Enhanced Self-Supervised Speaker Diarization Framework with Conformer and Hybrid Clustering

Mala J B[1,3], Alex Raj S M[1,3], Rajeev Rajan[2,3]
[1] College of Engineering Trivandrum
[2] Government Engineering College, Idukki, India
[3]APJ Abdul Kalam Technological University, Kerala, India.

***Abstract*—In this paper, we propose a novel self-supervised speaker diarization framework built upon iterative hybrid hierarchical clustering and conformer-based deep representational learning. The proposed network is trained with x-vectors using contrastive loss derived from pseudo-labels generated in the previous step, while the clustering algorithm utilizes latent vectors from the representation network to generate the pseudo-labels. While state-of-the-art diarization systems typically use agglomerative hierarchical clustering (AHC) for all clustering iterations, our approach first applies first-integer neighbour clustering hierarchy (FINCH) to generate a reduced and refined set of initial pseudo-labels, followed by AHC. The proposed conformer-based hybrid clustering model, employing the FINCH+AHC combination, achieves a DER of 10.69% on the CallHome dataset with faster convergence, significantly outperforming the AHC-only system with a relative improvement of 8.69%. This demonstrates that effective initial learning and high-quality speaker embeddings can enhance the performance of self-supervised learning (SSL) systems.**

***Index Terms*—speaker diarization, x-vectors, self-supervised learning, conformer, hybrid clustering, FINCH**

I. Introduction

Speaker diarization, the task of identifying “who spoke when” in a multi-participant audio recording [1], has garnered significant attention in the speech community owing to its importance in the areas of automatic speech recognition [2], meeting transcriptions and analysis, audio forensics, surveillance etc [3]. However, the performance of speaker diarization systems is adversely affected by overlapping speech, brief speaker turns, speaker similarity etc. Despite these challenges, the field has made significant progress in recent years, driven by advancements in deep learning techniques.

Speaker diarization system typically consists of a voice/speech activity detection module (VAD/SAD), a speech segmentation module, and an embedding extraction module followed by a clustering module. Speaker-specific embeddings such as i-vectors [4], d-vectors [5], or x-vectors [6] are extracted from short speech segments and subsequently clustered to assign speaker-specific labels. Over the years, speaker embeddings used in diarization have evolved [7], [8], yet many state-of-the-art systems still rely on unsupervised clustering methods like agglomerative hierarchical clustering (AHC), [7], [9], [10], spectral clustering [11], Bayesian HMM [12] etc for generating diarization results.

Recently, SSL techniques have emerged as a promising approach for leveraging unlabeled data to learn more discriminative and informative representations than traditional unsupervised techniques. SSL methods focusing on joint optimization of speaker embeddings and clustering algorithms have been widely explored. Most of the previous works in SSL based audio processing focused on optimizing clustering strategies, representational neural networks, and non-clustering loss functions [13]. In SSL based speaker diarization, graph-based clustering strategies have been proposed in [14], [15] for grouping speaker embeddings. Prachi *et al* utilizes path integral clustering in [16], [17] while [18] employs AHC to cluster the embeddings. In [19], a trained encoder model is used to self-generate pseudo-labels. Both triplet loss and contrastive loss [20], [21] are incorporated as non-clustering loss functions within an SSL framework. Furthermore, [22] explores the use of dynamic triplet loss and multinomial loss for improved representation learning. However, limited works have investigated the significance of initial learning and its impact on the overall self-supervised learning process.

In this work, we propose an SSL system based on deep representational hybrid clustering framework designed to improve initial learning and quality of speaker embeddings. This work is inspired by Prachi *et al* [18] where AHC is used for deriving the pseudo-labels in all the iterations of an SSL framework. In our approach, to improve initial learning, we employ a combination of FINCH [23] and AHC and the proposed approach is applied on the refined speaker embeddings generated by the conformer architecture. Clustering the refined pseudo-labels constitutes the forward pass, while updating the representation framework parameters corresponds to the backward pass in the recurrent framework. With this enhanced FINCH+AHC based recurrent framework, we aim to achieve strong speaker embeddings and generate distinct, accurate speaker clusters.

II. System Description

This section describes the proposed self-supervised speaker diarization framework built on FINCH+AHC hybrid clustering and the baseline system based on single clustering strategy.

A. Proposed Approach

The proposed approach using FINCH+AHC hybrid clustering is shown in Fig 1. X-vectors generated for 0.75s audio segments are given as input to the self-supervised recurrent framework. X-vector extraction is given in more detail in section III B. Algorithm 1 outlines the implementation of the

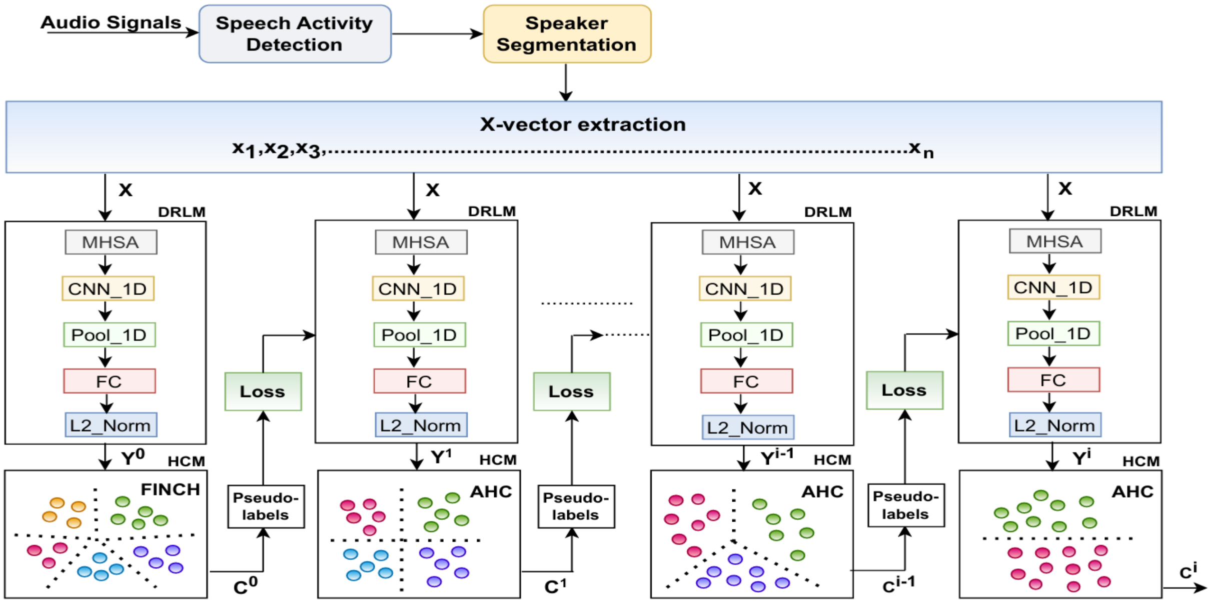


Fig. 1: Proposed self-supervised speaker diarization architecture based on conformer embeddings and FINCH+AHC.

proposed approach. The recurrent framework consists of two modules - deep representation learning module (DRLM) and hierarchical clustering module (HCM).

Deep representation learning module (DRLM): DRLM (DNN or Conformer) maps raw x-vectors X to more refined latent x-vectors Y using a non-linear mapping f and learnable parameters θ, f$_\theta$: X $\rightarrow$ Y. Thus, x-vectors from higher-dimensional (D) input feature space is mapped to a lower-dimensional (d) cluster-friendly latent embedding space.

Hierarchical clustering module (HCM): Here, the latent vectors are grouped using a clustering algorithm to generate pseudo-labels, which are fed as supervised labels for the next iteration. Our contribution is the introduction of FINCH algorithm for the generation of initial pseudo-labels. Y^0 from DRLM is given as input to the initial block in HCM which generates the initial set of cluster labels C^0. FINCH generates a more refined set of cluster labels and is based on first-neighbour relations. The algorithm generates hierarchical partitions with high purity. Hence, FINCH generates less number of initial clusters than AHC as the algorithm in its earlier stages emphasizes on nearest neighbours and local structure, whereas AHC's purity depends more on the linkage and affinity strategies chosen for merging clusters. It also possesses low computational overhead and is extremely fast.

FINCH is well-suited for speaker diarization tasks for three reasons: (i) it does not require apriori knowledge on the number of clusters; (ii) it produces clusters with very high purity; and (iii) it is both fast and scalable. FINCH clustering starts with creating an adjacency link matrix for each data sample.

$$A(i,j) = \begin{cases} 1 & \text{if } j = \kappa_i^1 \text{ or } \kappa_j^1 = i \text{ or } \kappa_i^1 = \kappa_j^1 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where κ_i^1 denotes the first neighbour of sample i. The adjacency matrix links each sample i to its first neighbour via j =κ_i^1, enforces symmetry through κ_j^1 = i, and links samples (i, j) that share a common neighbour with κ_i^1 = κ_j^1. Equation 1 combines both 1-nearest neighbour (1-nn) and shared nearest neighbour (SNN) graphs. Clustering is performed by identifying the connected components in the adjacency matrix.

In all the subsequent iterations, AHC is used as the clustering algorithm to generate the pseudo-labels. The iteration continues until the stopping criterion is satisfied: the number of unique clusters in the current step, C^i equals the target number of speakers N_c^* and the ratio of current loss to initial loss always remains below a specified threshold. Neural networks are trained using triplet or contrastive loss, computed based on the pseudo-labels generated in the previous iteration. Contrastive loss is calculated using

$$L = (1-y)d_W^2 + y(\max(0, m - d_W))^2 \quad (2)$$

where L is the contrastive loss, y is the label (0 for similar pairs, 1 for dissimilar pairs), $d_W = \|\mathbf{a} - \mathbf{p}\|_2$ is the Euclidean distance (L_2 norm) between the anchor ($\mathbf{a}$) and the paired embedding ($\mathbf{p}$), m is the margin enforcing separation for dissimilar pairs.

Integrating FINCH with AHC helps to utilize the strengths of both algorithms. Introduction of FINCH reduces the overall complexity as the initial clustering reduces the number of clusters that AHC needs to process. Thus this hybrid clustering framework can ensure better clustering performance compared to using either algorithm alone.

B. Baseline System

In this work, deep self-supervised hierarchical clustering discussed in [18] is used as the baseline system. We conducted experiments with x-vectors of 512D using four layer DNN architecture. Low-dimensional latent x-vectors, Y, resulting from DRLM module are fed as input to the HCM module. HCM uses AHC algorithm in all the iterations and generates a set of pseudo-labels, C. These pseudo-labels serve as supervisory signals for the next iteration, where the DNN is trained by minimizing the triplet loss computed from the previous pseudo-labels. The process of recurrently updating the DNN parameters and pseudo-labels continues until the stopping criterion given in section II A is satisfied. The AHC follows a bottom-up clustering approach, where the metric affinity measure defines the pairwise similarity between clusters, and the similarity measure determines the distance between x-vectors. AHC merges two clusters C^a and C^b based on the affinity measure $\mathcal{A}$ between two clusters. Mathematically, it is represented as:

$$\{\mathcal{C}_a, \mathcal{C}_b\} = \underset{\mathcal{C}_i, \mathcal{C}_j \in \mathcal{C}, i \neq j}{\operatorname{argmax}} \mathcal{A}\left(\mathcal{C}_i, \mathcal{C}_j\right) \tag{3}$$

where $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_S\}$, S- number of segments.

Algorithm 1: SSL framework using FINCH+AHC

```
Input: X= $\{x_1,x_2,.......x_S\} \in \mathcal{R}^{S \times D}$: x-vectors with
  dimension D extracted from S audio segments
$N_c^*$ = target number of clusters (speaker labels)
threshold $\in (0, 1]$
Output:
$C^i$ = Final cluster labels for each segment
$\theta$ = NN parameters.
Procedure:
Initial Clustering:
i=0
$\theta^0$ : initial NN parameters for PCA reduced x-vectors
$Y^0$ = $\{\mathrm{y}_1^0, y_2^0, \ldots, y_S^0\} \in \mathcal{R}^{S \times d}$, initial low-dimensional
  latent vectors,  where  d $\ll D$.
$C^0$ = $\{c_1^0,c_2^0,.......c_S^0\} \leftarrow FINCH(\mathrm{Y}^0)$
$L^0$ = initial contrastive loss computed using $C^0$
    1) Repeat:
        a) $i \leftarrow i + 1$
        b) Update NN parameters: $\theta^i \xleftarrow{L^{i-1}} \theta^{i-1}$
        c) $Y^i \leftarrow \theta^i(\mathrm{X})$
        d) $C^i$ = $\{c_1^i,c_2^i,.......c_S^i\} \leftarrow AHC(\mathrm{Y}^i)$
        e) Compute contrastive loss ($L^i$) using $C^i$
    2) Until: number of distinct clusters in $C^i$ equals $N_c^*$
       and ($L^i/L^0$) <= threshold
```

III. Performance evaluation

A. Dataset

CallHome (CH) English corpus [24] is used for the experiments. The corpus includes 120 unscripted telephone conversations between native English speakers, with each recorded conversation lasting 30 minutes. A subset of CH dataset is chosen for the experiments. The proposed approach is applied independently on each input audio. Voiced audio samples are split into 0.75s duration segments.

TABLE I: Conformer architecture for input shape (**x**,512), **x** - batch size.

No	Layer	Output Shape
1	Input Layer	(x,512)
2	Multi Head Self Attention	(x,1,512)
3	Conv1D	(x,1,512)
4	GlobalAveragePooling1D	(x,512)
5	Dense	(x,30)

B. Experimental Setup

The experimental setup involves following steps:

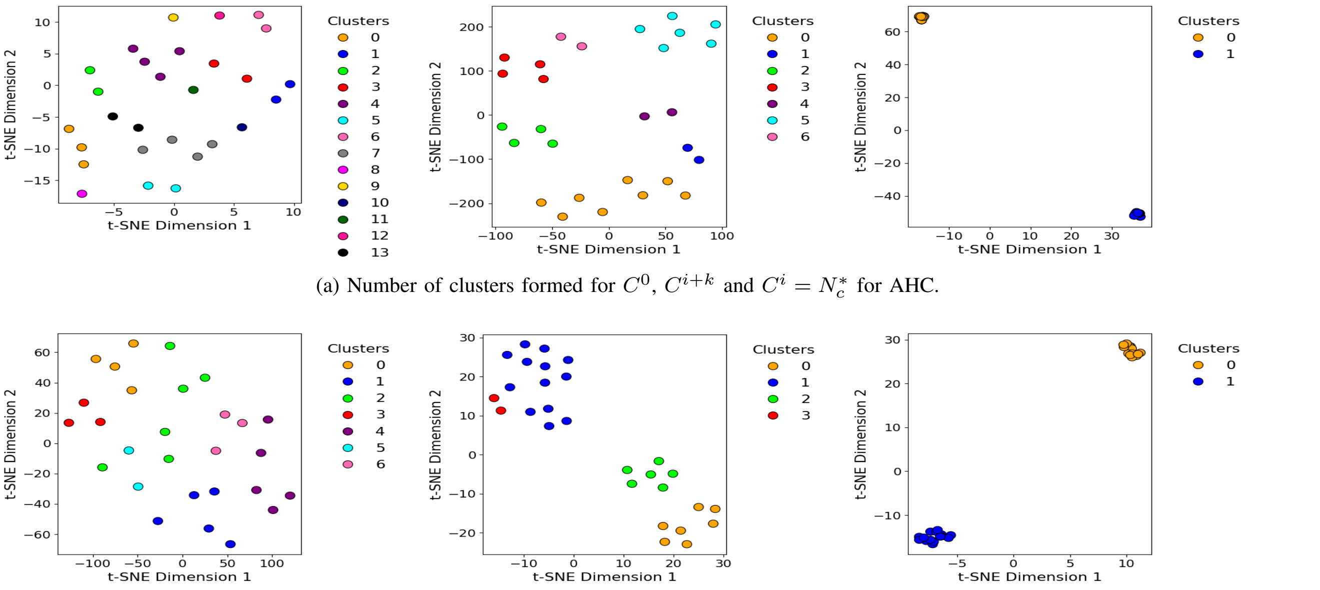
`Separation of voice segments:` Speech activity detection is first applied to the input audio recordings to remove un-voiced segments. Then, overlap detection is conducted to discard segments with speakers' overlap. Resulting audio samples are splitted into 0.75s speech segments.

`X-vector extraction:` X-vectors extraction is done using a pre-trained model from SpeechBrain [25]. Model utilizes a TDNN architecture with statistical pooling and is trained on the VoxCeleb1 and VoxCeleb2 datasets sampled at 16kHz, containing 7205 speakers, using categorical cross-entropy as the loss function. For each 0.75s segment of the audio, x-vector embeddings of 512 dimensions are extracted.

`SSL with AHC:` For single clustering strategy based on AHC, we have conducted experiments with DNN and conformer. Baseline DNN architecture has four layers (512, 256, 64 and 30) with L_2 normalization and ReLU activation at each layer, trained using triplet loss. The first layer is initialized with whitening transform, and the final layer with principal components obtained from PCA for better convergence. Next to evaluate the impact of high-quality embeddings on the recurrent network, DNN is replaced by conformer. Table I details the conformer architecture used. Multi-head self-attention (MHSA) uses eight attention heads with 64 as the key dimension. The use of MHSA as well as 1D convolutional layer enables the conformer to capture the long-term as well as local dependencies in input audio. Finally, L_2 normalization is applied to generate well defined high-quality latent vectors. For each iteration of the recurrent framework, neural networks are trained using pseudo-labels and triplet/contrastive loss calculated from previous step. The value of margin for contrastive loss is chosen as one. The resulting latent x-vectors are in turn, given to the AHC algorithm until the network satisfies the stopping criterion.

`SSL with FINCH+AHC:` Proposed **`Model-1`** (DNN) and Proposed **`Model-2`** (Conformer) are implemented with FINCH+AHC hybrid clustering. Both models follow the same architecture as in SSL with AHC. Y^0, initial latent x-vectors, are given as input to the FINCH algorithm. FINCH generates an initial set of high-purity pseudo-labels, C^0, with cosine as distance measure. We have used clusters from the first partition as C^0. The loss is computed using C^0, and gradients are back propagated to update the model's weights. Experiments are performed using both triplet and contrastive loss.

For all the experiments, Adam is chosen as the optimizer with a learning rate of $1e^{-3}$ and a patience of ten for early stopping. The stopping criterion enables the network to converge to a 2-speaker system while simultaneously ensuring



(a) Number of clusters formed for C^0, C^{i+k} and $C^i = N_c^*$ for AHC.

(b) Number of clusters formed for C^0, C^{i+k} and $C^i = N_c^*$ for FINCH+AHC hybrid clustering.

Fig. 2: Comparison of cluster formations using AHC and Proposed Model 2 (FINCH+AHC) for threshold = 0.5 using conformer architecture.

TABLE II: DER for proposed approaches using contrastive loss.

No	System	Approach	DER%
1	D.Snyder et al [26]	x-vector+AHC	26.07
2	P. Singh et al (DNN) [18] (Baseline)	DNN+AHC	19.38
3	P. Singh et al (Conformer)	Conformer+AHC	13.61
4	Proposed **Model-1**	DNN+FINCH+AHC	16.61
5	Proposed **Model-2**	Conformer+FINCH+AHC	**10.69**

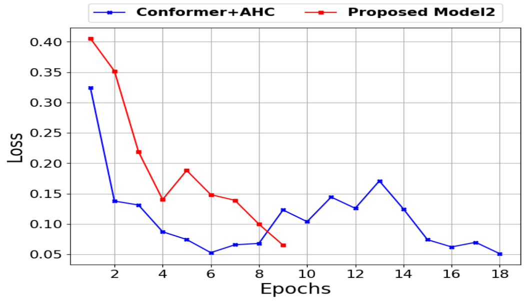


Fig. 3: Loss convergence of Conformer+AHC and Proposed Model2 (FINCH+AHC) using contrastive loss for threshold = 0.5.

that the loss in each iteration remains below the threshold. The value of threshold is empirically chosen as 0.5 based on ablation studies. All experiments are implemented using Keras-Tensorflow and the execution environment runs on Google Colab and utilizes NVIDIA T4 Tensor Core GPUs.

IV. Result Analysis

Diarization error rate (DER) [27] is used as the performance metric for evaluation. Table II shows DER obtained for proposed approaches using contrastive loss. Unsupervised learning using AHC gives a DER of 26.07% with cosine as the cluster similarity measure and average linkage as the affinity measure. **Model-1** reports a DER of 16.61%. Thus the introduction of FINCH has given a relative improvement of 2.77% than the DNN based baseline system (19.38%).

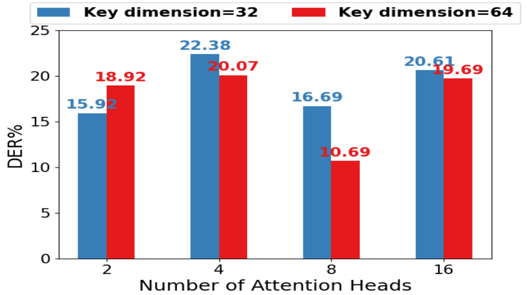


Fig. 4: DER for Proposed Model 2 for various number of attention heads and key dimension in the conformer architecture.

This improvement attributes to the ability of FINCH in generating high purity clusters thereby reducing the initial number of clusters, thus facilitating faster convergence and improved clustering. Conformer+AHC achieved a DER of 13.61%, demonstrating that high-quality embeddings can improve DER, due to the ability of conformer to capture and learn long-term dependencies and local features of speech. Experiments with **Model-2** further reduced DER to 10.69%, a remarkable relative improvement of 8.69% than the baseline system. Fig 2 illustrates the effectiveness of using hybrid FINCH+AHC clustering. It is worth noting that the initial number of clusters has reduced from 14 to 7 for **Model-2**, a notable improvement over conformer+AHC. Fig 3 shows the loss convergence for conformer+AHC and **Model-2**. With fewer initial clusters, **Model-2** achieves faster convergence in fewer epochs, leading to a sharper loss reduction and faster stabilization compared to the conformer+AHC approach. DER% obtained for various MHSA configurations in the conformer architecture of **Model-2** is shown in Fig 4. Lowest DER is reported when number of attention heads is 8 and key dimension is 64, efficiently dividing the 512D input,

TABLE III: DER for various losses.

No	System	Loss	DER%
1	P.Singh et al (DNN) [18] (Baseline)	Triplet	19.38
2	P.Singh et al (Conformer)	Triplet	14.15
3	Proposed `Model-1`	Triplet	18.00
4	Proposed `Model-2`	Triplet	12.38
5	P.Singh et al (Conformer)	Contrastive	13.61
6	Proposed `Model-1`	Contrastive	16.61
7	Proposed `Model-2`	Contrastive	**10.69**

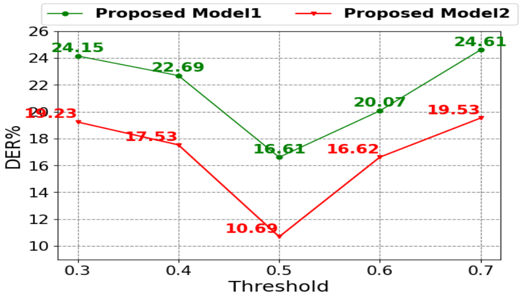


Fig. 5: DER at different thresholds for proposed models using contrastive loss.

with each head processing a 64D subspace, ensuring efficient feature extraction. Table III summarizes the experiments on non-clustering loss. **`Model-1`** and **`Model-2`** achieve lowest DER of 16.61% and 10.69%, respectively, for contrastive loss. This improvement shows that contrastive loss has efficiently optimized speaker embeddings than triplet loss, thus enhancing speaker separation. Fig 5 shows that **`Model-1`** and **`Model-2`** achieves the lowest DER at a threshold of 0.5.

V. Conclusion

This work proposed a novel self-supervised deep hybrid clustering model for speaker diarization, emphasizing the importance of initial learning and the necessity of high-quality speaker embeddings. We integrated FINCH and AHC with conformer-based embeddings to improve SSL performance. While the baseline system reports a DER of 19.38% on CallHome dataset, our contrastive loss based proposed approaches achieves a DER of 16.61% (**`Model-1`**) and 10.69% (**`Model-2`**) respectively, a notable improvement of 8.69% over the baseline, with faster convergence. From the metrics, it is clear that the proposed approaches can significantly enhance the performance of speaker diarization systems.

References

[1] T. J. Park, N. Kanda, D. Dimitriadis, K. J. Han, S. Watanabe, and S. Narayanan, "A review of speaker diarization: Recent advances with deep learning," *Computer Speech & Language*, vol. 72, p. 101317, 2022.

[2] K. Manohar and R. Rajan, "Improving speech recognition systems for the morphologically complex malayalam language using subword tokens for language modeling," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2023, no. 1, p. 47, 2023.

[3] L. Serafini, S. Cornell, G. Morrone, E. Zovato, A. Brutti, and S. Squartini, "An experimental review of speaker diarization methods with application to two-speaker conversational telephone speech recordings," *Computer Speech & Language*, vol. 82, p. 101534, 2023.

[4] 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.

[5] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, "Speaker diarization with LSTM," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 5239–5243.

[6] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 5329–5333.

[7] H. Aronowitz, W. Zhu, M. Suzuki, G. Kurata, and R. Hoory, "New advances in speaker diarization." in *Interspeech*, 2020, pp. 279–283.

[8] B. KC, R. Rajan *et al.*, "Attention-augmented x-vectors for the evaluation of mimicked speech using sparse autoencoder-LSTM framework," in *Proc. Interspeech 2024*, 2024, pp. 3804–3808.

[9] W. H. Day and H. Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods," *Journal of classification*, vol. 1, pp. 7–24, 1984.

[10] J. Mala, S. Alex Raj, and R. Rajan, "X-vector-based speaker diarization using Bi-LSTM and interim voting-driven post-processing," in *International Conference on Text, Speech, and Dialogue*. Springer, 2024, pp. 161–173.

[11] T. J. Park, K. J. Han, M. Kumar, and S. Narayanan, "Auto-tuning spectral clustering for speaker diarization using normalized maximum eigengap," *IEEE Signal Processing Letters*, vol. 27, pp. 381–385, 2019.

[12] F. Landini, J. Profant, M. Diez, and L. Burget, "Bayesian HMM clustering of x-vector sequences (vbx) in speaker diarization: theory, implementation and analysis on standard tasks," *Computer Speech & Language*, vol. 71, p. 101254, 2022.

[13] S. Liu, A. Mallol-Ragolta, E. Parada-Cabaleiro, K. Qian, X. Jing, A. Kathan, B. Hu, and B. W. Schuller, "Audio self-supervised learning: A survey," *Patterns*, vol. 3, no. 12, 2022.

[14] P. Singh, A. Kaul, and S. Ganapathy, "Supervised hierarchical clustering using graph neural networks for speaker diarization," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2023, pp. 1–5.

[15] Y. Wei, H. Guo, Z. Ge, and Z. Yang, "Graph attention-based deep embedded clustering for speaker diarization," *Speech Communication*, vol. 155, p. 102991, 2023.

[16] P. Singh and S. Ganapathy, "Self-supervised representation learning with path integral clustering for speaker diarization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1639–1649, 2021.

[17] W. Zhang, D. Zhao, and X. Wang, "Agglomerative clustering via maximum incremental path integral," *Pattern Recognition*, vol. 46, no. 11, pp. 3056–3065, 2013.

[18] P. Singh and S. Ganapathy, "Deep self-supervised hierarchical clustering for speaker diarization," in *Interspeech*, 2020, pp. 294–298.

[19] Y. Dissen, F. Kreuk, and J. Keshet, "Self-supervised speaker diarization," in *Interspeech*, 2022, pp. 4013–4017.

[20] A. Saeed, D. Grangier, and N. Zeghidour, "Contrastive learning of general-purpose audio representations," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021, pp. 3875–3879.

[21] V. Sharma, M. Tapaswi, M. S. Sarfraz, and R. Stiefelhagen, "Clustering based contrastive learning for improving face representations," in *15th IEEE International Conference on Automatic Face and Gesture Recognition*, 2020, pp. 109–116.

[22] Y. Ding, Y. Xu, S.-X. Zhang, Y. Cong, and L. Wang, "Self-supervised learning for audio-visual speaker diarization," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 4367–4371.

[23] S. Sarfraz, V. Sharma, and R. Stiefelhagen, "Efficient parameter-free clustering using first neighbor relations," in *IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8934–8943.

[24] Linguistic Data Consortium. CABank CallHome English Corpus. TalkBank. Https://ca.talkbank.org/access/CallHome/eng.html, last accessed 2023/05/10.

[25] M. Ravanelli, T. Parcollet, A. Moumen, S. de Langen, C. Subakan, P. Plantinga, Y. Wang, P. Mousavi, L. Della Libera, A. Ploujnikov *et al.*, "Open-source conversational AI with SpeechBrain 1.0," *Journal of Machine Learning Research*, vol. 25, no. 333, pp. 1–11, 2024.

[26] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *IEEE International conference on acoustics, speech and signal processing*, 2019, pp. 5796–5800.

[27] H. Bredin, "pyannote.audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe," in *Interspeech*, 2023, pp. 1983–1987.