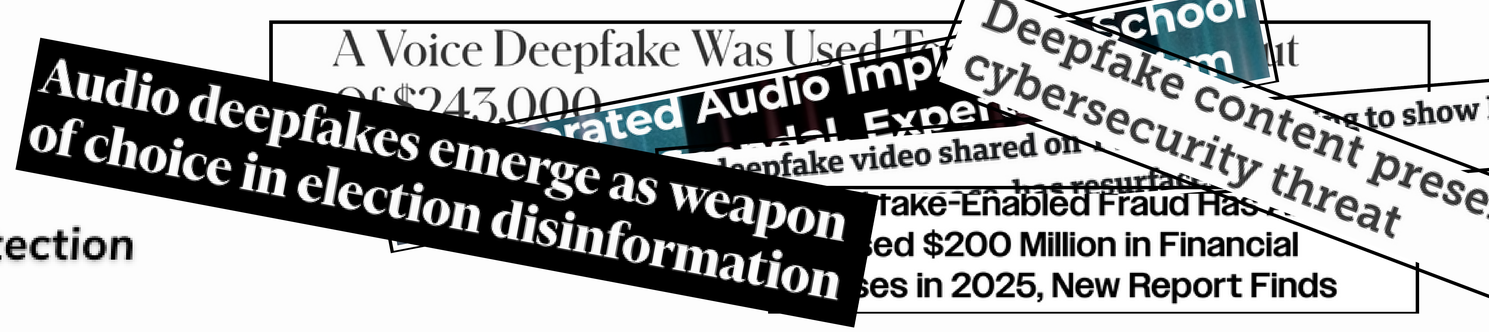
The Transformers: Samuel Liu, Andrew Kim, Ghali Maata, Sami Nourji

IS THIS REALLY YOU?

Identifying Key Features for Deepfake Audio Detection



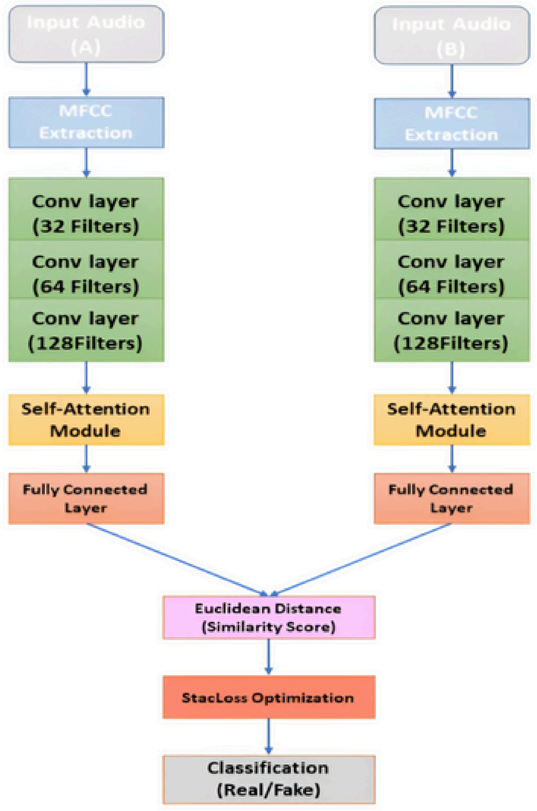
INTRODUCTION

Audio deepfakes threaten the reliability of voice-based authentication systems. While accurate detection models exist, they are computationally heavy, making them impractical for fast decision-making in urgent, real-world settings. As deepfake spoofing methods are getting better and more widespread, a need for fast, real-time deepfake detection solutions becomes pressing. To design more efficient solutions, it is critical to understand which features most effectively distinguish bona fide (real) from spoofed (deepfake) speech. Our aim is to investigate the role of individual MFCC coefficients in deepfake detection. We use a Siamese CNN architecture combined with batch-hard triplet loss to learn a meaningful embedding space. This architecture is inspired by Shaaban et al. (2025), whose Siamese CNN model achieved 98% accuracy and 2.95% Equal Error Rate (EER) and has become a state-of-the-art approach for audio deepfake detection tasks.

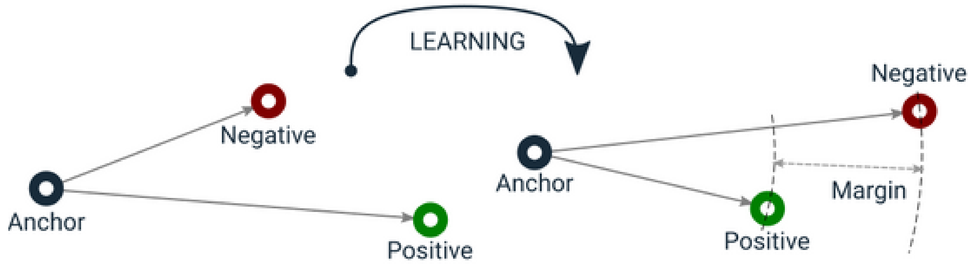
PREPROCESSING

We use the ASVspoof 2019 LA dataset, converting each audio file into a 13 × 400 MFCC tensor. Each tensor is identified with a speaker ID and a bona fide/spoof label. To make the model is more lightweight, we perform random, speaker-balanced subsampling and construct triplets before each epoch (“online”) based on label and speaker constraints. The triplets consist of an anchor, a positive (same speaker, same label), and a negative (same speaker, opposite label). During training, batch-hard mining selects the hardest positive (farthest Euclidean) and negative (closest Euclidean) examples within each batch, maximizing the training signal without storing “offline” triplet lists.

METHODOLOGY



Model Architecture: Each input is a Mel-spectrogram with 13 frequency bands and 400 time frames. Each of three convolutional blocks applies a 3 × 3 convolution, batch normalization, ReLU activation, and 2 × 2 max pooling. We preserve mid-level details by adding the output of the second block into the third through a residual connection; the resulting feature map passes through multi-head self-attention, improving focus on the most discriminative time-frequency regions. Adaptive average pooling, a linear layer, and L2-normalization produces the final embedding. Dropout is used for generalizability. Two identical branches share these weights in a Siamese structure. Optimization on batch-hard triplet loss seeks to cluster same-label audios but separate real-fake pairs.



Feature-ablation protocol: After finalizing the network design, we conducted a leave-one-feature-out (LOFO) study to identify which MFCC coefficients are most important for detection. We first trained a baseline model using all 13 MFCC features. We then created a version of the dataset with a feature removed for each corresponding coefficient k (1–13), (reducing inputs to 12 × 400) and re-trained the model under identical settings.

RESULTS

Figure 1.

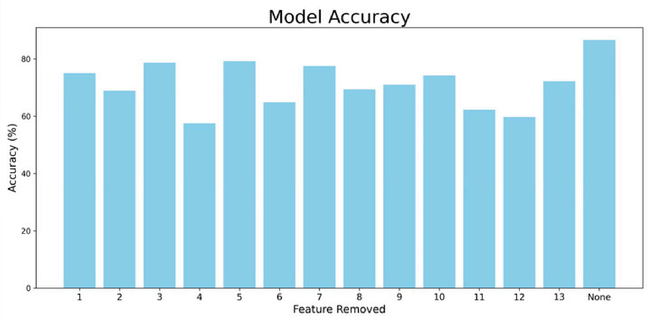


Figure 2.

Table: Relative Accuracy/EER

Model	Δ Acc. (%)	Δ EER (%)
4	-29.0	29.07
12	-26.9	26.89
11	-24.3	24.43
6	-21.7	21.83
2	-17.7	17.66
8	-17.2	17.43
9	-15.6	15.67
13	-14.4	14.59
10	-12.4	12.5
1	-11.6	11.6
7	-9.0	9.0
3	-7.9	8.05
5	-7.4	7.48

Figure 3. Relative to baseline of accuracy 86.60% and EER of 13.35%

DISCUSSION

When implementing Shaaban et al. (2025), early challenges included the implementation of batch-hard triplet loss. Though the idea of “hard mining” was clearly documented, the paper left out critical implementation details, such as whether triplets were mined across an entire epoch or within batches. The paper hinted at mining hard triplets globally over the full training set; however, reproducing this would have been computationally prohibitive, especially since our project required retraining the model 14 times for feature ablation. Since this “global triplets” computation is quadratic in the batch size, the paper reports 30-hour training times. To make the training lightweight for repeated runs, we adapted our approach: as mentioned in the preprocessing section, we computed hard triplets within each random, speaker-balanced batch rather than mining globally. This was still robust and dramatically reduced computational bottlenecks, enabling us to train multiple ablation models efficiently without sacrificing accuracy. We ensured stable training by increasing epochs fivefold (100 to 500) and tuning hyperparameters such as the number of speakers per batch accordingly. Upon reflection, our discoveries on feature rankings and dimensionality reduction could inform future real-time deepfake detection by making models more lightweight, potentially integrating with media pipelines (e.g. news platforms).