K-means to Spectral Clustering: Improving Diarization Accuracy with ECAPA-TDNN Embeddings

Chanchla Tripathi
chanchalatripathi@gmail.com

Anshu Vairagade
anshuvairagade7@gmail.com

Jagdish Kachhawah
jagdishkachhawahjk@gmail.com

Ekta Tajne
tajneekta@gmail.com

Vaishnavi Asare
vaishnaviasare16@gmail.com

Kanishka Katekhaye
kanishkakatekhaye@gmail.com

CSE Department, Yeshwantrao Chavan College of Engineering, Nagpur, India

Abstract

The research focuses on creating a system which determines speakers and their related time allocations within multiple speaker audio data. Our diarization system adopts speaker embeddings derived from ECAPA-TDNN technology together with spectral clustering to segment and cluster speech sections based on speaker identities. The assessment of the system depends on TOEFL English practice along with VoxCeleb dataset conversation audio while combining input files creates conditions similar to real-world multi-speaker scenarios. Spectral clustering goes beyond K-means because it detects intricate similarity relationships between speaker embeddings which produces stronger resistance against challenging acoustic conditions like overlapping speech and speaker similarity. Our approach demonstrates the potential for high-accuracy, scalable speaker diarization in diverse and noisy conversational environments. The results were promising: our approach was able to accurately identify and separate multiple speakers in a single audio file. This research adds to the growing field of speaker diarization by showing that combining ECPA-TDNN embedding with the right clustering method can lead to better performance, even in tricky audio scenarios. These findings could help improve the accuracy and efficiency of systems used in everything from automated transcription to smart assistants and audio analytics.

***Index Terms*—Speaker diarization, ECAPA-TDNN, spectral clustering, voice segmentation.**

I. Introduction

Audio recordings require the identification of voice individuals and their temporal position through speaker diarization which serves as an essential element of speech processing. Speaker diarization processing allows thorough examination and refinement for various domains like news broadcasting (anchor clarity evaluation) and group discussions (participation equality) and political discourse (public trust support), educational training (public speaking skills) and media production (speaker engagement discovery and audience preferences).

Speaker diarization faces significant challenges [15] while becoming more important because it struggles to work in real-world environments involving two or more speakers and periods of silence as well as voices with similar attributes. The application of MFCC clustering as well as D-vectors [9,11] to speaker segmentation faces limitations in inadequate complexity processing that results in accuracy deterioration and higher confusion levels between segments. A speaker diarization framework combines ECAPA-TDNN-based speaker embeddings [12] and spectral clustering as a solution to overcome current method limitations. Speaker embedding discriminative power gets improved through this approach while similarity-based clustering detects non-linear speaker relations. The clustering algorithm of spectral methods outperforms K-means because it successfully handles difficult acoustic conditions related to speaker similarity together with overlapping speech. The framework receives testing through multitrack audio recordings made up of speakers from TOEFL English practice conversations along with the VoxCeleb dataset [5]. Spectral clustering achieves better results than K-means in actual situations when tested through diarization quality evaluations. Visual displays using affinity matrices demonstrate structural differences brought by our method in the research. The system achieves high accuracy through the clustering quality evaluation that uses Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) to demonstrate matching results with human-annotated ground truth.

In summary, our contributions include:

1) This research introduces a speaker diarization system that implements spectral clustering with ECAPA-TDNN embeddings.
2) The research executed benchmark testing using authentic dataset information against K-means clustering.
3) The system demonstrates operational functionality even in challenging acoustic conditions.
4) Performance visualization for clustering is achieved through comprehensive analysis of affinity matrices.
5) Experimental results confirm the system's higher accu-

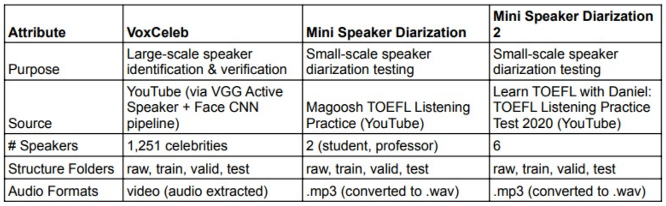
racy, scalability, and reliable cluster functionality.

II. Acoustic Embedding Preparation

2.1 Dataset Description

We selected ten audio files containing the multiple speakers in different scenario such as meeting, conversation and interview. Table 2.1 shows the complete description of dataset use for selecting the test audio files.

Table 2.1 Summary of dataset Used



Attribute	VoxCeleb	Mini Speaker Diarization	Mini Speaker Diarization 2
Purpose	Large-scale speaker identification & verification	Small-scale speaker diarization testing	Small-scale speaker diarization testing
Source	YouTube (via VGG Active Speaker + Face CNN pipeline)	Magoosh TOEFL Listening Practice (YouTube)	Learn TOEFL with Daniel: TOEFL Listening Practice Test 2020 (YouTube)
# Speakers	1,251 celebrities	2 (student, professor)	6
Structure Folders	raw, train, valid, test	raw, train, valid, test	raw, train, valid, test
Audio Formats	video (audio extracted)	.mp3 (converted to .wav)	.mp3 (converted to .wav)

2.2 Voice Activity Detection and Preprocessing

VAD stands as the first operation in our speaker diarization pipeline because it works to separate speech segments from silent and ambient noise sections during preprocessing. A deep learning model from PyAnnote-Audio [13] performs VAD analysis with frame-level accuracy up to 0.01s so we can use its pretrained model effectively. Energy-based and heuristic methods are surpassed notably by this method while operating in noisy auditory conditions. The model requires 16 kHz sampling rate with mono channel audio so the input audio receives both physical resampling and stereo channel combination into mono format. Short audio segments that do not meet the processing window receive zero padding treatment until they reach the window size and all speech-detected segments are applied to the segmentation and embedding components.

2.3 Segmentation Strategy

After VAD completes its speech separation task the next operation segments the audio into overlapping sections to enhance embedding extraction reliability. The individual speech segments receive a processing that divides them into 1.5-second sections using a 0.75-second stride which results in fifty percent overlapping segments. The overlapping windows help both deal with boundary effects and improve the ability to detect when speakers change. The fixed-length chunking technique provides better robustness because it follows ECAPA-TDNN training protocols [12] and keeps track of brief speaker characteristics. The application of overlap ensures that segments show continuous context between each other across boundaries which helps prevent model fragmentation while making the model more sensitive to the speaker dynamics between segments.

2.4 Embedding Extraction with ECAPA-TDNN

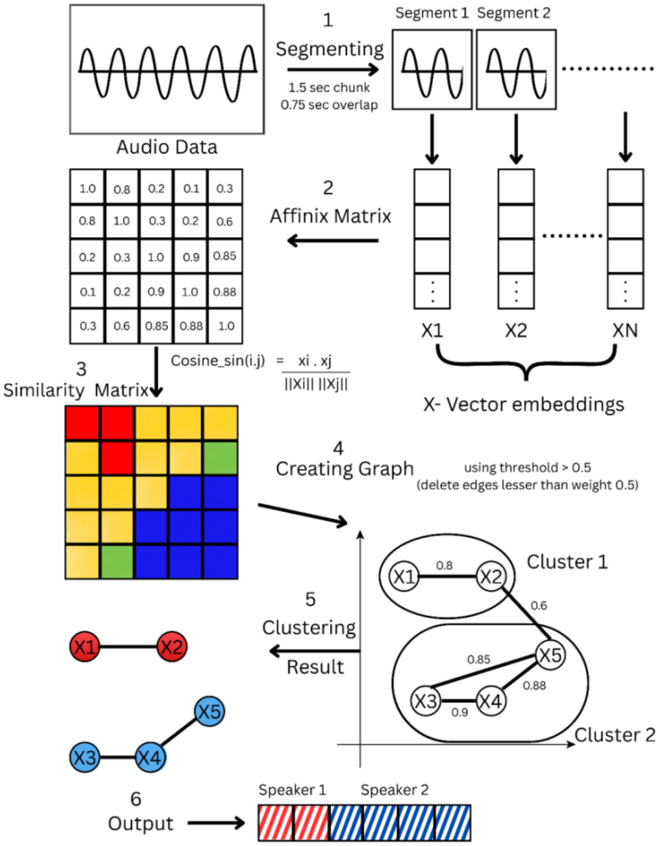
The Input segments pass through the ECAPA-TDNN model running on SpeechBrain to produce 512-dimensional speaker embeddings also known as x-vectors. The framework adopts both channel attention and residual connections to extract precise properties of vocal identities from the input. x-vectors function as data compression methods [6] for representing distinctive speaking traits of individual speakers.

2.5 Feature Representation and Robustness

MFCC-based features serve as a source for embedding extraction which monitors spectral envelope in addition to timbre characteristics and frequency patterns. A process which incorporates speaker confidence measurements and contextual information delivers reliability when operating under noisy and overlapping situations. The process requires standardized preprocessing for every input audio file. The system excludes corrupted or unreadable files from the processing. A disabled gradient computing process during embedding extraction generates speaker descriptors which become ready for subsequent clustering functions.

III. Methodology

Speaker diarization — figuring out who is speaking when during an audio recording — follows a clear, structured process. The workflow breaks down the steps used in this research to divide, represent, and cluster speech into different speaker sections.



1.0	0.8	0.2	0.1	0.3
0.8	1.0	0.3	0.2	0.6
0.2	0.3	1.0	0.9	0.85
0.1	0.2	0.9	1.0	0.88
0.3	0.6	0.85	0.88	1.0

Fig. 3.1 Speaker diarization pipeline workflow

Fig. 3.1 shows overview of the speaker diarization process using spectral clustering. The input audio is segmented into overlapping chunks and each segment is converted into an x-vector embedding. A cosine similarity matrix is computed to form an affinity matrix. A similarity graph is then created by retaining only edges with weights greater than 0.5. Spectral clustering is applied to this graph to identify speaker clusters, and the final diarization output assigns speaker labels to segments accordingly.

3.1 Audio Preprocessing and Embedding Creation

Everything kicks off with the raw audio input. This audio is split into overlapping chunks, usually around 1.5 seconds long with a 0.75-second overlap. Overlapping the segments helps catch speaker changes more smoothly, without jarring cuts. Each of these chunks is then passed through a pre-trained deep neural network, which pulls out x-vector embeddings.

3.2 Affinity Matrix Construction

Once the embeddings are extracted from all the audio pieces, the next thing we need to figure out is how similar they are to each other. For that, we build what's called an affinity matrix. Now, in the simplest case, the idea is pretty straightforward — you just check how close two embeddings are using something called cosine similarity, and the formula for that goes like this:

$$\text{cosine\_sim}(a, b) = \frac{x_a \cdot x_b}{\|x_a\| \|x_b\|} \tag{1}$$

Equation (1) does the job when things are simple, but speech data in real life is messy — people talk differently, speeds vary, and so on. So, instead of sticking with just cosine similarity, we step it up by using spectral clustering. Here, the affinity matrix, which we'll call A, is built a little differently. It uses something called the Kullback-Leibler (KL) divergence, and this involves fitting a Gaussian Mixture Model (GMM) [2] to each audio segment first. The formula for this updated affinity matrix looks like:

$$A[a][b] = e^{\left(-\frac{d^2(s_a, s_b)}{\sigma^2}\right)} \tag{2}$$

In equation (2), the distance between any two segments, say s_i and s_j, is calculated using a symmetric version of KL divergence, and that's defined by:

$$d(s_a, s_b) = \text{KL}(f \| g) + \text{KL}(g \| f) \tag{3}$$

Here, f and g are just the names for the GMMs fitted to segments s_i and s_j. By doing it this way, we get a much richer sense of how similar two pieces of speech really are, even if they're a bit noisy or different in style.

3.3 Graph Construction and Laplacian Computation

After we've got the affinity matrix sorted out, the next move is to build a similarity graph. In this graph, each node stands for a speech segment, and the edges connecting them show how similar the segments are, based on the values we calculated earlier. Trim away the edges that have really low similarity scores using a threshold < 0.5. Once the graph is built, we need a way to dig into its structure more deeply. That's where the Laplacian matrix, L, comes in. Basically, it helps us capture important information about how the nodes are linked. There are two versions of the Laplacian we look at. First, the unnormalized Laplacian, which is given by:

- **Unnormalized Laplacian:**

$$L = D - A \tag{4}$$

And the normalized Laplacian, which is calculated like this:

- **Normalized Laplacian:**

$$L_{\text{sym}} = D^{-1/2} A D^{-1/2} \tag{5}$$

In both cases, D is what's called the degree matrix — you get it by summing up the weights of all edges connected to a node, so mathematically it's:

$$D_{ab} = \sum_b A_{ab}. \tag{6}$$

The normalized version, L_{sym}, is especially handy because it keeps the similarities balanced, even when different nodes have different numbers of connections. This makes it easier later when we try to cluster the nodes into meaningful groups.

3.4 Eigen Gap Heuristic for Determining Optimal Number of Clusters

The Laplacian matrix L obtained through equation (4) and (5), either unnormalized or normalized, undergoes eigenvalue decomposition to reveal essential structural information from the similarity graph. This decomposition yields eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ and corresponding eigenvectors $v_1, v_2, \ldots, v_n$, which capture spatial relationships among speech segments. The spectral embedding space results from projecting original embeddings through the eigenvectors corresponding to the smallest eigenvalues (excluding zero for normalized Laplacians) [1]. This transformation restructures the data to reduce noise and improve speaker separability by creating a space where clusters are more distinguishable. The transformed representation is defined as:

$$Y = [v_1, v_2, \ldots, v_K] \in \mathbb{R}^{n \times K} \tag{7}$$

where K is the number of clusters (speakers) to be identified and n is the number of speech segments. The matrix Y calculated using equation (7) consists of the top K eigenvectors arranged as columns. The appropriate number of clusters K is determined using the Eigen Gap Heuristic. This technique analyzes the sorted eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ and selects K based on the index at which the largest difference between consecutive eigenvalues occurs:

$$K = \arg\max_i (\lambda_{i+1} - \lambda_i) \tag{8}$$

This approach allows the clustering process to be guided by the inherent structure of the similarity graph, reducing the risk of overfitting or underfitting caused by manual selection of K.

3.5 Clustering via K-means

After determining the optimal K, the data embedded in spectral space Y is clustered using the K-means algorithm. K-means partitions the data into K groups by minimizing the within-cluster variance. Formally, it solves:

$$\underset{C_1,\ldots,C_K}{\arg\min} \sum_{k=1}^{K} \sum_{y_i \in C_k} \|y_i - \mu_k\|^2 \tag{9}$$

where μ_k denotes the centroid of cluster C_k, and y_i represents the i-th data point in the reduced space.

This clustering step in equation (9) assigns each speech segment to its corresponding speaker based on proximity in the transformed space. The integration of spectral embedding, the Eigen Gap Heuristic, and K-means clustering enhances the diarization system's ability to accurately identify speakers, especially under challenging acoustic conditions or when the number of speakers is unknown.

3.6 Post-processing

After the previous operations two methods of label smoothing alongside Cross-EM refinement operate together to detect and rectify minor errors during the process of clarifying data. The final step is just making sure we've got a nice, smooth timeline where we can clearly see who's talking when.

IV. Experiment

The test data we used in our experiments are audio records from TOEFL English practice test YouTube Channel and VoxCeleb dataset audio that contains conversation between multiple speakers with scenarios like school, meeting, interview etc. There are ten audio records which we segmented into 1.5 seconds with 0.75 second of overlap. All the ten audio files were labelled by human annotators to form the ground truth for performance evaluations.

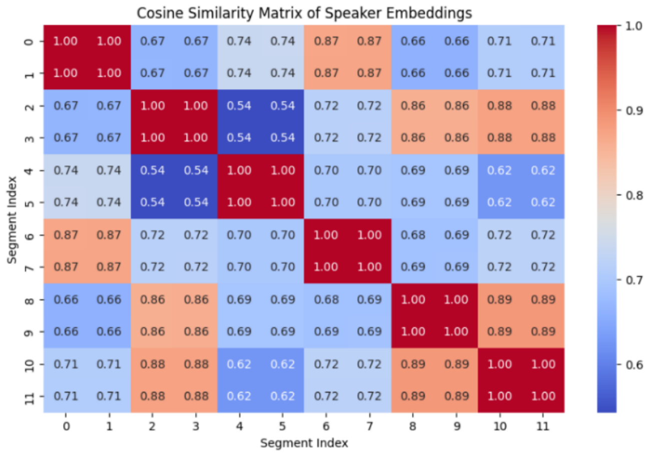


Fig. 4.1 Similarity matrix for speaker embeddings

Fig. 4.1 illustrates the cosine similarity matrix constructed using speaker embeddings extracted via the ECAPA-TDNN model. Each cell in the matrix represents the cosine similarity between pairs of audio segments, with values ranging from 0.54 to 1.00. Diagonal entries exhibit a perfect similarity score of 1.00, reflecting the self-similarity of each segment. The ECAPA-TDNN model was employed due to its proven robustness in capturing discriminative speaker characteristics from speech signals. The resulting embeddings provide a compact yet informative representation of speaker-specific features. By computing pairwise cosine similarity among these embeddings, we can quantitatively assess the similarity between segments in terms of speaker identity. This similarity matrix plays a crucial role in our speaker diarization pipeline by serving as the input for spectral clustering. The clear blockwise high-similarity patterns observed (e.g., segments [2, 3], [6, 7], and [10, 11]) indicate recurring speaker identities and validate the discriminative capability of the extracted embeddings. These patterns assist the spectral clustering algorithm in grouping segments belonging to the same speaker, thereby enhancing diarization performance and ensuring speaker consistency across non-contiguous speech segments.

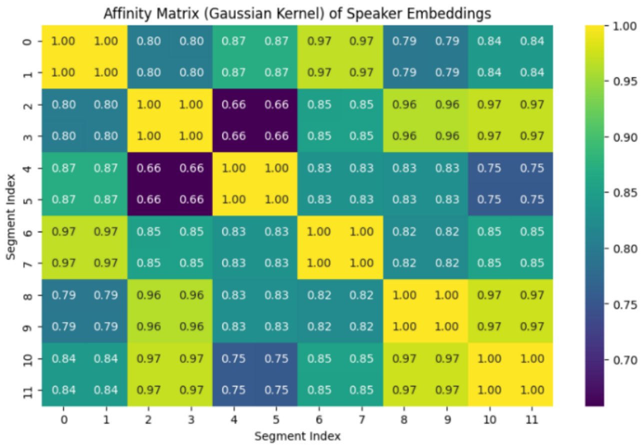
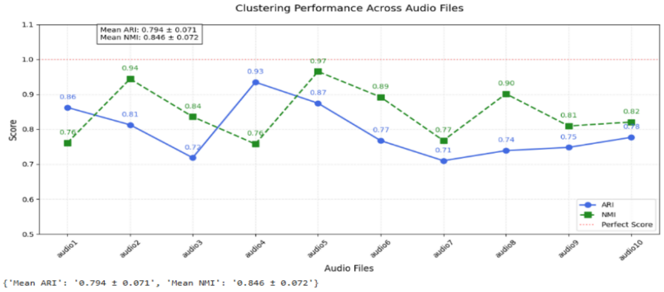


Fig. 4.2 Affinity matrix of speaker embedding

A Gaussian (RBF) kernel was applied to ECAPA-TDNN-based speaker embeddings cosine similarity matrix to generate the affinity matrix displayed in fig. 4.2 . The conversion of original similarity ratings transforms them into an appropriate format for graph clustering models through a method which strengthens nearby connections while weakening global relations. High values found in the affinity matrix reveal powerful bond strengths between speaker segments as they approach a value of 1.0. The spectral clustering algorithm requires input from the matrix which transforms into the weighted adjacency matrix of a similarity graph. The detection of speaker clusters becomes more accurate due to the consistent patterns found in high-affinity blocks among segments [2, 3], [6, 7] and [10, 11]. The correct implementation of this step will boost clustering performance which leads to producing robust and well-separated results for speaker diarization within the proposed system pipeline.

Spectral Clustering Evaluation

The fig. 4.3 shows how spectral clustering of the similarity matrix performs on the proposed speaker diarization pipeline for ten audio files. Clustering performance assessment consists of two standard evaluation metrics [16] named Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) that measure the similarity between predicted speaker labels and reference annotations. The measurement data reveals solid and reliable distribution that produces results showing 0.794 ± 0.071 average ARI alongside 0.846 ± 0.072 average NMI. The majority of files show strong mutual information linking predicted clusters to actual clusters as demonstrated by rising NMI scores that reach 0.97 in audio05 data. The model demonstrates effective segment grouping ability by maintaining a high score on ARI together with ARI levels which slightly reduced compared to other metrics.



```
{'Mean ARI': '0.794 ± 0.071', 'Mean NMI': '0.846 ± 0.072'}
```

Fig. 4.3 Clustering Performance Across Audio Files

The diversification between audio files stems from three main factors: speakers talking simultaneously, environmental noises and variations in speaking duration. The pair of audio04 and audio05 delivered optimal clustering affiliations but audio03 and audio07 showed decreased performance in speaker differentiation due to acoustic conditions.

The research demonstrates that both similarity-based affinity matrix processing and spectral clustering enable effective speaker diarization by obtaining high clustering accuracy across various audio environments.

V. Result

We adopt the ”diarization error” metric as defined by the NIST Rich Transcription Evaluation [14], which provides a standardized way to assess speaker diarization performance. This metric serves as the primary evaluation criterion for enabling consistent and comparable results.

$$d_{\text{err}} = \frac{T_{\text{false\_alarm}} + T_{\text{miss}} + T_{\text{wrong}}}{T_{\text{ref}}} \tag{10}$$

Where d_{err} is diarizarion error rate. $T_{\text{false\_alarm}}$ is total length of non-speech segments classified as speech. T_{miss} is total length of speech segments classified as non-speech or silence. T_{wrong} is total length of speech segments classified as speech but assigned to the wrong speaker. T_{ref} is total length of all speech segments in the ground truth.

In addition to d_{err}, we also introduce the following purity metric:

$$\text{purity} = \frac{T_{\text{pure}}}{T_{\text{speaker}}} \tag{11}$$

Where T_{pure} is pure time and T_{speaker} is total speaker time.

For each speaker identified by the system, we find a reference speaker from the ground truth that shares the longest time with the system speaker. The pure time is the sum of all these shared times. The purity metric is useful for the applications which care less about over-segmentation (i.e., one speaker may be separated into multiple clusters) but more about the ”cleanliness” of each cluster.

The Diarization Error Rate (DER) is computed using the pyannote.metrics library [13], with a 0.25-second collar and ignoring overlapping speech, following standard evaluation protocol.

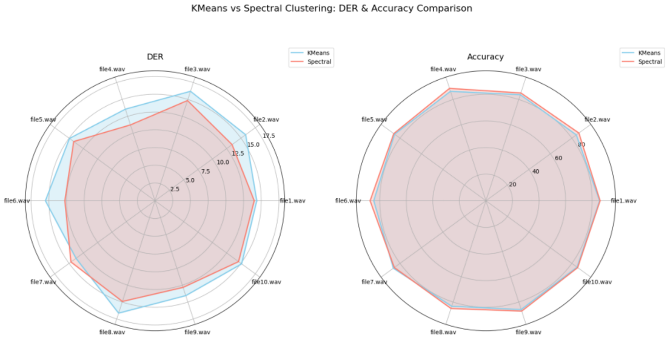
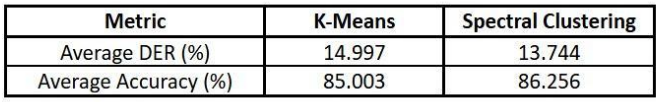


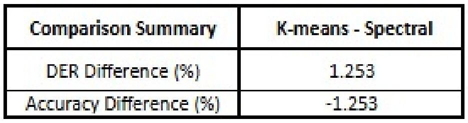
Fig. 5.1 K-means vs Spectral Clustering: DER and Accuracy Comparison

Table 5.1 Average performance metrics



Metric	K-Means	Spectral Clustering
Average DER (%)	14.997	13.744
Average Accuracy (%)	85.003	86.256

Table 5.2 Comparison of average performance metrics



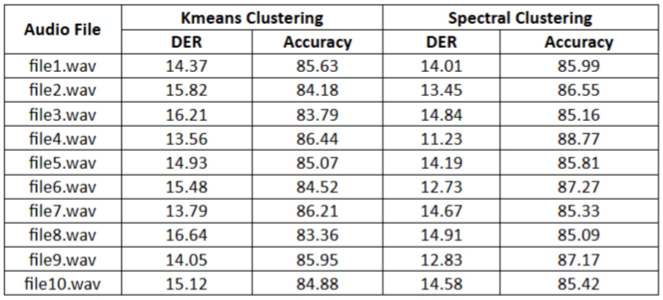
Comparison Summary	K-means - Spectral
DER Difference (%)	1.253
Accuracy Difference (%)	-1.253

The fig. 5.1 presents a performance comparison between K-means and Spectral Clustering in determining diarization error rate (DER) across ten audio files. The speaker diarization metric DER integrates all identification problems including missed speech calls together with incorrect alarms and speaker mixing. As shown in table 5.1 the implementation of spectral clustering for our system produces 13.74% Diarization Error

Rate while achieving 86.26% clustering purity. K-Means-based system demonstrates average DER at 14.99% together with system accuracy at 85.00%. The spectral clustering method brings enhanced speaker separation together with more precise label assignment than K-Means by achieving 1.25% better Diarization Error Rate and 1.26% increased clustering accuracy.

Table 5.1 demonstrates how the systems perform regarding their average metrics. The spectral clustering algorithm demonstrates consistent improvement in both error rate reduction and clustering purity which creates better accuracy and reliability for diarization systems as shown in table 5.2. Spectral clustering produces DER results that are 1.25% lower while delivering clustering accuracy measurements that are 1.25% higher than the results from K-Means. Spectral clustering enables better speaker boundary detection and label assignment at a moderate level which indicates its superiority for speaker diarization applications.

Table 5.3 Spectral Clustering shows consistent improvements over K-means.



Audio File	Kmeans Clustering		Spectral Clustering	
	DER	Accuracy	DER	Accuracy
file1.wav	14.37	85.63	14.01	85.99
file2.wav	15.82	84.18	13.45	86.55
file3.wav	16.21	83.79	14.84	85.16
file4.wav	13.56	86.44	11.23	88.77
file5.wav	14.93	85.07	14.19	85.81
file6.wav	15.48	84.52	12.73	87.27
file7.wav	13.79	86.21	14.67	85.33
file8.wav	16.64	83.36	14.91	85.09
file9.wav	14.05	85.95	12.83	87.17
file10.wav	15.12	84.88	14.58	85.42

The results found in table 5.3 support the persistent superior performance of spectral clustering compared to K-means for most of the audio files tested. Spectral clustering leads to substantial DER reductions in file3.wav, file4.wav and file6.wav among others and specifically file4.wav experiences the greatest decrease at 2.3% DER reduction. Spectral clustering demonstrates a better capability than K-means for capturing non-linear patterns in similarity data which leads to improved speaker separation.

The DER measurements for file1.wav and file5.wav show slightly similar accuracy between K-means and spectral clustering approaches, yet spectral clustering delivers superior final diarized results in most cases. The graphical observation confirms spectral clustering methods produce superior results than K-means when these techniques operate on speaker embedding similarity data under acoustically difficult conditions that contain overlapping speech or similar speaker identities. The identified results prove spectral clustering to be a superior option than K-means clustering for speaker diarization systems that improves accuracy.

VI. Conclusion

This research presents a speaker diarization method using ECAPA-TDNN embeddings with spectral clustering, evaluated against a K-means baseline. Tested on 10 recordings from TOEFL and VoxCeleb, spectral clustering improves performance consistently, achieving an average DER of 13.74% and purity of 86.26%, outperforming K-means (14.99% DER, 85.00% accuracy). Spectral clustering models non-linear speaker similarities better than K-means, handling overlapping speech and speaker similarity more effectively. Matrix visualizations show spectral clustering captures speaker structures better, achieving Mean ARI = 0.794 and NMI = 0.846. These results validate spectral clustering as a robust, scalable, and more accurate alternative for speaker diarization.

VII. References

[1] Indu, D., Srinivas, Y., ”A Cluster-Based Speaker Diarization System Combined with Dimensionality Reduction Techniques,” *International Journal of Intelligent Systems and Applications in Engineering (IJISAE)*, vol. 12(14s), pp. 125–132, 2024.

[2] Jadhav, Ajinkya N., Dharwadkar, Nagaraj V., ”A Speaker Recognition System Using Gaussian Mixture Model, EM Algorithm and K-Means Clustering,” *International Journal of Modern Education and Computer Science (IJMECS)*, vol. 11, pp. 19–28, 2018.

[3] Ning, Huazhong, Liu, Ming, Tang, Hao, Huang, Thomas, ”A Spectral Clustering Approach to Speaker Diarization,” *Proceedings of INTERSPEECH 2006 - ICSLP*, pp. 2178–2181, 2006.

[4] Zhang, Aonan, Wang, Quan, Zhu, Zhenyao, Paisley, John, Wang, Chong, ”Fully Supervised Speaker Diarization,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2019, pp. 6301–6305, 2019.

[5] Nagrani, Arsha, Chung, Joon Son, Zisserman, Andrew, ”VoxCeleb: A Large-Scale Speaker Identification Dataset,” *Interspeech*, vol. 2017, pp. 2616–2620, 2017.

[6] Snyder, David, Garcia-Romero, Daniel, Sell, Gregory, Povey, Daniel, Khudanpur, Sanjeev, ”X-Vectors: Robust DNN Embeddings for Speaker Recognition,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2018, pp. 5329–5333, 2018.

[7] Chanchla A. Tripathi, Vishal V. Panchbhai, Lalit B. Damahe, Mahesh R. Shirole, Shruti Tiwari, Raunak Rathi, Prateek Varma, ”A review of techniques for semantic understanding of the text with term weighting,” *AIP Conf. Proc.*, vol. 3139, no. 1, p. 100006, Aug. 2024.

[8] L. Zelnik-Manor and P. Perona, “Self-tuning spectral clustering,” NIPS, 2004.

[9] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, “Speaker diarization using deep neural network embeddings,” in International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2017, pp. 4930–4934.

[10] S. H. Shum, N. Dehak, R. Dehak, and J. R. Glass, “Unsupervised methods for speaker diarization: An integrated and iterative approach,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 21, no. 10, pp. 2015–2028, 2013.

[11] G. Sell and D. Garcia-Romero, “Speaker diarization with plda i-vector scoring and unsupervised calibration,” in Spoken Language Technology Workshop (SLT), 2014 IEEE, 2014, pp. 413–417.

[12] B. Desplanques, J. Thienpondt, and K. Demuynck, “ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification,” in Proc. Interspeech, 2020, pp. 3830–3834.

[13] H. Bredin, “pyannote.metrics: a toolkit for reproducible evaluation, diagnostic, and error analysis of speaker diarization systems,” hypothesis, vol. 100, no. 60, pp. 90, 2017.

[14] NIST, “Rich transcription 2004 spring meeting recognitio evaluation plan,” http://www.nist.gov/speech/, 2004.

[15] Q. Jin and T. Schultz, “Speaker segmentation and clustering in meetings,” ICSLP, 2004.

[16] S. Zhang, H.-S. Wong, and Y. Shen, ”Generalized Adjusted Rand Indices for cluster ensembles,” Pattern Recognition, vol. 45, no. 6, pp. 2214–2226, 2012.