Beyond Speaker Identity: Generalizable Deepfake Detection using Audio Embeddings and Time Delay Neural Networks

Devon Campbell (dec2180)

***Abstract*—This study presents a novel approach to detecting spoofed audio in environments with limited speaker-specific data. Using Time Delay Neural Network (TDNN) architectures to extract audio embeddings, this method captures the essential spectral properties of audio samples. The model classifies audio clips as genuine or spoofed by calculating similarity metrics between test and reference audio embeddings. Notably, this approach prioritizes intrinsic audio content qualities over speaker identity, thereby minimizing the reliance on extensive speaker data and enhancing model generalizability across unknown speakers. By dynamically updating the optimal similarity threshold, the model balances sensitivity and specificity, opting to trade a level of absolute accuracy for significantly improved generalizability. This shift is particularly advantageous in security-sensitive environments where robust and flexible audio verification is crucial. The results affirm the potential of this framework to enhance security measures in diverse operational settings.**

***Index Terms*—Audio deepfake detection, time delay neural networks, audio embeddings, spoofed audio detection, audio decurity**

I. Introduction

VOICE-BASED biometrics are increasingly favored for authentication due to the vulnerability of traditional methods like keypads and text passwords. Thus, ensuring the reliability and security of voice authentication is crucial for widespread adoption [6]. As speaker verification gains traction as a security measure, there has been a corresponding surge in attempts to bypass these systems using vocal spoofing techniques, such as creating audio deepfakes (AD). Impersonation [8] and Replay [2] utilize genuine pre-recorded vocal samples of the target to mimic their natural speech via a microphone. Conversely, Speech synthesis [11] and Voice conversion [18] generate speech algorithmically from a dataset of the target's vocal samples. Given these emerging threats, it is imperative to develop robust defenses against audio deepfakes to preserve the integrity and trustworthiness of voice-based security systems.

II. State of the Art

Certain studies exploring countermeasures for spoofing attacks pursue customized audio feature extraction [14], while others prioritize developing specialized classifier models for detecting spoofed audio [22]. These studies underscore the importance of leveraging ML techniques to identify ADs. These supervised learning frameworks rely on the manual extraction of spectral acoustic features and extensive preprocessing. Nonetheless, complex and dynamic representations of the voice can be derived from time-frequency analysis, notably the frequency cepstral coefficients. The application of cepstral coefficients have shown strong results within the domain of automatic speach verification (ASV) as front-end features for models. For instance, The combination of linear frequency cepstral coeff icients and inverted MFCCs were highly effective in discerning spoofed and bonafide audio [15]. Further, by combining conventional cepstral processing with the constant Q transformation of speech signals, constant Q cepstral coefficients (CQCCs) demonstrated promising generalization capabilities strong results in ASV [21].

Beyond the development of front-end audio features, advancing back-end classifiers play a critical role in AD detection. For instance, the configuration of a deep neural network (DNN), composed of convolution neural network and gated recurrent units, directly accepted spectrograms as input features with no knowledge-based intervention. In doing so, the extraction of custom acuoustice features was forgone by the end-to-end DNN [24]. Further, by implementing several variants of the residual convolutional neural network (ResNet) according to varying selections of acoustic feature representations (MFCC, Log-magnitude STFT, and CQCC), the efficacy of the front-end features were assessed and the success of the ResNet classifier was reiterated[4]. Expanding on this framework, an ASV scheme that promoted significant data augmentation techniques and the pairing of a transformer encoder with a residual network (TE-ResNet), showed strong results for fake speech detection.

However, these supervised learning processes require substantial computational resources and typically experience a decline in performance when applied to data and techniques that were not encountered during the training phase [3]. Considering the rapidly-advancing nature of AD attacks and techniques, a model's ability to generalize to new attacks is a critical performance metric and a necessary indicator of its genuine applicability. While incorporating a greater number of attack types and combining several detection systems can improve a model's ability to generalize, these adjustments do not address the model's underlying weakness to adapt to novel deepfake techniques.

As such, one-class methods of been experimentally explored due to their intrinsic generalization capabilities. Because they are trained exclusively on a single class, any data that significantly deviates from it is identified as anomalous [27]. In the context of spoofed audio detection, if the model is solely trained on genuine speech data, any audio that significantly

deviates from this will be classified as fake. By removing AD data from the training process, the model will be inherently independent from the technique(s) applied to manipulate the audio, naturally guaranteeing a generalization.

While omitting deepfake data from the model training seems counter-intuitive, there is precedence within the spoofed media detection field for the application this technique. For instance, the one-class method has been successfully applied for video deepfake detection, where the model was exclusively trained on genuine video biometric features [5] [1] [7].

Consequently, the implementation of the person-of-interest (POI) method of training for audio-only datasets showed promised. Using only the assumption of the speaker's identity, the authenticity of an audio segment as real or fake could be reliably determined when trained exclusively on the embedded vectors from the speaker's genuine audio [17]. Embeddings, also known as x-vectors, are compact feature vectors used in speech processing, particularly for speaker recognition. These vectors encapsulate distinct characteristics of a speaker's voice, such as pitch, tone, and manner of articulation, helping to differentiate one speaker from another [20]. Designed to be invariant to the spoken content, background noise, and channel distortions, x-vectors enable effective speaker identification by providing a dense representation that facilitates comparison through distance metrics or classifiers. Their robustness to acoustic variability and adaptability to different recording conditions and noises enhances their utility in real-world speaker recognition tasks.

However, the application of x-vectors and speaker verification techniques for AD detection requires extensive audio data from a specific speaker to classify the speech segments, limiting the model's ability to generalize to any spoken audio. This limitation underscores the importance of this study, which aims to detect spoofed audio by comparing audio embedding, whose authenticity is unknown to the model, to a reference set of embeddings of genuine audio. This approach emphasizes the intrinsic classification of audio clips as genuine or fake based on their content qualities, rather than any specific knowledge of the speaker. Because this framework does not rely on the speaker's identity, the model intrinsically generalizes to speakers that are not directly included in the dataset, enhancing its applicability and versatility.

III. Methodology

A. Input Representation

Mel-Frequency Cepstral Coefficients (MFCCs) are utilized as the primary audio processing feature in speaker verification systems. The configuration specifics are outlined as follows:

Parameter	Value
Sample Frequency	16000 Hz
Frame Length	25 milliseconds
Low Frequency	20 Hz
High Frequency	7600 Hz
Number of Mel Bins	30
Number of Cepstral Coefficients	30
Snip Edges	False

TABLE I Audio Processing Parameters

These MFCC settings ensure robust feature extraction, crucial for the accurate classification and differentiation of speakers in verification tasks.

B. Trunk Architecture

The network initiates with an **Input Layer** that accommodates 30-dimensional feature vectors, each representing a frame of audio extracted from cepstral coefficients, predominantly MFCCs. These features are adept at capturing essential spectral properties critical for differentiating speakers.

Central to the model's architecture are the **Time Delay Neural Network (TDNN) layers**, designed to effectively manage speech sequences, which exhibit significant temporal dynamics:

- **tdnn1-tdnn3**: These layers incorporate context appending to analyze the current frame along with two preceding and two succeeding frames, thereby grasping temporal dependencies across a wide context. This setup, with each layer consisting of 512 units, facilitates intricate pattern learning within the speech signal.
- **tdnn4-tdnn5**: Following tdnn3, tdnn4 persists in processing at 512 units, while tdnn5 increases to 1500 units, emphasizing its critical role in integrating temporal features before pooling. This increase highlights its significance in compiling a comprehensive speech signal representation.

A **Stats Pooling Layer** follows, summarizing the frame-level features into an utterance-level representation by computing statistics—mean and standard deviation—across up to 10,000 frames. This transformation is pivotal for speaker verification, focusing on capturing the overarching speech characteristics essential for identifying speaker identity.

Post-pooling, the architecture transitions to segment-level processing with:

- **tdnn6**: Primarily used for extracting the x-vector, this layer is optimized to distill speaker-specific features into a dense vector, effectively capturing the essence of the speaker's identity.
- **tdnn7**: Further refines the embedding from tdnn6.

The output of the TDNN are sets of 512-dimensional x-vectors. The TDNN's architecture (Fig. 1) is tailored to enhance performance in speaker recognition tasks, leveraging advanced temporal and spectral feature extraction techniques integral to identifying distinct speaker traits.

C. Loss Function

The final output layer of the network includes a log-softmax function. The dimensionality of the output layer is set to 6759, corresponding to the number of distinct classes—speakers, in this case—in the dataset. The softmax loss function is employed to maximize the probability assigned to the correct class (speaker) for each training example, thereby training the network to differentiate effectively between thousands of speakers.

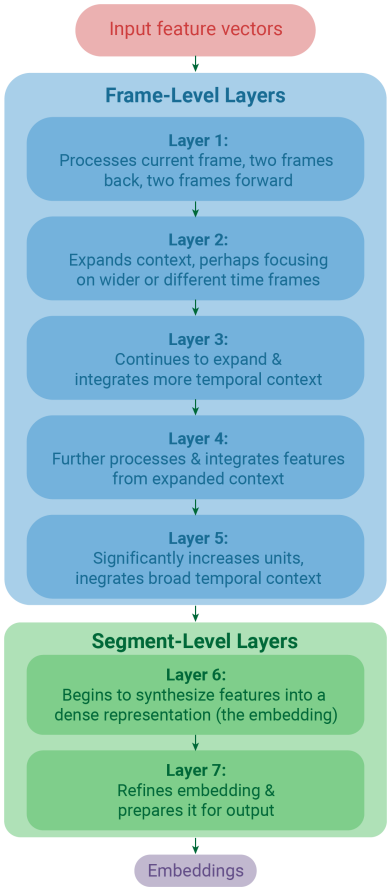

Fig. 1: This diagram illustrates the implemented TDNN architecture. It starts with the input layer receiving audio feature vectors, progresses through several frame-level layers (1-5) that extend the temporal context of the input, and includes a stats pooling layer that summarizes the frame-level features. Finally, in segment-level layers (6-7), the TDNN synthesizes, refines, and outputs a dense speaker embedding for output.

D. Embedding Analysis

This approach leverages genuine audio embeddings as a reference to evaluate the authenticity of test audio embeddings whose validity is unknown. This reference set acts as a standard against which the characteristics of potentially spoofed audio samples are compared. By employing this methodology, the model aims to distinguish genuine audio samples from spoofed ones based on their embedded dissimilarities.

Two primary similarity analysis techniques are employed: Centroid-Based (CB) Similarity and Maximum Similarity (MS) using various metrics (Fig. 2). In the CB method, the mean vector (centroid) of genuine reference embeddings acts as the benchmark, capturing the average properties of authentic audio samples. Test samples are compared to this centroid using cosine, Euclidean, Manhattan, and Minkowski distances to evaluate their similarity. These metrics consider different aspects of distance and orientation in the embedding space.

The MS method enhances this approach by assessing each test embedding against every embedding in the genuine reference set, using the aforementioned distance metrics to ascertain the highest similarity score. This score determines the test sample's closest alignment with any authentic audio sample in the dataset, offering a comprehensive measure for authenticity evaluation.

Both strategies employ a dynamically optimized threshold to distinguish genuine from spoofed samples. A test sample surpassing this threshold is classified as genuine; otherwise, it is marked as spoofed. This dual-threshold approach optimizes both sensitivity (accuracy in identifying genuine samples) and specificity (efficacy in rejecting spoofed samples), crucial for maintaining the system's classification integrity.

IV. Experiments

A. Datasets

1) VoxCeleb: Instead of comparing to a reference set associated with an individual claimed speaker, this approach leverages a vast dataset of genuine audio data. VoxCeleb is a large-scale speaker identification dataset comprising over a million utterances from thousands of speakers, sourced from YouTube videos. This dataset offers a diverse array of voice samples across various genders, accents, and languages, making it an ideal source for generating a broad and representative collection of reference embeddings for authentic audio [12].

In this framework, the reference embeddings are created directly from the VoxCeleb dataset to serve as a benchmark for genuine audio. These embeddings capture a wide variety of vocal characteristics, representative of authentic human speech patterns. The authenticity of each testing audio clip is evaluated by comparing its embeddings with the reference set using techniques such as cosine similarity or maximum-similarity approaches, due to precedence for success with these metrics [17]. A high degree of similarity to the reference embeddings suggests that the audio clip is genuine, whereas a lower similarity indicates potential spoofing. This comparison does not rely on matching to specific speakers but rather to an aggregated or statistical representation of genuine speech patterns across the VoxCeleb dataset. Fig. 3 illustrates the processing pipeline for the VoxCeleb audio files.

2) ASVspoof: The evaluated embeddings were generated from the publicly-available ASVspoof datasets, which are commonly used in AD detection studies, providing a benchmark for evaluating automatic speaker verification systems against various forms of audio spoofing attacks.For training and developing the model, the ASVspoof2017 and ASVspoof2019 were leveraged [23]. While the ASVspoof2017 dataset does not differentiate by attack type, ASVspoof2019 provides data on the following two attack types: logical access (LA), where the attacker employs voice conversion and speech synthesis, and physical access (PA), where the attacker implements replay attacks) [13]. The present model emphasizes the LA attack type. For evaluation, the ASVspoof2021 dataset will be employed, as this edition included deep fakes as a third category of spoofed audio [25]. Fig. 4 illustrates the processing pipeline for the ASVspoof audio files. A summary of the dataset sizes, in terms of speakers and utterances, in provided in Table II.

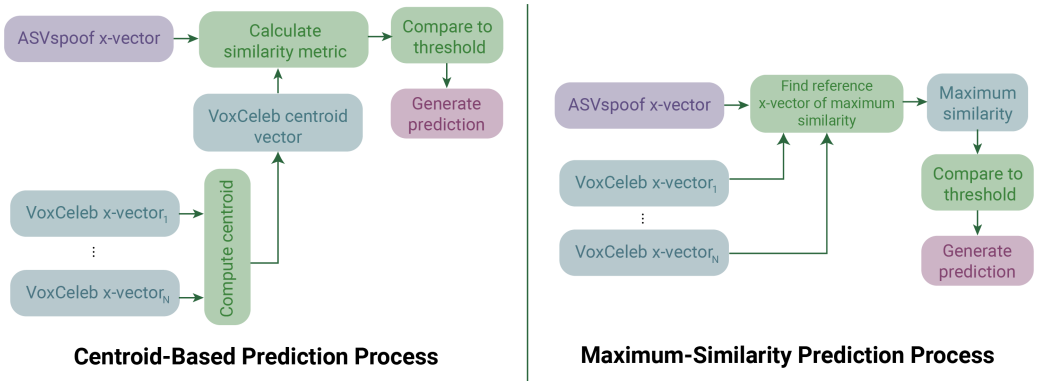

Fig. 2: This diagram depicts two methodologies for testing in speaker verification systems: Centroid-Based (CB) and Maximum-Similarity (MS). The CB method calculates a centroid from the reference set and compares the test audio against this centroid using a similarity metric. In contrast, the MS approach compares the test audio with each reference vector individually, using the highest similarity score for decision-making, offering a robust evaluation mechanism for detecting deepfake audios.

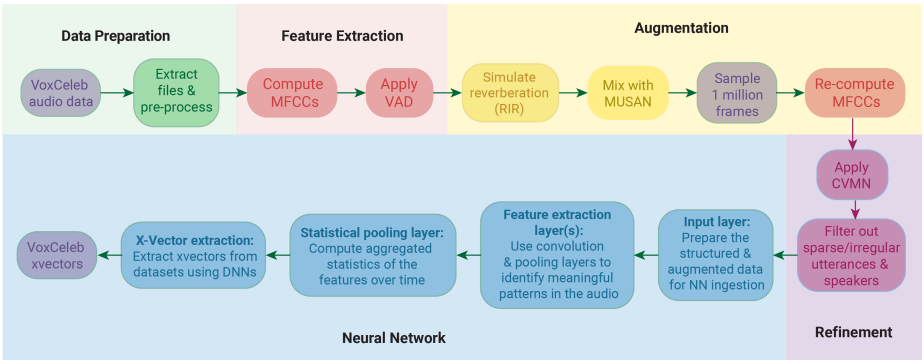

Fig. 3: This diagram outlines the workflow for transforming raw VoxCeleb audio data into embeddings for speaker recognition. Initially, VoxCeleb audio is extracted and preprocessed, followed by MFCC computation and Voice Activity Detection (VAD). The features are then augmented with simulated reverberation, mixed with MUSAN noise, and 1 million frames are sampled and reprocessed for MFCCs. These features are prepped for neural network ingestion, using convolution and pooling layers to detect relevant audio patterns. The process concludes with custom variance normalization and the exclusion of sparse or irregular utterances and speakers to produce the final x-vectors.

TABLE II Summary of Speaker and Utterance Data Across Datasets

Dataset	Subset	Speakers	Genuine Utterances	Spoofed Utterances
VoxCeleb2	Development	5994	1,092,009	-
	Evaluation	118	36,237	-
ASVspoof2017	Training	10	1,508	1,508
	Development	8	760	650
ASVspoof2019	Training	20	2,580	280,000
	Development	20	2,548	22,296
	Evaluation	67	7,355	63,882
ASVspoof2021	Evaluation	93	14,869	519,059

B. Data Augmentation & Refinement

The data augmentation stage enhances the robustness of this speech recognition model, equipping it to perform reliably under diverse and realistic acoustic conditions. This section outlines the augmentation techniques and resources employed to comprehensively augment the training dataset.

1) Room Acoustic Simulation:

- **Room Impulse Responses (RIRs)**: This study utilizes real and simulated RIRs from the RIRS_NOISES database to replicate the acoustic characteristics of various environments. This variability is crucial for training the model to handle different acoustic settings [9]. Each training audio file is convolved with a selection of RIRs, focusing solely on introducing reverberation effects without additional noise.

2) Background Noise Augmentation:

- **MUSAN Corpus**: Employed to simulate realistic background noise conditions, this corpus includes diverse recordings of music, speech, and general noise [19].
- **Music and Noise**: Tracks are mixed with clean speech at varying SNRs (0 to 15 dB), simulating environments ranging from quiet rooms to noisy streets.

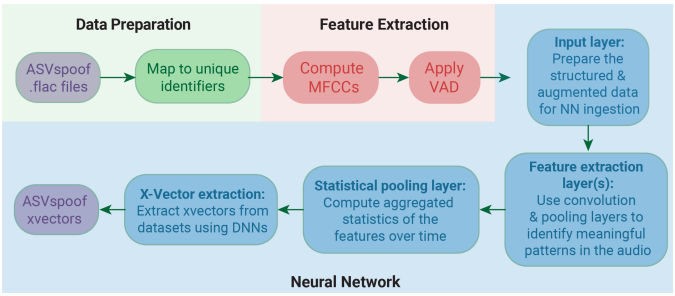

Fig. 4: This diagram depicts the process of transforming raw ASVspoof audio data into x-vectors. After mapping unique identifiers to their corresponding audio files MFCCs are calculated, and Voice Activity Detection is applied. The resulting features are processed by a neural network, beginning with an input layer that prepares data for ingestion. Subsequent convolutional and pooling layers in the feature extraction stage identify meaningful patterns, while a statistical pooling layer aggregates feature statistics over time. DNNs then extract the xvectors for use in downstream tasks.

- **Speech/Babble**: Overlapping speech tracks create a 'babble' effect, mixed at higher SNRs (13 to 20 dB), to mimic crowded environments.

The final training set merges this augmented subset with the original clean data, effectively doubling its size. This expanded dataset includes a broad array of acoustic scenarios, crucial for developing a speech recognition system that performs well across varied real-world environments.

3) Data Refinement: To refine the dataset, the model applies Cepstral Mean and Variance Normalization to both clean and augmented speech to reduce variability caused by different recording conditions and channel characteristics. Additionally, nonspeech frames are removed to enhance signal quality and focus on speaker-specific traits. Subsequently, utterances that are too short are filtered out of the dataset.

C. Implementation Details

The implementation of the project leverages a virtual machine based on an n1-highmem-16 machine type. It operates under an Intel Haswell CPU platform with a 64-bit architecture and is equipped with 16 virtual CPUs, providing robust processing power. For graphical processing and acceleration, the VM includes an NVIDIA Tesla P100 GPU. This GPU is essential for efficiently processing deep learning models, particularly those involving convolutional neural networks, which are compute-intensive and benefit significantly from GPU acceleration.

D. Embedding Analysis

The x-vector analysis was implemented through a Python script using the PyTorch library, designed for practical evaluation and classification of audio embeddings [16]. The script is structured to process audio data in batches, making it suitable for handling large datasets efficiently. The initialization of the threshold optimized using the precision-recall characteristics of the training dataset to maximize the F1 score. This process involved adjusting the threshold based on the performance metrics obtained during the evaluation of the development set, fine-tuning it to enhance both precision and recall effectively.

The experiment outputs several key performance metrics that help assess the efficacy of the employed similarity metrics. These include accuracy, F1 score, and potentially the Equal Error Rate (EER). These metrics provide quantitative insights into the classification performance of each method. Additionally, Detection Error Tradeoff (DET) curves and plots depicting the distribution of similarity scores across different labels were generated [10]. These visualizations offer a comprehensive view of the thresholds and distribution of similarity scores, illustrating how well each method can differentiate between real and spoofed audio samples.

E. Scoring

Scoring is applied to quantify the similarity between the test samples and the references using various distance metrics. Each test sample is scored against the aggregate model derived from the reference samples. For the Centroid-Based method, the test sample is compared to the centroid of the reference set using the following metrics: Cosine, Euclidean, Minkowski, and Manhattan distances. Each metric offers a different perspective on the data's geometric structure, influencing the similarity scores. For Maximum-Similarity, each test sample is scored against all samples in the reference set, and the highest score is selected as the representative score.

F. Evaluation Metrics

The primary evaluation metrics used in this study include Accuracy, F1 Score, and the EER. Accuracy and F1 Score provide insights into the overall effectiveness of the similarity scoring method at classifying the samples correctly. The EER is particularly critical as it represents the point at which the false positive rate equals the false negative rate, providing a balanced metric that is especially useful in biometric systems such as speaker verification. The DET curve, plotting the false negative rate against the false positive rate on a logarithmic scale, visually complements the EER by illustrating the trade-offs between type I and type II errors across various thresholds.

G. Results

The AD detection methods demonstrates varying levels of effectiveness across different similarity metrics. The results, consolidated from multiple performance evaluations, indicate a clear superiority of the Cosine similarity metric over others like Euclidean, Minkowski, and Manhattan in terms of both accuracy and F1 scores. Specifically, in the evaluation dataset, Cosine similarity achieved the highest accuracy (78.91%) and F1 score (40.65%) (Table III). This suggests that Cosine similarity, which measures the cosine of the angle between two vectors, is more effective in capturing the inherent characteristics of genuine versus spoofed audios without being misled by magnitude differences.

TABLE III Performance Metrics of Centroid-Based (CB) and Maximum-Similarity (MS) Methods

Method	Metric	Dataset	Accuracy (%)	F1 Score (%)
Centroid-Based	Cosine	Train	80.24	42.19
		Dev	79.55	41.22
		Eval	78.91	40.65
	Euclidean	Train	71.68	33.42
		Dev	77.24	30.91
		Eval	70.68	28.31
	Minkowski	Train	64.68	33.42
		Dev	77.24	24.91
		Eval	66.68	17.31
	Manhattan	Train	67.19	33.23
		Dev	52.39	22.05
		Eval	50.56	31.31
Max. Similarity	—	Train	14.39	25.17
		Dev	12.39	22.05
		Eval	18.57	31.31

The Euclidean and Minkowski metrics, which focus more on geometric distances, exhibited significantly lower performance than the Cosine similarity metric, particularly in the evaluation set. Manhattan distance, characterized by its summation of the absolute differences of their coordinates, performed the poorest, especially highlighted by its highest EER of 44.30% (Table IV). This high EER implies a less favorable balance between false positives and negatives, indicating poor generalization when encountering diverse audio samples. The Maximum Similarity method, on the other hand, yields markedly lower performance metrics across all datasets. This could imply that while searching for the maximum similarity helps identify the closest matches within the dataset, it may not effectively discriminate between genuine and spoofed audios, possibly due to overfitting to specific characteristics of the most similar genuine samples.

TABLE IV Equal Error Rates (EERs) of Centroid-Based Methods

Metric	EER (%)
Cosine	26.51
Euclidean	33.01
Minkowski	35.90
Manhattan	44.30

The DET curve visually supports these findings. The curve for Cosine similarity is closest to the origin, suggesting a lower rate of both false negatives and positives—ideal traits for security systems reliant on speaker verification. The curves for Euclidean and Minkowski metrics show moderate performance, while the Manhattan curve's position furthest from the origin corroborates its inadequate performance in this application setting.

By using only audio embeddings as an input feature, this work progresses towards discerning underlying structural differences between spoofed and genuine audio. However, the generalization this enables comes with a notable trade-off in prediction accuracy. In comparison to Pianese et al. [17], who also incorporated speaker identity into their audio embeddings for AD detection, their approach demonstrates the significant benefits of using specific speaker traits to enhance detection

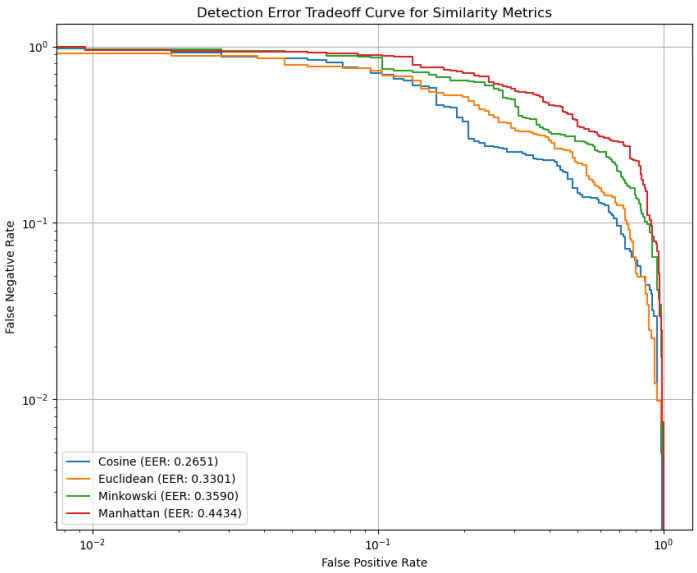

Fig. 5: The provided DET curve illustrates the performance of various similarity metrics by plotting the False Negative Rate against the False Positive Rate on logarithmic scales. The curves displays different metrics: Cosine, Euclidean, Minkowski, and Manhattan, with each curve annotated with the EER for the respective metric.

accuracy, as reflected in their remarkably low Equal Error Rates (EERs) and high Area Under Curve (AUC) metrics with both Centroid-Based (CB) and Maximum-Similarity (MS) analyses on the ASVspoof2019 dataset.

The superior performance of their model, consistently achieving EERs below one percent, underscores the potential advantages of integrating spoken features in addition to the audio embedding. Further refinement of feature extraction and embedding processes could also be explored to enhance the model's ability to differentiate between spoofed and genuine audio more effectively.

V. Conclusion

This study proposes an approach to audio deepfake (AD) detection that emphasizes intrinsic audio content qualities over speaker identity. While promising, significant advancements are necessary for this technique to compete effectively in the AD detection landscape.

Looking ahead, there are several avenues to enhance and expand this model's capabilities. To refine the model without compromising its generalizability, integrating additional features common in the AD detection field could offer more robust guidance. These features include spectral properties like spectral flux, spectral roll-off, and spectral bandwidth, as well as prosodic features such as intonation patterns, speaking rate, and rhythm [26]. These enhancements could help detect unnatural speech patterns often found in synthesized or altered audio, thereby solidifying the approach as a viable option within the broader spectrum of AD detection methods.

Additionally, exploring more advanced neural network architectures and delving deeper into unsupervised or semi-supervised learning models could further the system's ability

to adapt to new and evolving spoofing techniques. By continuing to balance sensitivity and specificity and improving upon the dynamic thresholding methods, this framework could become increasingly effective in security-sensitive environments, making significant strides in the fight against audio deepfake threats.

REFERENCES

[1] Shruti Agarwal, Tarek El-Gaaly, Hany Farid, and Ser-Nam Lim. Detecting deep-fake videos from appearance and behavior, 2020.

[2] Federico Alegre, Artur Janicki, and Nicholas Evans. Re-assessing the threat of replay spoofing attacks against automatic speaker verification. In *2014 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–6, 2014.

[3] Zaynab Almutairi and Hebah Elgibreen. A review of modern audio deepfake detection methods: Challenges and future directions. *Algorithms*, 15(5), 2022.

[4] Moustafa Alzantot, Ziqi Wang, and Mani B. Srivastava. Deep residual neural networks for audio spoofing detection, 2019.

[5] Matyáš Boháček and Hany Farid. Protecting world leaders against deep fakes using facial, gestural, and vocal mannerisms. *Proceedings of the National Academy of Sciences*, 119(48), November 2022.

[6] Deepak Ramesh Chandran. Use of ai voice authentication technology instead of traditional keypads in security devices. *Journal of Computer and Communications*, 10(06):11–21, 2022.

[7] Davide Cozzolino, Andreas Rössler, Justus Thies, Matthias Nießner, and Luisa Verdoliva. Id-reveal: Identity-aware deepfake video detection, 2021.

[8] Rosa González Hautamäki, Tomi Kinnunen, Ville Hautamäki, Timo Leino, and Anne-Maria Laukkanen. I-vectors meet imitators: on vulnerability of speaker verification systems against voice mimicry. In *Proc. Interspeech 2013*, pages 930–934, 2013.

[9] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L. Seltzer, and Sanjeev Khudanpur. A study on data augmentation of reverberant speech for robust speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5220–5224, 2017.

[10] A. Martin, G. Doddington, Terri Kamm, M. Ordowski, and Mark Przybocki. The det curve in assessment of detection task performance. *The DET Curve in Assessment of Detection Task Performance*, pages 1895–1898, 01 1997.

[11] Masanori MORISE, Fumiya YOKOMORI, and Kenji OZAWA. World: A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Transactions on Information and Systems*, E99.D(7):1877–1884, 2016.

[12] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. Voxceleb: Large-scale speaker verification in the wild. *Computer Speech Language*, 60:101027, 2020.

[13] Andreas Nautsch, Xin Wang, Nicholas Evans, Tomi H. Kinnunen, Ville Vestman, Massimiliano Todisco, Héctor Delgado, Md Sahidullah, Junichi Yamagishi, and Kong Aik Lee. Asvspoof 2019: Spoofing countermeasures for the detection of synthesized, converted and replayed speech. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(2):252–265, 2021.

[14] Sergey Novoselov, Alexandr Kozlov, Galina Lavrentyeva, Konstantin Simonchik, and Vadim Shchemelinin. Stc anti-spoofing systems for the asvspoof 2015 challenge. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5475–5479, 2016.

[15] Monisankha Pal, Dipjyoti Paul, and Goutam Saha. Synthetic speech detection using fundamental frequency variation and spectral features. *Computer Speech Language*, 48:31–50, 2018.

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[17] Alessandro Pianese, Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Deepfake audio detection by speaker verification, 2022.

[18] Berrak Sisman, Junichi Yamagishi, Simon King, and Haizhou Li. An overview of voice conversion and its challenges: From statistical modeling to deep learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:132–157, 2021.

[19] David Snyder, Guoguo Chen, and Daniel Povey. Musan: A music, speech, and noise corpus, 2015.

[20] David Snyder, Daniel Garcia-Romero, Alan McCree, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. Spoken Language Recognition using X-vectors. In *Proc. The Speaker and Language Recognition Workshop (Odyssey 2018)*, pages 105–111, 2018.

[21] Hemlata Tak, Jose Patino, Andreas Nautsch, Nicholas Evans, and Massimiliano Todisco. An explainability study of the constant q cepstral coefficient spoofing countermeasure for automatic speaker verification, 2020.

[22] Massimiliano Todisco, Héctor Delgado, and Nicholas Evans. Constant q cepstral coefficients: A spoofing countermeasure for automatic speaker verification. *Computer Speech amp; Language*, 45:516–535, Sep 2017.

[23] Massimiliano Todisco, Xin Wang, Ville Vestman, Md Sahidullah, Hector Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi Kinnunen, and Kong Aik Lee. Asvspoof 2019: Future horizons in spoofed and fake audio detection, 2019.

[24] Jee weon Jung, Hye jin Shim, Hee-Soo Heo, and Ha-Jin Yu. Replay attack detection with complementary high-resolution information using end-to-end dnn for the asvspoof 2019 challenge, 2019.

[25] Junichi Yamagishi, Xin Wang, Massimiliano Todisco, Md Sahidullah, Jose Patino, Andreas Nautsch, Xuechen Liu, Kong Aik Lee, Tomi Kinnunen, Nicholas Evans, and Héctor Delgado. Asvspoof 2021: accelerating progress in spoofed and deepfake speech detection, 2021.

[26] Jiangyan Yi, Chenglong Wang, Jianhua Tao, Xiaohui Zhang, Chu Yuan Zhang, and Yan Zhao. Audio deepfake detection: A survey, 2023.

[27] You Zhang, Fei Jiang, and Zhiyao Duan. One-class learning towards synthetic voice spoofing detection. *IEEE Signal Processing Letters*, 28:937–941, 2021.