A Comprehensive Literature Review on Speaker Diarization

Sudipto Das Suvro

***Abstract*—Speaker diarization (SD) refers to determining *"who spoke when"* in an audio recording. This survey provides a comprehensive overview of the field, beginning with problem formulation and classical approaches based on Bayesian segmentation, MFCC/i-vector embeddings, and agglomerative clustering followed by major deep learning advances, including x-vectors, ECAPA-TDNN, self-supervised Transformer embeddings, pyannote.audio pipelines, and end-to-end neural diarization (EEND). Widely used datasets and standard evaluation metrics such as DER, JER, and CDER are summarized to support benchmarking and reproducible research. Moreover, recent research trends are also added which highlight robust overlap-handling methods, self-supervised representations, audio–visual fusion, real-time diarization, and large speech models such as Whisper. Lastly, open challenges—including domain adaptation, multilingual robustness, low-latency processing, and limited-labeled-data scenarios—are discussed alongside a comparative analysis of state-of-the-art systems. Overall, this review synthesizes the developments that have shaped modern diarization and outlines promising directions for future research. Implementation code for the experimental pipelines are available at: *https://github.com/sudiptosuvro/speaker-dia*.**

***Index Terms*—Speaker diarization, PyAnnote, ECAPA-TDNN, x-vectors, EEND, diarization error rate, clustering, speech processing.**

I. Introduction

Recorded audio often contains speech from multiple people in conversation. It is useful to label such signals with speaker turns, noting when each speaker is talking and identifying each speaker. [1] SD is the process of automatically identifying and segmenting an audio recording into distinct speech segments, where each segment corresponds to a particular speaker. In simpler words, the goal is to answer the question: *who spoke when?* [2]. It provides speaker segmentation labels over time, forming the basis for downstream tasks such as automatic speech recognition (ASR), audio indexing, spoken dialogue understanding, and multimodal transcription.

The process involves the audio signal analysis to detect changes in speaker identity and then grouping segments that belong to the same speaker. The diarization task is inherently challenging due to speaker overlap, environmental noise, reverberation, varying number of speakers, and domain mismatch across datasets. An example of speaker diarization is shown in Fig. 1, where the audio is partitioned into speaker-labeled segments.

Graduate Student, Department of Electrical and Computer Engineering, The University of Alabama in Huntsville, Huntsville, AL, USA.
Email: sudipto.suvro@uah.edu

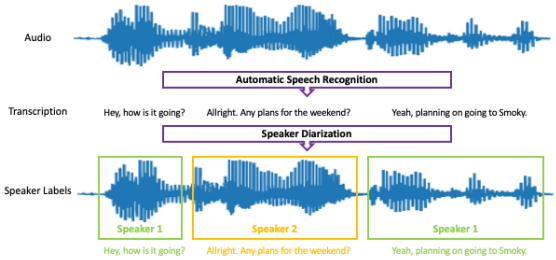

Fig. 1: Example of a speaker diarization.

Traditional SD systems follow a pipeline consisting of voice activity detection (VAD), segmentation, speaker embedding extraction, and clustering. Following advances in deep representation learning, diarization has shifted toward neural embeddings and end-to-end frameworks [3]. This review consolidates classical principles, deep learning methodologies, datasets, metrics, and recent research trends that define modern diarization.

II. Problem Definition

Let $x(t)$ denote a continuous-time speech waveform observed over the interval $t \in [0, T]$. The speaker diarization task aims to determine *"who spoke when"* by assigning a speaker identity to each time instant (or frame) of the signal [4], [5]. Let $\mathcal{S} = \{1, 2, \ldots, K\}$ be the set of speakers appearing in the recording, where K may be known or unknown. The objective is to estimate a time–speaker activity function:

$$y_k(t) \in \{0, 1\}, \qquad k \in \mathcal{S},$$

where $y_k(t) = 1$ indicates that speaker k is active at time t. Discretizing $x(t)$ into frames yields a binary activity matrix:

$$\mathbf{Y} = [y_{t,k}] \in \{0, 1\}^{T \times K},$$

where $y_{t,k}$ expresses whether speaker k is active in frame t. The diarization system must estimate $\widehat{\mathbf{Y}}$ using only the observed waveform $x(t)$.

Alternatively, the output can be expressed as a sequence of labeled segments:

$$\widehat{\mathcal{D}} = \left\{ \left(t^{(i)}_{\text{start}}, t^{(i)}_{\text{end}}, s^{(i)} \right) \right\}_{i=1}^{N},$$

where each segment i is assigned a speaker label $s^{(i)} \in \mathcal{S}$. Classical pipelines decompose the problem into VAD, segmentation, embedding extraction, and clustering, whereas end-to-end neural diarization (EEND) approaches learn a direct mapping:

$$f_\theta : x(t) \rightarrow \mathbf{Y},$$

without relying on explicit segmentation or clustering steps.

III. Classical Diarization Approaches

A. Voice Activity Detection

Earlier systems relied on:

- *Energy-based VAD*: Early systems used simple thresholds on short-term energy or zero-crossing rate to detect speech regions. This method assumes speech generally has higher energy than background noise. By dividing signals into short frames (e.g., 20-30ms), energy is computed to classify it as speech/non-speech [6]. This is highly sensitive to background noise, making it unreliable for real conversational data.
- *GMM-HMM VAD*: A Gaussian Mixture Model provides the acoustic feature distribution of speech and noise, while the Hidden Markov Model enforces temporal continuity to avoid rapid fluctuations between speech/non-speech states [7]. This statistical distribution of speech vs. non-speech and probabilistic structure makes the system more robust under moderately noisy or reverberant conditions.
- *WebRTC VAD*: The WebRTC project, initiated by Google is an open-source real-time VAD designed for handling audio in real-time communication, including Voice over IP applications [8]. It splits audio into multiple frequency sub-bands and evaluates speech likelihood using spectral features, perceptual weighting, and adaptive noise modeling, making it a practical standard in many commercial systems.
- *SVM-based VAD*: Support Vector Machines were applied for supervised VAD using spectral features which offers improved performance over GMMs under variable noise conditions. It formulates speech detection as a binary classification problem, learning an optimal hyperplane between speech and non-speech feature vectors [9]. Although SVMs maximize the margin between classes, making the system better to unseen noise types, they depend on the availability and quality of labeled training data, which limits their scalability in real-world diarization scenarios.

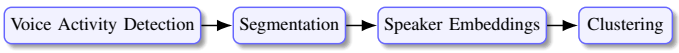

Fig. 2: Classical diarization pipeline

B. Speaker Segmentation

Speaker segmentation identifies points where the speaker identity changes. Before deep learning, segmentation relied on statistical change-detection measures that compared acoustic distributions between adjacent time windows. Pre-deep learning methods include:

- *Bayesian Information Criterion (BIC)*: BIC compares two hypotheses—whether adjacent audio windows are from one or two speakers—by modeling each region with Gaussians [10]. It calculates a penalized likelihood score that balances model fit with model complexity. When the two-Gaussian model provides a significantly better fit, a speaker change is declared. This makes BIC robust and widely used, but also computationally heavy and sensitive to window size.
- *Generalized Likelihood Ratio (GLR)*: GLR compares likelihoods of single vs. two-Gaussian models to detect acoustic change points. GLR is computationally simpler than BIC but more sensitive to threshold selection. This system measures how likely it is that two adjacent segments originate from different statistical distributions [11]. If the ratio exceeds a certain threshold, a boundary is declared. Its lower computational cost enabled real-time segmentation, but performance could degrade quickly in noisy or mismatched conditions.
- *KL divergence segmentation*: Uses the Kullback–Leibler divergence to measure statistical dissimilarity between adjacent windows. Particularly effective for speaker-dependent acoustic differences. It quantifies how one probability distribution diverges from another, making it suitable for detecting gradual acoustic changes between speakers [12] which does not require explicit model estimation like BIC, but it is sensitive to feature distribution estimation.

C. Embeddings: From MFCCs to i-Vectors

Speaker embeddings provide compact numerical representations of speaker characteristics. Classical diarization systems evolved from simple spectral features (MFCCs) to probabilistic latent-vector representations (i-vectors), enabling more robust speaker modeling before the deep-learning era. The evolution progressed as:

- *MFCC + GMM-UBM*: MFCCs capture spectral envelope characteristics of speech, while the Universal Background Model (UBM) models general speech distribution using a GMM. The GMM-UBM framework computes the likelihood of observed MFCC frames under a speaker-independent model, then uses MAP adaptation to derive speaker-dependent parameters [13]. These adapted models serve as early embeddings, enabling distance-based speaker comparison.
- *Joint Factor Analysis (JFA)*: JFA models each speaker's feature distribution in a high-dimensional supervector space derived from UBM parameters. By explicitly isolating channel factors, it reduces false boundaries caused by background conditions or microphone changes [14]. It became a key stepping toward the total-variability modeling approach used by i-vectors.
- *i-vectors with PLDA scoring*: The i-vector framework compresses each speech segment into a low-dimensional vector in the "total variability space," capturing both speaker and session characteristics. PLDA (Probabilistic Linear Discriminant Analysis) is then used to compute similarity scores between speakers [15]. Unlike JFA, i-vectors assume a single, unified variability space, enabling efficient extraction and scalability to large

datasets. PLDA further models between-speaker and within-speaker variability, making i-vectors the dominant approach in classical diarization and speaker verification. This remained state-of-the-art until x-vectors and deep learning took over.

D. Clustering

Clustering assigns speech segments to speaker identities in an unsupervised manner. Before deep learning, clustering formed the core of diarization, grouping embeddings or acoustic features into coherent speaker clusters. The effectiveness of clustering directly determined diarization accuracy. The classical diarization pipeline is illustrated in Fig. 2, which sequentially applies VAD, segmentation, speaker embedding extraction, and clustering. Common clustering techniques:

- *Agglomerative Hierarchical Clustering (AHC)*: AHC iteratively merges the most similar clusters based on a distance metric (e.g., cosine, PLDA, or BIC) [16]. It naturally produces a hierarchical tree (dendrogram), allowing systems to cut the tree at an optimal threshold to determine the number of speakers. Its flexibility in choosing similarity measures made it adaptable to various embedding types. However, AHC struggles in long recordings with overlapping speakers or quickly changing speech patterns.
- *Spectral clustering*: Spectral clustering converts similarity matrices into graphs and uses eigenvectors of the graph Laplacian to identify well-separated speaker groups [17]. This method captures the global structure of the data, making it effective when clusters are non-linearly separable or when speaker boundaries are unclear. By leveraging the eigen-space representation, spectral clustering is more robust to noisy pairwise similarity estimates than AHC.
- *Variational Bayes HMM*: A Variational Bayes Hidden Markov Model (VB-HMM) models speaker transitions over time by assigning each time frame to a hidden speaker state and uses variational inference to estimate state probabilities [18]. This temporal smoothing prevents rapid speaker label switching and helps maintain speaker consistency across segments.
- *K-means*: K-means partitions embeddings into K clusters by minimizing within-cluster distance, typically using cosine similarity or Euclidean distance [19]. k-means requires the number of speakers to be known in advance and performs poorly when clusters vary in size or shape.

IV. Deep Learning Approaches

A. d-Vectors and x-Vectors

Deep speaker embeddings marked a major shift from statistical approaches such as i-vectors. The first deep embeddings, known as d-vectors, were introduced by Google using deep LSTM networks to map short speech segments into a discriminative speaker space. Although effective, d-vectors were sensitive to phonetic variability and required relatively long speech segments to achieve stable performance.

A more robust alternative, x-vectors, was proposed by Snyder et al [20]. X-vectors employ a Time-Delay Neural Network (TDNN) architecture, followed by a statistics pooling layer that aggregates frame-level features over an entire speech segment. This produces a fixed-dimensional embedding independent of segment length and enables more reliable clustering in diarization pipelines. When combined with PLDA scoring and hierarchical clustering, x-vectors became the dominant approach in pre–ECAPA-TDNN diarization systems.

[21] proposed a diarization system based on x-vectors, PLDA, and AHC as a front-end for a speaker recognition system. The system first trains an x-vector TDNN model on heavily augmented VoxCeleb data and extracts x-vector embeddings from short overlapping segments of each recording. These embeddings are scored using PLDA and clustered with AHC to produce speaker labels for the diarization stage. The diarized segments are then used to extract speaker-specific x-vectors, which are compared using a PLDA backend for final speaker verification decisions. The authors also propose an alternative clustering scheme that removes the need for a domain-sensitive AHC threshold, improving robustness across datasets.

B. ECAPA-TDNN

ECAPA-TDNN (Emphasized Channel Attention, Propagation, and Aggregation) is a recent advancement over the x-vector architecture that significantly improves speaker embedding quality. Proposed by [22], ECAPA-TDNN incorporates several architectural enhancements to strengthen channel interaction modeling, temporal receptive fields, and embedding robustness. The ECAPA-TDNN + VBx system is outlined in Fig. 3, combining neural embeddings with VBx-based clustering and HMM resegmentation for refined diarization output.

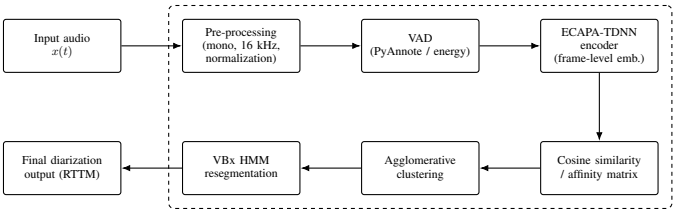

Fig. 3: ECAPA-TDNN + VBx diarization pipeline.

First, ECAPA-TDNN adopts *Res2Net* multi-scale residual blocks to increase the effective receptive field while preserving computational efficiency. These blocks enable fine-grained multi-scale feature extraction, making the model more robust to rapid spectral variations in speech. Second, a *channel-dependent attentive statistics pooling* mechanism is used in place of conventional mean–variance pooling. This module learns importance weights for different channels, enabling the network to emphasize speaker-informative frequency bands. Third, ECAPA-TDNN integrates *Squeeze-and-Excitation (SE)*

layers, which recalibrate channel responses and improve discriminability, particularly under noisy or overlapping conditions.

C. Pyannote Audio:

Pyannote.audio is one of the most influential modern toolkits for speaker diarization, providing end-to-end neural models for voice activity detection (VAD), segmentation, speaker embedding extraction, overlap detection, and clustering. Originally introduced by Bredin et al. [5], the framework has evolved into a unified diarization library with state-of-the-art performance across AMI, DIHARD, VoxConverse, and ICSI benchmarks. Its popularity stems from its modular design, strong pretrained models, and consistent reproducibility—making it widely used in research, industry, meeting transcription systems, and hybrid ASR pipelines.

System Architecture: Pyannote follows a pipeline-based neural architecture in which each stage is implemented as a learnable model:

- *Neural VAD:* A powerful waveform-based model trained to detect speech regions with high temporal resolution [23].
- *Segmentation:* A frame-level speaker-change detector using CRF layering or transformer-based architectures.
- *Speaker Embeddings:* CNN-, TDNN-, or ECAPA-TDNN–based embeddings trained using large-scale contrastive or metric-learning objectives [22].
- *Overlap Detection:* Neural models predicting time regions where multiple speakers are active, a crucial step for reducing diarization error in multi-speaker settings [24].
- *Clustering Backend:* Hierarchical clustering, spectral clustering, or Bayesian HMM/VBx-based clustering for final speaker assignment.

Integration with Pretrained Models: Starting with version 3.0, pyannote.audio integrates pretrained HuggingFace Transformers models [25]. These models offer state-of-the-art accuracy with minimal fine-tuning and have become widely adopted due to their generalization to unseen speakers and domains.

Pipeline Workflow: To increase the number of positive training samples for overlapped speech detection, pyannote.audio creates artificial samples by summing two random audio chunks [26]. A typical pyannote diarization pipeline includes:

1) VAD to detect speech regions.
2) Segmentation to mark speaker turns.
3) Neural speaker embeddings extraction.
4) Clustering.
5) Overlap handling.

Fig. 4 presents the PyAnnote diarization pipeline, integrating neural VAD, segmentation, embedding extraction, and clustering within a unified end-to-end framework. This

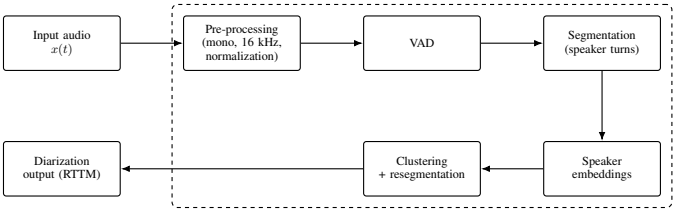

Fig. 4: PyAnnote speaker diarization pipeline.

modular workflow makes pyannote suitable for meetings, broadcast audio, telephone conversations, and multi-speaker podcasts. Key advantages include reproducible pretrained models, excellent overlap handling, support for variable-length inputs, and strong performance across domains. However, it has Limited performance on highly noisy or far-field recordings and high computational cost for long audio files. Overall, pyannote.audio represents a practical, high-performance implementation of modern diarization research and serves as a bridge between academic state-of-the-art and production-level speech processing systems.

Proposed by [27], the system first applies the *pyannote v2.1* diarization pipeline, which performs local speaker segmentation using a 5-second sliding window, extracts neural speaker embeddings, and groups them using global agglomerative clustering. These diarized segments are then processed with a pre-trained *ECAPA-TDNN* model, which generates stronger speaker embeddings using attention, SE-Res2 blocks, and multi-layer feature aggregation. Extensive real-time data augmentation (noise, reverberation, dropout, speed perturbation) is used to improve embedding robustness. Finally, clustering and spectral methods are used to assign global speaker labels, achieving improved diarization accuracy on the AMI meeting dataset.

TABLE I: Comparison of Modern Diarization Models

Method	Strength	Weakness
x-vector + AHC	Reliable, simple, well-studied	Poor overlap handling
ECAPA-TDNN	SOTA embeddings; robust short segments	High computation; GPU required
Pyannote.audio	Full pipeline; pretrained HF models	Limited long-file speed; API gating
WavLM-based SSL	Noise-robust; strong generalization	Large model, slow for real-time
EEND	End-to-end, overlap-aware, no clustering	Hard to train; domain mismatch

D. Self-Supervised Transformer Embeddings

Self-supervised learning (SSL) has significantly advanced speaker diarization by providing robust, large-scale pretrained representations that generalize well across domains. SSL models are trained on massive unlabeled speech corpora, learning acoustic and phonetic structure without supervision. Their contextual representations provide stronger speaker discrim-

ination, robustness to noise, and improved performance on short or overlapping speech segments.

- *Wav2Vec2.0*: [28] introduced a contrastive learning framework where a Transformer contextualizes latent speech representations derived from a CNN encoder. These embeddings capture high-level speech patterns and have been shown to improve clustering quality in diarization pipelines, especially when combined with VBx or pyannote back-ends.
- *HuBERT*: [29] proposed masked prediction of pseudo-labels derived from k-means clustering. By iteratively refining cluster assignments, HuBERT learns phonetic-aware and temporally stable features, making it particularly effective for segmentation and speaker turn detection.
- *WavLM*: [30], one of the most recent SSL models, was designed specifically for "full-stack" speech tasks, including diarization. Through joint masked prediction and denoising training on 94,000 hours of noisy and clean audio, WavLM produces representations robust to real-world noise, reverberation, and multi-speaker overlap. WavLM embeddings combined with ECAPA-TDNN back-ends have achieved state-of-the-art results on AMI, DIHARD III, and VoxConverse.
- *Whisper-based diarization* leverages OpenAI's Whisper ASR model [31], which uses a large Transformer encoder trained on 680k hours of multilingual audio. Although Whisper was not designed for diarization, its encoder representations, when paired with a clustering or segmentation model, produce surprisingly strong diarization performance due to their multilingual, noise-robust nature.

E. End-to-End Neural Diarization (EEND)

EEND first introduced by [4], formulates speaker diarization as a multi-label frame classification problem, directly predicting the joint activity matrix of all speakers without relying on clustering or segmentation back-ends. Unlike x-vector or pyannote-based pipelines where speaker embeddings and clustering determine speaker assignment, EEND learns the correspondence between acoustic features and speaker activity in a fully neural way, enabling it to model overlapping speech naturally. Extensions such as EEND-EDA (Encoder–Decoder Attractors) [32] introduce attractor vectors to handle the permutation ambiguity between output speaker streams, allowing the system to dynamically associate each predicted activity stream with a speaker instance. Other variants incorporate vector-clustering mechanisms to improve robustness on long recordings [33], while EEND with speaker counting [34] estimates the number of active speakers directly from the audio, enabling flexible diarization without a fixed-speaker assumption. Collectively, these advances make EEND a powerful paradigm for real-time and overlap-heavy diarization tasks.

Table I summarizes the strengths and weaknesses of major modern diarization approaches, including x-vector baselines, ECAPA-TDNN, PyAnnote, WavLM-based SSL systems, and EEND.

V. Datasets for Speaker Diarization

A disadvantage of SD approaches based on models is their reliance on external data for the training, which makes them less robust in acoustic conditions not seen in training. So, data are important. To evaluate SD performance, one typically uses annotated datasets, i.e., sets of audio containing multiple people talking, where time stamps are labelled manually. Training can be carried out on any audio, but supervised training would use labelled datasets, which could be from natural sources (e.g., YouTube, public broadcasts). Training sometimes uses "synthetic" datasets, where individual speakers' speech (i.e., from monologues, not conversations) are superimposed. Such simulated mixtures often have added noise and reverberation with simulated room impulse responses. SD performance varies with how well the simulated data used in training matches real (target domain) conversations [38]. Table II presents the popular datasets and their characteristics.

VI. Evaluation Metrics

A. Diarization Error Rate (DER)

The DER [35] is the most common metric to evaluate diarization systems. The DER is made up of three different errors. These include:

- False Alarm (FA): When speech is recognized even though there is no speech in the segment.
- Missed Speech (MS): If speech is not recognized although there *is* speech in the segment.
- Speaker Confusion (SC): If the wrong speaker is assigned to a segment.

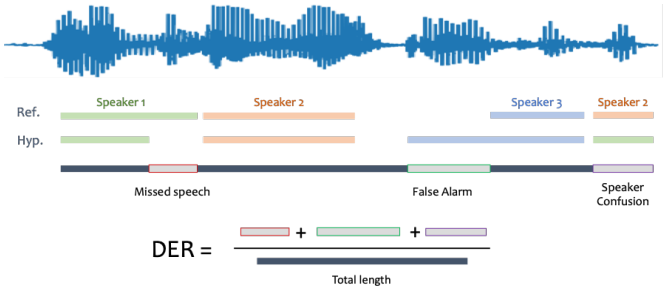

Fig. 5: Visualization of DER.

The DER is then calculated from the sum of the errors divided by the duration of the whole audio file, as shown below:

$$\text{DER} = \frac{\text{FA} + \text{MS} + \text{SC}}{\text{Total Duration Of Time}}$$

A visualization of the Diarization Error Rate (DER) is shown in Fig. 5, illustrating how missed speech, false alarms, and speaker confusion contribute to the overall error. Although the overall DER is reported in the evaluation of diarization systems, in some cases, the individual error components are also reported.

TABLE II: Commonly Used Speaker Diarization Datasets and Their Characteristics

Dataset	Domain / Scenario	Languages	Size / Duration	Notes / Challenges
ICSI Meeting Corpus [39]	Conference/meeting audio	English	75 h	Early SD dataset; multi-speaker, multi-channel
NIST Rich Transcription [40]	Meeting / conference	English	Several hours across RT02–RT09	Standard benchmark for early diarization research
CALLHOME	Telephone speech	Multilingual	500 sessions, 2–6 speakers	Short conversations (2–5 min), strong channel variability
DIHARD I–III [41]	"Hard" real-world audio	Many (varies by domain)	40–60 h across editions	Highly challenging: noise, reverberation, overlapped speech, domain mismatch
AMI Meeting Corpus [42]	Meeting room conversations	English	100 h	Multi-microphone, natural overlap, multi-party interaction
CHiME Series [42]-[43]	Dinner parties	English	50–60 h	Strong background noise, far-field arrays, overlapping conversations
Fearless Steps [44]	Apollo mission recordings	English	Thousands of hours	Extremely noisy, long-duration archive data, multi-channel recordings
IberSpeech RTV [45]	Broadcast TV	Spanish	20–30 h	Mixed-language broadcast, natural transitions
Aishell-4 [46]	Mandarin meeting corpus	Mandarin	211 recorded sessions, 4-8 speakers	8-channel microphone array speech
VoxConverse [47]	Real conversations (YouTube)	English	20 h	Video+audio diarization; realistic noise
VoxCeleb 1/2	Interview videos on YouTube	English	2,000+ h, 7000+ speakers	Used mainly for training embeddings (x-vectors, ECAPA)
LibriCSS [48]	Read speech mixtures	English	10 h, 40 speakers	natural room reverberation, realistic spatial cues, and microphone variability
DISPLACE 2023	Multi-language conversations	Multilingual	—	High noise, reverberation, overlapping speech
This American Life [49]	Podcast-style conversations	English	—	Freely available; clear but multi-speaker

TABLE III: Comparison of DER, JER, and CDER Metrics

Metric	Description and Characteristics
DER	Frame-duration-based error metric combining Missed Speech, False Alarm, and Speaker Confusion. Sensitive to long-duration segments.
JER	Intersection-over-Union based metric focusing on segment-level overlaps. More robust to segmentation errors but still time-weighted.
CDER	Utterance-level metric that merges same-speaker segments and evaluates correctness at the conversational-unit level. More sensitive to short utterances and dialogic structure.

B. Jaccard Error Rate (JER)

Another metric used in some studies is the JER [36]. Introduced for DIHARD; measures segment overlap accuracy. The aim of this metric is to rate each speaker with the same weight, regardless of the speaker's total speaking time. To do this, the error per speaker is first calculated and then divided by the speaker's speaking time. The error per speaker is the sum of FA and MS, as shown in Equation below:

$$\text{JER} = \frac{1}{N} \sum_{i=1}^{N_{\text{ref}}} \frac{\text{FA}_i + \text{MS}_i}{\text{TOTAL}_i}$$

C. Conversational Diarization Error Rate (CDER)

DER fails to give enough importance to short conversational phrases, which are short but important on the semantic level. Also, a carefully and accurately manually annotated testing dataset suitable for evaluating the conversational SD technologies is still unavailable in the speech community. Besides, JER uses Jaccard similarity index that is the ratio between the sizes of the intersections and unions of two sets of segments. In this scenario, CDER evaluation metric, which calculates the SD accuracy at the utterance level where all types of mistakes are equally reflected regardless of the length of the spoken sentence, was proposed [37].

For CDER calculation, firstly, consecutive utterances were merged that belong to the same speaker. For instance, if A_i denotes the i-th utterance from speaker A, and NS represents a non-speech segment, a sequence such as $A_1, NS, A_2, NS, B_1, A_3, A_4, B_2, A_5, C_1$ will be merged into $A'_1, B'_1, A'_3, B'_2, A'_5, C'_1$. Each merged utterance (e.g., A'_1) preserves the timestamp of the first utterance in the sequence (the start time of A_1) and the timestamp of the last utterance in that sequence (the end time of A_2). After obtaining the merged utterances for both the reference and the hypothesis, each reference merged utterance was matched to its corresponding hypothesis utterance. For every matched pair, their time intervals need to compare next to determine whether the prediction is correct or not. This comparison is based on the overlap ratio between the reference interval R and the hypothesis interval H; if

$$\frac{|R \cap H|}{|R \cup H|} < \eta,$$

the utterance is labeled as an error. The total number of such errors is accumulated, and finally, the CDER is computed according to:

$$\text{CDER} = \frac{N_{\text{Total Mistakes}}}{N_{\text{Total Utterrences}}}$$

This process ensures that all types of diarization errors are treated equally at the utterance level, making CDER more sensitive to conversational scenarios where even short segments contain meaningful information.

Table III compares the primary evaluation metrics used in diarization—DER, JER, and CDER.

VII. Recent Research Trends

A. Overlap-Handling Neural Systems

Recent diarization research places strong emphasis on handling overlapping speech, which remains one of the most challenging aspects of natural conversations. End-to-End Neural Diarization (EEND) models [4], [32] directly estimate multi-speaker activity without clustering, enabling accurate overlap attribution even in dense, multi-talker recordings. Multi-channel spatial models and beamforming-based techniques further improve overlap resolution by leveraging microphone array geometry to separate simultaneous speakers in far-field environments.

B. Self-Supervised Speech Models

SSL has reshaped diarization by providing robust representations learned from massive unlabeled corpora. Models such as Wav2Vec2.0, HuBERT, and especially WavLM [30] significantly improve speaker discrimination, noise robustness, and generalization. When combined with ECAPA-TDNN or VBx clustering, SSL-based embeddings achieve state-of-the-art performance on AMI, DIHARD III, and VoxConverse. The ability of SSL models to learn from raw audio without labels is particularly impactful in low-resource diarization settings.

C. Audio-Visual Diarization

Audio-visual diarization has gained renewed momentum with the introduction of datasets such as VoxConverse-AV and models like VGG-Sync [50]. By fusing facial embeddings, lip motion, and synchronization cues with audio signals, these systems achieve superior speaker tracking in noisy or crowded environments.

D. Real-Time Diarization

Real-time diarization is increasingly important for live transcription, broadcast production, and online conferencing platforms. Streaming EEND variants [51] address latency constraints by chunking audio into small frames while maintaining high accuracy. Additionally, low-latency ECAPA embeddings and lightweight clustering algorithms enable diarization systems to run efficiently on edge devices and real-time ASR pipelines aiming to bridge the gap between offline diarization research and practical deployment requirements.

VIII. Open Challenges & Future Directions

Although significant progress has been made in speaker diarization, several open research challenges remain and continue to motivate modern work. One persistent obstacle is the reliable detection and attribution of overlapping speech, which remains the dominant source of diarization error in natural conversations [24]. Robustness to noise, reverberation, far-field microphone distortions, and domain mismatches—such as those encountered in telephone, broadcast, and meeting-room audio—also remains limited, even with state-of-the-art neural and self-supervised models [30]. Another emerging direction is cross-lingual and multilingual diarization, particularly for conversational and code-switched speech, where pretrained models often fail to generalize reliably across languages [52]. Low-latency and streaming diarization is also becoming increasingly important for online platforms such as Zoom, Teams, and Whisper-based ASR pipelines, yet end-to-end systems like EEND still struggle with real-time constraints and limited training data [51]. Furthermore, despite the availability of large pretrained models, the field lacks abundant labeled data for supervised adaptation, making semi-supervised and self-supervised learning crucial for future advancements. Lastly, multimodal diarization, combining audio with visual cues, represents a promising direction for achieving more stable speaker tracking in complex environments [53]. Addressing these challenges will be essential for developing diarization systems that are accurate, scalable, and robust across real-world acoustic conditions.

IX. Conclusion

Speaker diarization has progressed from classical Bayesian segmentation and clustering-based systems to modern deep-learning architectures such as x-vectors, ECAPA-TDNN, and EEND. Recent advances in self-supervised speech models (e.g., WavLM, HuBERT), robust overlap-handling techniques, and large speech models such as Whisper have further pushed the state of the art across challenging domains. Toolkits like pyannote.audio have made high-quality diarization accessible and reproducible, enabling rapid research and deployment. As multimodal (audio–visual) integration, domain adaptation, and real-time processing continue to improve, speaker diarization is expected to play an increasingly central role in large-scale ASR systems, meeting transcription, conversational AI, and intelligent human-machine interaction.

References

[1] Douglas O'Shaughnessy *et al.*, "Speaker Diarization: A Review of Objectives and Methods," in *MDPI*, 2025.

[2] La Javaness *et al.*, "Speaker Diarization: An Introductory Overview," in *Medium*, 2023.

[3] D. Snyder *et al.*, "X-vectors: Robust DNN embeddings for speaker recognition," in *IEEE ICASSP*, 2018.

[4] Y. Fujita *et al.*, "End-to-end neural speaker diarization," *ASRU*, 2019.

[5] H. Bredin, "pyannote.audio: Neural building blocks for speaker diarization," in *ICASSP*, 2020.

[6] L. R. Rabiner and M. R. Sambur, "An algorithm for determining the endpoints of isolated utterances," Bell System Technical Journal, 1975.

[7] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," IEEE Signal Processing Letters, 1999.

[8] Google WebRTC Project, "WebRTC Voice Engine," 2011.

[9] S. N. Davis et al., "Support vector machine based voice activity detection for noisy speech," INTERSPEECH, 2006.

[10] S. Chen and P. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the Bayesian Information Criterion," DARPA Broadcast News Workshop, 1998.

[11] S. S. Chen et al., "Robust speaker segmentation using GLR," ICASSP, 1999.

[12] H. Gish, M. Siu, and R. Rohlicek, "Segregation of speakers for speech recognition and speaker identification," ICASSP, 1991.

[13] D. A. Reynolds et al., "Speaker verification using adapted Gaussian mixture models," Digital Signal Processing, 2000.

[14] P. Kenny et al., "Joint factor analysis versus eigenchannels in speaker recognition," IEEE Trans. Audio, Speech, and Language Processing, 2007.

[15] N. Dehak et al., "Front-end factor analysis for speaker verification," IEEE Trans. Audio, Speech, and Language Processing, 2011.

[16] D. A. Reynolds and P. Torres-Carrasquillo, "The MIT Lincoln Laboratory RT-04F diarization systems," RT-04F Workshop, 2004.

[17] U. Von Luxburg, "A tutorial on spectral clustering," Statistics and Computing, 2007.

[18] M. J. F. Gales and S. J. Young, "The application of hidden Markov models in speech recognition," Foundations and Trends in Signal Processing, 2008.

[19] J. MacQueen, "Some methods for classification and analysis of multivariate observations," Berkeley Symposium on Mathematical Statistics and Probability, 1967.

[20] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in Proc. IEEE ICASSP, 2018, pp. 5329–5333.

[21] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "SPEAKER RECOGNITION FOR MULTI-SPEAKER CONVERSATIONS USING X-VECTORS," in Proc. IEEE ICASSP, 2019.

[22] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in Proc. Interspeech, 2020.

[23] X. Ma, H. Bredin, Y. Lei, "Speech activity detection: A deep learning-based approach," IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2021.

[24] S. Narayanaswamy, et al., "Neural overlap detection for multi-speaker diarization," in Proc. IEEE ICASSP, 2023.

[25] H. Bredin, "pyannote.audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe," in Proc. Interspeech, Dublin, Ireland, 2023.

[26] H. Bredin, "PYANNOTE.AUDIO: NEURAL BUILDING BLOCKS FOR SPEAKER DIARIZATION," ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020.

[27] Omkar Bhangari et al., "Deep neural networks for small footprint text-dependent speaker verification," in Proc. IEEE Convergence in Technology (I2CT), Pune, India. Apr 5-7, 2024.

[28] A. Baevski *et al.*, "wav2vec 2.0: A framework for self-supervised learning," *NeurIPS*, 2020.

[29] W. N. Hsu, B. Bolte, Y. H. Tsai, et al., "HuBERT: Self-supervised speech representation learning by masked prediction," in *Proc. NeurIPS*, 2021.

[30] S. Chen *et al.*, "WavLM: Large-scale pre-training for full-stack speech processing," *IEEE TASLP*, 2022.

[31] A. Radford, J. Kim, T. Xu, et al., "Robust speech recognition via large-scale weak supervision," in *Proc. ICML*, 2023.

[32] S. Horiguchi, Y. Fujita, N. Chen, et al., "Encoder–decoder attractors for end-to-end neural diarization," in *Proc. Interspeech*, 2020.

[33] Y. Fujita, S. Horiguchi, N. Chen, et al., "End-to-end neural diarization with vector clustering," in *Proc. IEEE SLT*, 2021.

[34] N. Kanda, Y. Gao, S. Horiguchi, et al., "Joint speaker counting, speech recognition, and diarization," in *Proc. IEEE ASRU*, 2021.

[35] J. G. Fiscus, J. Ajot, M. Michel, and J. S. Garofolo, "The rich transcription 2006 spring meeting recognition evaluation," in Machine Learning for Multimodal Interaction: Third International Workshop, MLMI 2006, Bethesda, MD, USA, May 1-4, 2006, Revised Selected Papers 3. Springer, 2006, pp. 309–322.

[36] N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman, "The Second DIHARD Diarization Challenge: Dataset, task, and baselines," Jun. 2019, arXiv:1906.07839 [cs, eess]. [Online].

[37] G. Cheng, Y. Chen, R. Yang, Q. Li, and others, "The Conversational Short-phrase Speaker Diarization (CSSD) Task: Dataset, Evaluation Metric and Baselines," 2022 13th International Symposium on Chinese Spoken Language Processing (ISCSLP) , 2022.

[38] O'Shaughnessy, D. "Speaker Diarization: A Review of Objectives and Methods", Appl. Sci. 2025, 15, 2002.

[39] J. Fiscus, J. Ajot, and J. Garofolo, "The Rich Transcription 2007 meeting recognition evaluation," in Proc. Int. Workshop on Rich Transcription, Baltimore, MD, USA, 10–11 May 2007, pp. 373–389.

[40] M. Przybocki and A. Martin, "A 2000 NIST Speaker Recognition Evaluation: Linguistic Data Consortium," Philadephia, PA, USA, 2011.

[41] Ryant, N.; Singh, P.; Krishnamohan, V.; Varma, R.; Church, K.; Cieri, C.; Du, J.; Ganapathy, S.; Liberman, M. "The third DIHARD diarization challenge", in Proceedings of the Interspeech, Brno, Czechia, 30 August–3 September 2021; pp. 3570–3574.

[42] C. Boeddecker, J. Heitkaemper, J. Schmalenstroeer, et al., "Front-end processing for the CHiME-5 dinner party scenario," in Proc. CHiME5 Workshop, Hyderabad, India, 2018.

[43] S. Cornell, S. Wiesner, M. Watanabe, et al., "The CHiME-7 DASR Challenge: Distant Meeting Transcription with Multiple Devices in Diverse Scenarios," arXiv:2306.17340, 2023.

[44] Hansen, J.H.; Joglekar, A.; Shekhar, M.C.; Kothapally, V.; Yu, C.; Kaushik, L.; Sangwan, A. The 2019 inaugural Fearless Steps challenge: A giant leap for naturalistic audio. In Proceedings of the Interspeech, Graz, Austria, 15–19 September 2019.

[45] Perero-Codosero, J.M.; Antón-Martín, J.; Merino, D.T.; Gonzalo, E.L.; Gómez, L.A.H. Exploring Open-Source Deep Learning ASR for Speech-to-Text TV program transcription. In Proceedings of the IberSPEECH, Barcelona, Spain, 21–23 November 2018; pp. 262–266.

[46] Y. Fu, Y. Cheng, L. Lv, et al., "AISHELL-4: An open source dataset for speech enhancement, separation, recognition and speaker diarization in conference scenario," in Proc. Interspeech, Brno, Czechia, 30 Aug.–3 Sept. 2021.

[47] Chung, J.S.; Huh, J.; Nagrani, A.; Afouras, T.; Zisserman, A. Spot the conversation: Speaker diarisation in the wild. In Proceedings of the Interspeech, Shanghai, China, 25–29 October 2020; pp. 299–303.

[48] Chen, Z.; Yoshioka, T.; Lu, L.; Zhou, T.; Meng, A.; Luo, Y.; Wu, J.; Xiao, X.; Li, J. "Continuous speech separation: Dataset and analysis", in Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 7284–7288.

[49] H. Mao, H. Li, S. McAuley, J. Cottrell, "Speech recognition and multi-speaker diarization of long conversations," in Proc. Interspeech, Shanghai, 25–29 Oct. 2020.

[50] J. S. Chung and A. Zisserman, "VGG-Sync: Cross-modal self-supervised learning for audio-visual synchronization," arXiv:1910.06464, 2019.

[51] N. Kanda et al., "Streaming end-to-end neural speaker diarization for real-time applications," IEEE SLT, 2022.

[52] N. Shah, et al., "The VoxSRC-23 Speaker Recognition Challenge," in *Proc. Interspeech*, 2023.

[53] Y. Zhu, et al., "AV-Diarization: Audio-visual speaker diarization in the wild," in *Proc. CVPR*, 2023.