aiming to develop and evaluate effective countermeasures against spoofed or artificially generated audios in various applications.

3.1 Categorization of Deep Fakes

Attempts to address audio deepfake issues involve various categories:

- **Replay-based Detection:** Uses techniques like far-field and cut-and-paste detection, leveraging deep convolutional neural networks.
  - **Pros:** Effective against replay attacks, uses deep convolutional neural networks.
  - **Cons:** May have limitations in certain scenarios, requires careful handling of data.
- **Synthetic-based Generation:** Involves the artificial production of human speech using models like WaveNet and subsequent systems.
  - **Pros:** Can produce highly realistic artificial voices.
  - **Cons:** Quality depends on the voice corpus, is expensive to create, and struggles with punctuation and ambiguity.
- **Imitation-based Generation:** Utilizes algorithms, including Generative Adversarial Networks (GAN), for transforming original speech to mimic another speaker.
  - **Pros:** Mimics target voice without altering linguistic information, uses neural networks for flexibility and high-quality results.
  - **Cons:** Confused with synthetic-based method, may require considerable preprocessing, and accent considerations are crucial.
- **Detection methods:** Focus on low-level and high-level aspects, with machine learning and deep learning models developed to discern between real and fake audio.
  - **Pros:** Can leverage both low-level and high-level features for detection.
  - **Cons:** ML methods may lack scalability, deep learning requires specific transformations, and excessive preprocessing can lead to high computational costs.

4 Dataset

This study utilized the **ASVspoof2019 LA** (Logical Access) dataset for training purposes. The dataset is structured to assist in the development and evaluation of countermeasures against spoofed or artificially generated audios, specifically targeting voice biometric systems, cybercrimes and other illegal activities.

- **Data Source:** The dataset was sourced from **Automatic Speaker Verification: Spoofing And Countermeasures Challenge 2019** [5]. The ASVspoof 2019 database, structured into two distinct sections for evaluating logical access (LA) and physical access (PA) scenarios, served as the foundational dataset for this study. Our analysis exclusively utilized the LA partition. This partition, along with its PA counterpart, originates from the VCTK base corpus, encompassing speech recordings from a total of 107 speakers, distributed between 46 males and 61 females. Both the LA and PA segments of the database are further segmented into three distinct subsets: training, development, and evaluation. These subsets consist of speech samples from 20 speakers (8 male, 12 female) for training, 10 speakers (4 male, 6 female) for development, and 48 speakers (21 male, 27 female) for evaluation, respectively.

- **Dataset Composition:** The dataset comprises audio files, each categorically labelled as "bonafide" to denote genuine human speech or "spoof" to indicate AI-generated audio. Accompanying these audio files is a label file, which provides essential details about each audio file, such as its filename and the corresponding authenticity label. A custom function was developed to parse each line of this label file, effectively segregating it into distinct components and associating each audio file with its relevant label. Labels are binary encoded: 1 for "bonafide" (genuine human speech) and 0 for "spoof" (AI-generated audio).

- **Preprocessing and Setup:** Audio preprocessing is a critical stage in the pipeline of any machine learning project dealing with audio data. Its significance in enhancing the performance of audio classification models cannot be overstated. The audio data was preprocessed to standardize the sample rate at 16,000 Hz and ensure a uniform duration of 5 seconds per audio clip. The dataset was divided into training and validation sets using an 80-20 split, ensuring a balanced representation of both classes.

5 Methodology

In this study, we have implemented two distinct approaches. The first approach follows the method proposed by Hua et al.[1], where raw audio is used directly as input. The second approach diverges by focusing on the extraction of specific audio features from the recordings, which are then used as input for classification. This dual-method strategy allows us to comprehensively evaluate and compare the effectiveness of each technique in audio classification.

5.1 DNN method without feature extraction

We have used Time-Domain Synthetic Speech Detection (TSSD) introduced by Hua et al.[1]. It utilizes a Resnet-based neural network for speech classification. This approach takes raw audio as input, eliminating the necessity for manually crafted features. The architecture, outlined in Figure 1, comprises seven 1D convolution layers, followed by batch normalization, ReLU, and a ResNet-style block. In our implementation, we incorporated four such blocks (M=4), accompanied by a max-pool layer with a kernel size of 4. Each module consists of three 1x3 convolution layers, and their output is concatenated with the original input transformed by a 1x1 convolution. Batch normalization (BN) and ReLU are applied after each of these layers. In the implementation, the number of channels (CR) is set to 16, 32, 64 and 128 for the first, second, third and final blocks respectively. The TSSD model code is referred from https://github.com/MarkHershey/AudioDeepFakeDetection.

5.2 DNN methods with feature extraction

In this approach, rather than directly utilizing raw audio as input, we opt to extract a selected set of audio features from the audio aiming to gain deeper insights into the audio. By providing these extracted features as input, we enrich the model with additional, relevant information, which is anticipated to yield enhanced performance results.

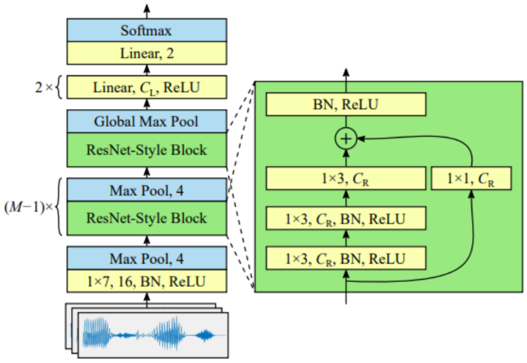

Figure 1: TSSD architecture

5.2.1 Feature Extraction

Feature extraction is a crucial step in analyzing audio data, as it involves transforming raw audio into a more comprehensible format for machine learning models. In this project, a comprehensive set of features was extracted from audio files to enable the effective differentiation between AI-generated and human audio. The extraction process was guided by several key parameters, ensuring a standardized approach across the entire dataset.

- **Key Parameters:**
  - **Sample Rate:** All audio files were standardized to a sample rate of 16,000 Hz. This rate is a balance between sufficient audio quality for feature extraction and computational efficiency.
  - **Duration:** Each audio clip was processed to have a uniform duration of 5sec. This consistency is crucial for comparable feature extraction across all samples.
  - **Number of Mel Bands:** For Mel spectrograms, 128 Mel bands were used, providing a detailed representation of the audio spectrum while maintaining computational feasibility.
  - **FFT Window Size:** The size of the Fast Fourier Transform window was set to 512. This parameter determines the frequency resolution of the analysis.
  - **Hop Length:** A hop length of 256 was chosen for the analysis. This defines the amount of overlap between successive FFT windows, impacting the time resolution.
  - **Maximum Time Steps:** The features were either padded or truncated to have a uniform size of 109 time steps, ensuring that each audio sample contributes equally to the dataset. 109 time steps were chosen based on the combination of sample rate, audio length, and windowing parameters.
- **Extracted Features:**
  - **Mel Spectrogram:** Captures the power spectrum of sound in the Mel scale, providing insights into the frequency content of the audio and also offering a representation closer to human auditory perception.

- **MFCCs (Mel-Frequency Cepstral Coefficients):** Thirteen coefficients were extracted, offering a compact representation of the audio spectrum. These coefficients succinctly represent the short-term power spectrum of a sound, typically used in voice recognition.
- **Chroma Frequencies (STFT):** Chroma features are a powerful tool for analyzing music, representing the intensity of each pitch class, and useful in distinguishing tonal content.
- **Spectral Contrast:** Measures the spectral peak, valley, and their differences across different frequency bands.
- **Tonnetz (Tonal Centroid Features):** Captures the tonal characteristics of sound, reflecting the harmonic and melodic relations, these features are particularly useful in distinguishing between different types of audio content.

The extraction process involved loading each audio file at the predefined sample rate, followed by padding or truncating the audio clips to maintain uniformity. Subsequently, each of the aforementioned features was computed and then concatenated to form a comprehensive feature set for each audio clip using the **librosa** [6] python package.

5.2.2 Models

For this approach of using extracted features as input instead of raw audio data, we designed three distinct neural network models to classify audio files as either human-voice or AI-generated. These models were trained using the aforementioned features5.2.1 extracted from audio data. The input layer for all three 3 models are same.

Input Shape: The extracted features were structured in a format suitable for the CNN models, with the shape [166, 109, 1], where 166 and 109 represent the height(combined features) and width(time steps) of the spectrogram, and 1 indicates single-channel data.

1. **Base CNN Model** :
   - **Architecture:** The baseline model was a simple CNN, starting with two convolutional layers (32 and 64 filters, kernel size 3x3), each followed by max pooling layers. The network then flattened the output and passed it through a dense layer with 128 units, followed by a dropout layer to prevent overfitting.
   - **Output Layer:** A softmax output layer with two units was used for binary classification (human vs. AI-generated audio).
   - **Compilation and Training:** The model was compiled with the Adam optimizer and categorical cross-entropy loss function, and trained on the dataset with a focus on achieving high accuracy.

   In terms of performance, the model achieved a 70% accuracy rate.

2. **Enhanced CNN Model**:
   - **Improved Architecture:** This model included three convolutional blocks (64, 128, 256 filters), each with batch normalization, max pooling, and dropout for regularization.
   - **Compilation:** Compiled with the same optimizer and loss function as the baseline model, emphasizing model performance and generalization.

The additional layers with the addition of batch normalization and increased dropout enhance the model's learning capability, leading to better performance in terms of accuracy (87%). The batch normalization helps in reducing the internal covariate shift. The inclusion of batch normalization after each convolutional layer helps in stabilizing the learning process and also improving the training speed.

3. **CNN-LSTM Model:**

   This model is a hybrid architecture that combines convolutional neural network (CNN) layers with Long Short-Term Memory (LSTM) units. The architecture in the LSTM model includes several important components that work together to process the data.

   - **Hybrid Architecture:** This advanced model combined CNN layers for spatial feature extraction with LSTM layers for temporal feature processing. It started with two convolutional layers (32 and 64 filters), each followed by max pooling.
   - **LSTM Integration:** The output was then reshaped and fed into two LSTM layers (64 units each), capturing temporal dependencies in the audio sequence.
   - **Compilation and Training:** Compiled with Adam optimizer and categorical cross-entropy loss, this model was trained to leverage both spatial and temporal characteristics of the audio data, making it particularly effective for the classification task.

   The addition of LSTM layers after the convolutional base is a key differentiator from the previous models and is particularly suited to the task of audio processing. Convolutional layers alone might miss the sequence information. However, this hybrid model combines CNN layers for spatial feature extraction with LSTM layers for temporal analysis. This hybrid approach allows the model to understand both the spectral characteristics of the audio (extracted by CNNs) and the time-related changes in these characteristics (captured by LSTMs). This can lead to more accurate classification results, as reflected in the reported 95% accuracy of this model.

Parameters	**Base CNN**	**Enhanced CNN**	**CNN-LSTM**
Input Size	169*109*1	169*109*1	169*109*1
Convolutional Layers	2	3	2
MaxPooling2d	2	3	2
Activation Function	Relu, Softmax	Relu, Softmax	Relu, Softmax
Type of Layers	Fully Connected	Fully Connected	Fully Connected
Batch Normalization	No	Yes	No
LSTM Network	-	-	2
Padding	No	Yes	Yes
Dropout Layers	1	4	1
Dense Layers	2	2	2

Table 1: Model architecture comparison of the different models we built for this project.

6 Results and Analysis

As discussed in the methodology section, the performance of the CNN-LSTM model is the best among all the models we tried with an accuracy of 99%.

Features Extracted	**Model**	**Accuracy**
No	TSSD	93%
Yes	Base CNN	70%
Yes	Enhanced CNN	87%
Yes	**CNN-LSTM**	**99%**

Figure 2: Results of most significant models in our experimentation

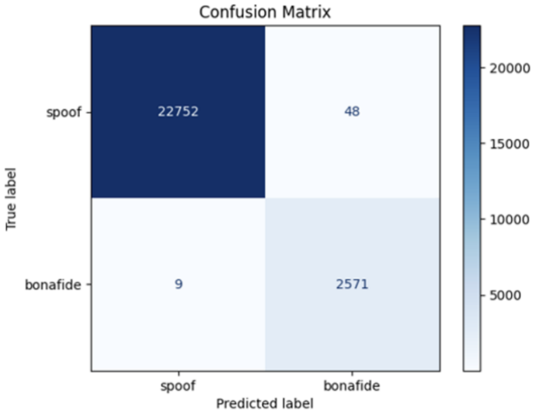

Figure 3: Confusion Matrix for the Best Performing Model(CNN-LSTM).

Figure 3 depicts a confusion matrix that represents the prediction of the CNN-LSTM model. This matrix can give us a lot of insights into how the model is performing in its precision, recall and overall accuracy.

The results shown in fig.3 illustrate a high number of true positives and true negatives, alongside a low count of false positives and false negatives. This pattern suggests that the model exhibits high accuracy, precision, and recall, underscoring its effectiveness in accurately classifying 'spoof' and 'bonafide' instances within the test set. This observation is further corroborated by the model's impressive accuracy rate of 99%, as shown in the tab.2, affirming its exceptional suitability for this audio classification task.

The reason why the CNN-LSTM is the best-performing model is because of its dual capability to extract spatial features and to model the temporal dependencies inherent in audio sequences, which are required for accurately classifying audio and detecting synthetic audios. The sequential processing of LSTM is particularly well-suited for audio and other time-series data, as it can maintain information over longer time periods, capturing the context that might be lost in traditional CNNs.

7 Interactive Platform for Audio Classification

In addition to developing the classification models, this project features an interactive web application designed to classify audio files as either human-voice or AI-generated. The application, developed using **Streamlit** package, serves as a user-friendly interface, allowing users to upload audio files for real-time classification. Project Webapp URL : `https://siliconvsvocalchords.streamlit.app/`

- **Audio File Upload and Processing:-**

  - **Upload Functionality:** The application allows users to upload audio files in various formats (FLAC, OGG, mp3). Once uploaded, these files are processed using the same preprocessing steps as the training dataset to ensure consistency.
  - **Real-Time Classification:** After preprocessing, the audio file is fed into the best-performing classification model (e.g., the LSTM model). The model then analyzes the audio and classifies it as either human or AI-generated. On average, the model takes less than 480±10 ms to classify the audio.

- **Displaying Results:-**
  - **Classification Output:** The result of the classification is displayed on the web application interface, providing users with immediate feedback.
  - **Feedback Mechanism:** The application includes an option for users to provide feedback on the classification results. This feature is essential for further refining and improving the model's accuracy.

- **Integration with Machine Learning Model:-**
  - **Backend Integration:** The application is seamlessly integrated with our best ML model. This integration ensures that the audio classification is performed efficiently and accurately.
  - **Model Updates:** The web application is designed to be easily updated with newer versions of the classification model, allowing for continuous improvement and adaptation to new data or techniques.

8 Conclusion/Future Work

Our study demonstrates the efficacy of using deep learning models, particularly the CNN-LSTM hybrid model, in accurately classifying audio files. With an impressive accuracy of 99%, the CNN-LSTM model stands out for its ability to capture both spatial features and temporal dependencies in audio sequences. This capability is critical in distinguishing between 'bonafide' and 'spoof' audios, making the model an invaluable tool in the fight against AI-generated audio fraud. Looking forward, there are several avenues for enhancing this work:

- **Ablation Study:** To measure the importance of the features extracted from the audio in classification.
- **Dataset Expansion/Augumentation:** Testing the models on additional datasets (like deepfakes) will help in assessing their robustness and generalizability. Also, we are planning to do some data augmentation techniques(like adding noise, pitch shift).
- **Model Architecture:** Moving forward, we aim to develop a new model that harnesses the power of ResNet, leveraging the extracted audio features. This approach is designed to significantly enhance our capability in audio classification tasks.

The interactive web application, a practical extension of our research, has the potential to serve a wide range of users, from security experts to general consumers, in identifying the authenticity of audio clips. As AI technology continues to evolve, our research provides a foundation for future innovations in audio authentication and security.

References

[1] G. Hua, A. B. J. Teoh, and H. Zhang, “Towards end-to-end synthetic speech detection,” *IEEE Signal Processing Letters*, vol. 28, pp. 1265–1269, 2021.

[2] W. Ping, K. Peng, A. Gibiansky, S. Ö. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, “Deep voice 3: 2000-speaker neural text-to-speech,” *CoRR*, vol. abs/1710.07654, 2017. [Online]. Available: http://arxiv.org/abs/1710.07654

[3] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” *CoRR*, vol. abs/1609.03499, 2016. [Online]. Available: http://arxiv.org/abs/1609.03499

[4] T. Kinnunen, N. Evans, J. Yamagishi, K. A. Lee, M. Sahidullah, M. Todisco, and H. Delgado, “Asvspoof 2017: Automatic speaker verification spoofing and countermeasures challenge evaluation plan *,” 09 2018.

[5] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee, L. Juvela, P. Alku, Y.-H. Peng, H.-T. Hwang, Y. Tsao, H.-M. Wang, S. L. Maguer, M. Becker, F. Henderson, R. Clark, Y. Zhang, Q. Wang, Y. Jia, K. Onuma, K. Mushika, T. Kaneda, Y. Jiang, L.-J. Liu, Y.-C. Wu, W.-C. Huang, T. Toda, K. Tanaka, H. Kameoka, I. Steiner, D. Matrouf, J.-F. Bonastre, A. Govender, S. Ronanki, J.-X. Zhang, and Z.-H. Ling, “Asvspoof 2019: A large-scale public database of synthesized, converted and replayed speech,” 2020.

[6] B. McFee, M. McVicar, D. Faronbi, I. Roman, M. Gover, S. Balke, S. Seyfarth, A. Malek, C. Raffel, V. Lostanlen, B. van Niekirk, D. Lee, F. Cwitkowitz, F. Zalkow, O. Nieto, D. Ellis, J. Mason, K. Lee, B. Steers, E. Halvachs, C. Thome, F. Robert-Stoter, R. Bittner, Z. Wei, A. Weiss, E. Battenberg, K. Choi, R. Yamamoto, C. Carr, A. Metsai, S. Sullivan, P. Friesch, A. Krishnakumar, S. Hidaka, S. Kowalik, F. Keller, D. Mazur, A. Chabot-Leclerc, C. Hawthorne, C. Ramaprasad, M. Keum, J. Gomez, W. Monroe, V. A. Morozov, K. Eliasi, N. D. Sergin, R. Hennequin, R. Naktinis, beantowel, T. Kim, J. P. Ãsen, J. Lim, A. Malins, D. Herenu, S. van der Struijk, L. Nickel, J. Wu, Z. Wang, T. Gates, M. Vollrath, A. Sarroff, Xiao-Ming, A. Porter, S. Kranzler, Voodoohop, M. D. Gangi, H. Jinoz, C. Guerrero, A. Mazhar, toddrme2178, Z. Baratz, A. Kostin, X. Zhuang, C. T. Lo, P. Campr, E. Semeniuc, M. Biswal, S. Moura, P. Brossier, H. Lee, and W. Pimenta, “librosa/librosa: 0.10.1,” Aug. 2023. [Online]. Available: https://doi.org/10.5281/zenodo.8252662