Speech Diarization for Cocktail Party Setting Data

Human Data Analytics Project Report

Nikoloska Nora[†],

***Abstract*—Speech Diarization or the problem known as "who spoke when" is a common processing step before Automatic Speech Recognition. The standard diarizaition pipeline consists of several sub-tasks that each have unique possibilities for performance optimization. This project explores the diarization pipeline with an text independent input in a cocktail party setting with speaker overlap and background music. The custom data is a recorded conversation along with ground truth data for processing end evaluation. The relationships between types of input features (MFCCs, trained RNN embeddings, Log Mel Filterbank Energies and STFT) with regards to obtained accuracy are reported. Selected clustering algorithms are evaluated (Spectral Clustering, K-Means, GMM, HDBSCAN and DBSCAN). Finally, an end to end BLSTM based diarization architecture with windowed features is proposed and tested with a custom implementation of a Diarization Error Rate (with and without overlap) metric. Best results are obtained for the SFT features. This report provides an insight into the possibilities of automatic diarization and end to end online inference.**

***Index Terms*—Speech Diarization, Cocktail Party Problem, Clustering, Voice Activity Detection, Speaker Embedding, BLSTM**

I. Introduction

The *"Cocktail Party Effect"* is a psychological phenomenon coined in 1953 by E. Colin Cherry [1] and is described as the human ability to shift the attention between different overlapping speakers in a party setting. In the world of automatic machine speech comprehension, microphone recorded data which fuses the overlapping voices in a single signal makes the task of source separation hard and thus forms the *"Cocktail Party Problem"*.

Among the different challenging tasks encapsulated by the cocktail party problem such as automatic speech recognition, speaker recognition and source separation, **speech diarization** is the problem known as *"who spoke when"* and aims to produce a speaker specific voice activation function for a given mixed audio data. The diarization task is particularly challenging in the cocktail party scenario because of the speaker overlap fragments and the presence of background music or environmental noise. The diarization task itself has applications in various domain problems such as telephone conversations in medical or security settings, conference meeting recordings and as a prepossessing step before any *ASR (Automatic Speech Recognition)* pipeline.

This project explores the different parts of the diarization task pipeline and the approaches to this problem in the context of cocktail party text independent data. The diarization pipeline components are: *Front-End processing*, *Voice Activity Detection (VAD)*, *Segmentation*, *Speaker Embedding*, *Clustering and Post Processing* [2].

This project focuses on the application of existing approaches on a custom conversation files, aiming to exploit unsupervised approaches and compare the obtained results. Many of the datasets used in the context of speech diarization are simulated datasets generated with the mixing of previously available speech recordings. Some examples are the NIST SRE CALLHOME dataset [3] that is composed of telephone recordings and the AMI Meetings dataset [4]: a collection of meeting room conversations. These are examples of real, but controlled conversations, and thus rarely contain overlaps. For simulating a dataset that targets the cocktail party problem, a common approach is generating overlapped recordings from the Wall Street Journal Corpus [5], but this approach generates unrealistic synthetic mixtures. Many of the speech diarization models, as seen in Section V underperform when fitted with real world noisy data. One attempt for a real world dataset is the recently released VoxConverse Corpus [6] which contains YouTube recordings of multilingual speakers with background noise, music and speaker overlap.

List of project contributions:

- This project showcases an overview of the available tools for fast and accurate diarization of custom two speaker recording in a cocktail party setting
- Various approaches in all of the parts of the diarization pipeline are explored by experiments with realistic data
- As an alternative to the standard pipeline, some End to End diarization architectures are explained, with focus on models based on Bidirectional LSTMs
- Finally, the results comparison from the proposed architecture for both short and long conversation files can influence the choice of optimal input features in the future

The report is structured as follows: Section II presents an overview of the most influential approaches in the literature for each of the tasks in the speech diarization pipeline. Section III explains the data collection and processing and highlights the chosen algorithms for each sub-problem. Section IV shows a proposed end to end diarization network with the architecture and implementation details. The obtained results from different experiment are presented and commented on in Section V. Finally, Section VI consists of project conclusion and future work ideas.

[†]Department of Mathematics, University of Padova, email: nora.nikoloska@studenti.unipd.it

II. Related Work

When thinking of the cocktail party problem the standard literature approach is speech separation using a supervised technique trained on synthetically generated mixed dataset. One of the standard methods for this problem is ICA (Independent Component Analysis) [7] and possibly the introduction of higher order moments [8]. Recently, the task has been reported as solved, with the application of a deep learning architecture on a dataset of musical mixtures [9]. However, the application this approach requires multiple recordings: one for each independent component on a relatively small distance from the source speakers.

The cocktail party problem has also been associated as synonymous with the task of speaker recognition, often modelled as a neural network [10]. One of the possible applications here can is a Siamese neural network [11]. These closely related tasks can also indicate the choice of extracted audio features, since they aim to maximize the distance between speakers. The authors in [12] recommend the practical use of spectral features such as *Mel-Frequency Cepstral Coefficients (MFCCs)* or *Linear Predictive Cepstral Coefficients (LPCCs)* and their combinations for this task.

The literal meaning of the word *'diarize'* is keeping a note or an event in a diary. From here, the goal of the task of speech diarization is to determine when different speakers are active in a recording. The speakers do not have to be annotated beforehand, which makes every permutation of speaker labels and obtained clusters equally correct.

The speech diarization pipeline starts with *Front End Speech Preprocessing* which aims to increase value of the speech segments. One of these noise reduction methods is speech enhancement based on an LSTM trained on speech [13]. Another approach is an ICA separation by expoiting MFCC features [14]. One important effect for the cocktail party problem occurs in reverberated environments where the signal from the speakers is echoed from the walls if the setting is in a closed space. Dereverberation methods aim to reduce this type of noise and increase the speech signals [15].

The next part in a diarization system after the initial preprocessing is *Voice Activity Detection* which aims to separate the silence from the speech segments. Some examples of VAD solutions are based on SVMs by using MFCC features [16], while others exploit LSTMS for noise reduction [17].

Another task is the feature representation in a manner that maximizes the difference between the speakers and minimizes the distance of vectors per speaker. There are various feature embeddings used for the diarization task. Apart form the MFCCs and LPCCs there are also proposed combinations of MFCCs and neural network predicted embeddings [18]. In the choice between short-term cepstral features that and long-termn prosodic features that contain more information about the intonation and rhythm, there is evidence that the combination of both is more favorable [19], even though short term features are better fitted for speaker embeddings.

The final part of the pipeline is clustering the speaker embeddings in separate groups which depends on the choice of a clustering algorithm. Very often, the *Spectral Clustering* algorithm is used in the diarization task [20]. The algorithm works as follows:

1) Generation of a distance matrix
2) Conversion of the distance matrix according to an affinity score or node similarity into an affinity matrix A
3) Calculation of the degree matrix D and the Laplacian $L = D - A$
4) Matrix generation of the k largest eigenvalues of L which determines the number of clusters
5) Vector normalization
6) Clustering in the k-dimensional space often using K-Means

This algorithm has better performance than others for detecting modified cluster shapes and cluster number.

III. Processing Pipeline

This section explains each sub-problem present in the standard pipeline show on Figure 1 for the speech diarization task. It is tailored to the given input data and includes *Overlap detection* while excluding some optional front end and post processing of the data.

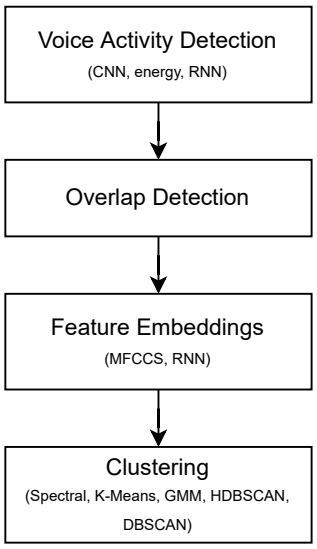
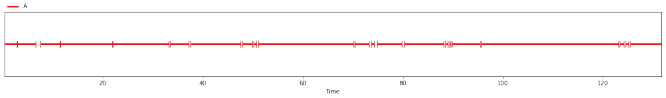


Fig. 1: Processing Pipeline

A. Data Collection

All of the Experiments with various tools in this section are done on a 2 minute recording with two speakers (male and female) on a relative distance with background music and overlapping speech. In addition to the mixed recording, ground truth recordings are created for both speakers in order to evaluate the results. The first attempt done with the pyannote-audio full speech diarization pipeline is shown on Figure 2 and as it can be seen that only one speaker is detected. From here, the analysis follows the diarization pipeline by parts.



(a) Pyannote result

(b) CNN based VAD result

Fig. 2: Processing tools results

B. Voice Activity Detection

Several tools for speech detection have been used on the given input recording. The first one is a CNN based speech segmentation tool: inaSpeechSegmenter trained on the following classes: male and female speech, music and noise. As seen on Figure 2 the model fails to recognize anything but noise, presumably due to the frequent overlap between the music and speech in the given recording. Interestingly, the white noise levels in the given recording are very low and after a short initial peak do not surpass the value of 0.07. In the process of ground truth creation, introduced noise by the amplitude thresholding operation is removed by the noisereduce spectral gating application.

The second method is an energy based VAD that calculates the energy ratios per each window and selects the speech regions based on a threshold. The ground truth percentage of speech presence in the recording is 78%. The accuracy of this method is 22% when computed by frame and 69% when the vector is resampled to 2 samples per second.

The best accuracy of 93.46% for VAD is found when running inference on the RNN based voice detector: webrtcvad. Because of the high accuracy, the detected silence segments are removed and the signal is updated before being used as input the clustering step.

C. Overlap Detection

The percentage of ground truth overlap between the two speakers for the given recording is 33%. Another attempt with pyannote-audio has been made for overlap detection with a low accuracy of 19%. For this reason, the detected overlapped regions are not removed from the input signal.

D. Feature Embedding

The next part of the diarization pipeline is feature generation. Two types of feature embeddings have been tested together with different clustering methods. The first one is the standard approach of extracting *Mel-Frequency Cepstral Coefficients*. The MFCC algorithm consists of:

1) Pre-emphasis and framing of the input signal into shorter frames
2) *Fast Fourier Transformation* and short term power spectrum calculation
3) Frequency Warping and computation of *Mel frequency Filter Bank*
4) The logarithm operation
5) *Discrete Cosine Transform* of the scaled speech
6) Selection of the first 12 coefficients

In order to create the feature vector, the first and derivatives that of the 12 coefficients are added to represent the change of the coefficients in time. For increased accuracy, the energy value per sample is added to the vector with its first and second derivative. The final vector has length of

$$[12 + 12 + 12 + 1 + 1 + 1] = 39$$

features.

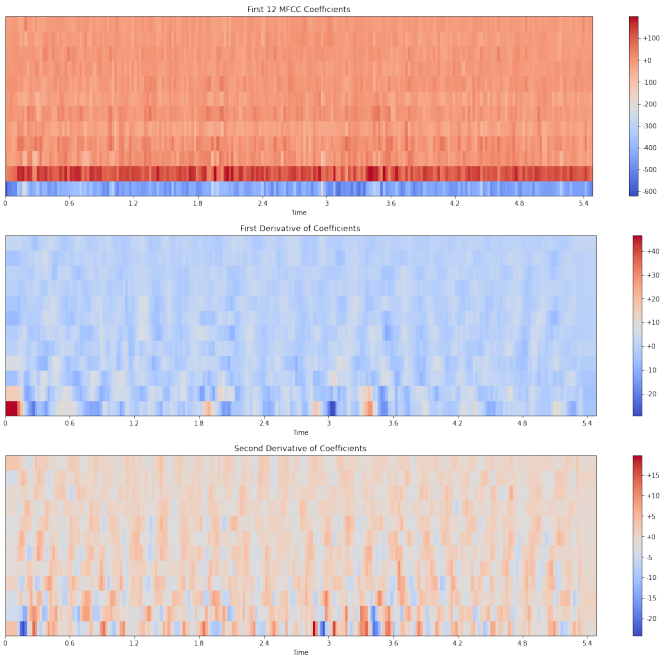


Fig. 3: MFCC Features with first and second derivatives

The second feature embedding is computed by using the Resemblyzer encoder proposed as a method in the task of speaker verification [21]. This is a pre-trained model that generates a vector of 256 embeddings per frame for a given audio.

E. Clustering

The final step of the diarization pipeline without considering the postprocessing is the clustering part. Table 1 shows a comparison in accuracy values for each clustering algorithm and choice of features. The task of computing the accuracy for different permutations of the predicted labels is often solved by the *Hungarian algorithm*. Here, since there are two speakers, the reported accuracy values are the maximum between the two alternatives of speaker - label assignment.

Method	MFCC	Resemblyzer
Spectral Clustering	12.71%	16.10%
K - Means	12.71%	13.98%
GMM	13.56%	13.98%
HDBSCAN (min 5)	23.3%	0.0%
DBSCAN	0.0%	0.0%

TABLE 1: Comparison between the feature choice and clustering algorithms

The accuracy values reported in Table 1 are very low, but they also depend on the prepossessing done by the voice activity detection. Another limitation is that the ground truth values are sampled with a rate of two samples per second, whereas sampling every 1-2 seconds is often used in order to preserve speaker uttterances. In any case, it can be seen that the Resemblyzer features perform better than MFFCC almost always, except in the case of *HDBSCAN*. This algorithm is the only one that allows soft clustering which means that the samples can belong to more than one cluster. The output of the algorithm contains probabilities in addition to the cluster labels. In this example, the label is considered in the accuracy computation only if it has a probability greater than 0.5 which poses an additional limit to the algorithm. This algorithm failed to detect more than one cluster when using the Resemblyzer embeddings, even when the minimum number of samples per cluster is set to 2. The same is the case with the DBSCAN algorithm which obtains 0.0% accuracy on both feature sets.

F. Diarization Error Rate

Instead of computing all accuracy values separately, there is a standardized error rate for the diarization task. The *Diariazion Error Rate* (DER) is defined as:

$$DER = \frac{FA + MISS + OVERLAP + CONFUSION}{Total\ Time}$$

where:

- False Alarms (FA) are true silences in which the VAD predicts speech
- MISS segments are true speech segments the VAD fails to detect
- OVERLAP error is failure to detect overlap and it is sometimes omitted from the calculation
- CONFUSION is the only error that refers to the clustering step and increases with the number of falsely clustered speakers

Alternatives to the DER are the *Jaccard Error Rate (JAR)* which is computed per speaker and the *Word-level Diarization Error Rate (WDAR)* used in speech recognition problems.

IV. End to End Diarization

As an alternative to the standard diarization pipeline, End to End approaches with deep learning have gained attraction. The goal in these architectures is to tackle the problem of speech diarization at once, instead of splitting it to the pipeline sub-tasks. The diarization taks is normally modelled as a multilabel classification problem that can handle speaker overlaps and silence. The problem arises with the need for a permutation invariant loss function that will not penalize the speaker assignment to any given label.

The model shown on Figure 4 is inspired by the *Bidirectional LSTM (BLSTM)* architecture show in [22]. As the authors proposed, there are 5 interconnected BLSTM layers each of 256 units. They are structured in blocks with Batch Normalization layers that re-center the batch distribution and Dropout layers that regulate overfitting. After this are two Dense layers of 64 and 32 units, each followed by a Dropout layer. The output of each BLSTM layer can gain information in both directions: the past and the future. This is used because the diarization input contains the features from a chosen number of preceding and succeeding frames. This creates an input vector of the following shape:

$$[\#samples, windowsize, \#features]$$

. The output in the two speaker case is transformed in a multiclass classification problem using one hot encoding for the following classes:

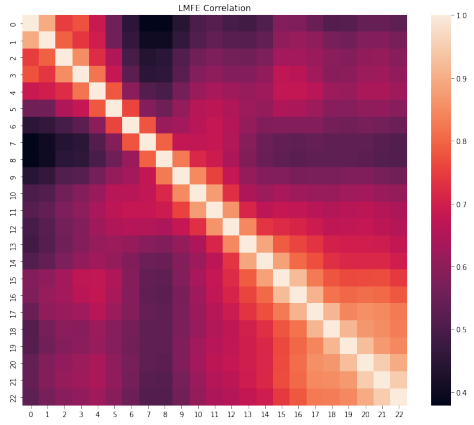
$$(0 = Silence, 1 = Speaker1, 2 = Speaker2, 3 = Overlap)$$

. The permutation invariance is instead simulated by reintroducing the true samples of a given speaker with opposite labels. A custom DER metric was implemented that computes the DER and DER without overlaps during training.

Other End to End diarization are based on CNNs [23], Encoder - Decoder architectures

V. Results

This section presents the training and testing results for the BLSTM model. Table 2 shows selected metric values for different choice of input features. The first figure shows the correlation between the 22 Log Mel Filterbank Energies as proposed by the the authors in [22]. The second figure shows the standard MFCC and the third shows the *Short-Time Fourier Transform* features. The last 4 features of the obtained vector are removed because of high correlation. It can be seen that there are large correlation values near the diagonal.



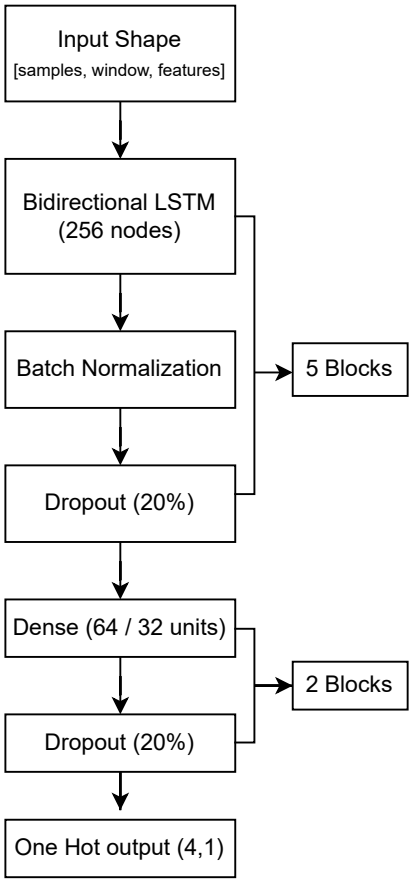


Fig. 4: Model Architecture

Each of the given models was trained on a reduced model of just 2 BLSTM blocks with a 2 minute log audio recording. The features were generated using windowing of size 7 with 3 past and 3 future features. Each model was trained for 20 epochs with batch size of 128, using Adam optimizer with a learning rate of 10^{-3}.

Features	Accuracy	MSE	F1 Score	DER	DER Θ
LMFE	66.06%	0.13	62.23 %	25%	19%
MFCCS	82.79%	0.06	82.31%	13%	7%
STFT	86.34%	0.05	85.95%	11%	6%

TABLE 2: Validation results for short data files
[Features: Log Mel Filterbank Energies, Mel-Frequency Cepstral Coefficients, Short-time Fourier transform, Metrics: Accuracy, Mean Squared Error, F1-Score, Diarization Error Rate and Diarization Error Rate without overlaps]

Finally, detailed training metrics of the best model using standardized STFT features is shown on Figure 6. This model uses the same data as input and the training time is increased to 30 epochs. The computed DER with and without overlap is shown on Figure 7. The obtained F1-Score metric for this model is 88.37% and the final DER values are 0.08 and 0.04 with and without overlap.

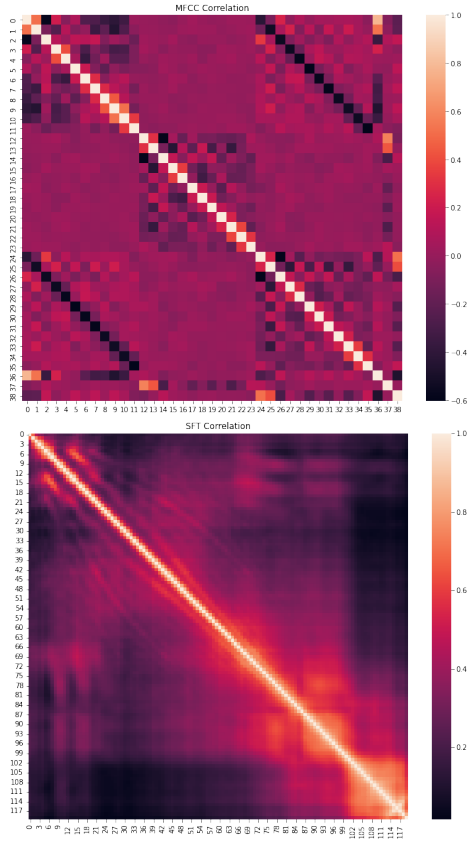


Fig. 5: Feature Correlation Matrices

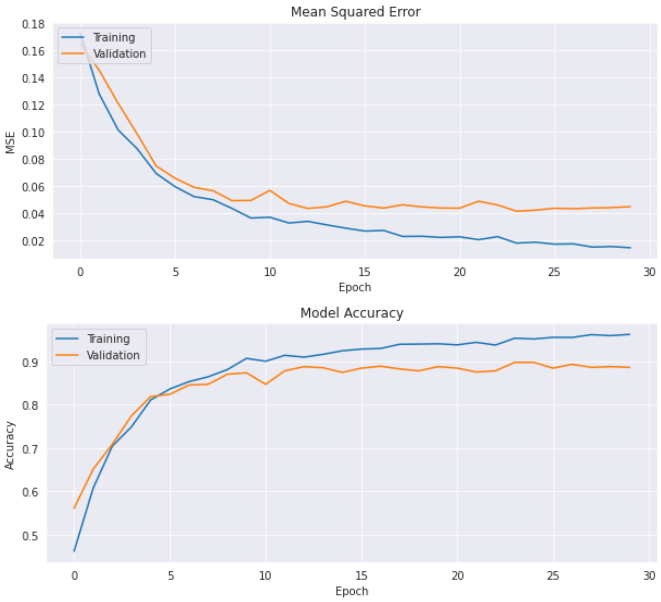


Fig. 6: Mean Squared Error and Accuracy

Figure 8 shows the confusion matrix for the final model and the classification of 5634 samples.

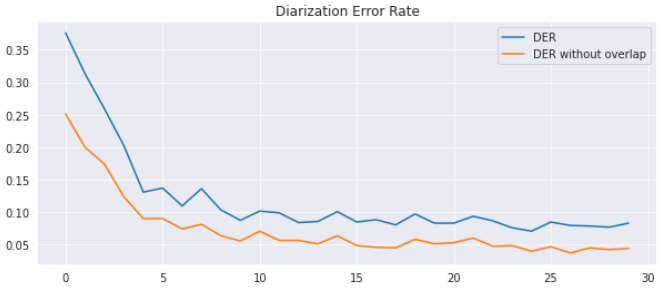


Fig. 7: Diarization Error Rate

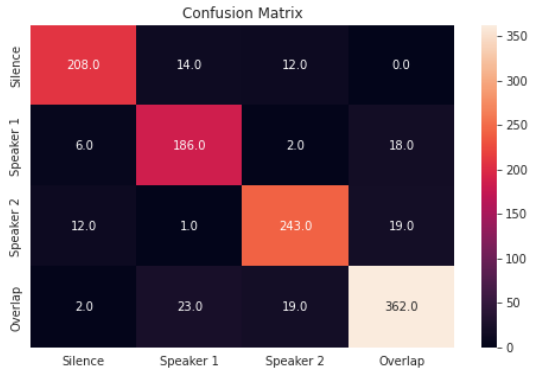


Fig. 8: Confusion matrix for final model

VI. Concluding Remarks

Many of the open, available diariation tools are not very precise when dealing with noisy input with background music and speaker overlap. In the presented implementation end-to-end architectures outperform the standard diarization pipeline, but the computational cost of the training is high for long conversations. With regard to the input features, neural network embeddings seem to be performing better than MFCCs and when compared as inputs to an end to end architecture, the standardized STFT features show the highest accuracy. The diarization error with the overlapping component is often overlooked which makes existing architectures hard to compare when adressing the cocktail party scenario.

In this project I found myself introduced to the world of audio processing and all of the speech recognition subproblems. I have explored audio processing techniques and a BLSTM architecture like I haven't been able to do so far. I learned what speech diarization is and the complex types of processing it entails. From a practical point of view, I learned the librosa API, Keras backend functions for custom loss and metric implementation and I got introduced to many existing projects as well as new clustering algorithms. The main challenges I faced are in the attempts of implementing a permutation invariant model, the formulation of the tasks and the problems and navigating the realm of audio data processing in general.

Some of my ideas for future work:

- Implementation of a permutation invariant loss with a low computational cost
- Investigation of the unsupervised clustering application as part of the End to End architectures
- Implementation of collar aware DER that does not penalize mistakes in close proximity to speaker change segments

References

[1] E. C. Cherry, "Some experiments on the recognition of speech, with one and with two ears," vol. 25, pp. 975–979, 1953.

[2] 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," 2021.

[3] O. Sadjadi, "Nist sre cts superset: A large-scale dataset for telephony speaker recognition," 2021-08-16 04:08:00 2021.

[4] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner, "The ami meeting corpus: A pre-announcement," in *Machine Learning for Multimodal Interaction* (S. Renals and S. Bengio, eds.), (Berlin, Heidelberg), pp. 28–39, Springer Berlin Heidelberg, 2006.

[5] D. B. Paul and J. M. Baker, "The design for the wall street journal-based csr corpus," in *HLT*, 1992.

[6] A. Brown, J. Huh, J. S. Chung, A. Nagrani, and A. Zisserman, "Voxsrc 2021: The third voxceleb speaker recognition challenge," 2022.

[7] A. HyvÃ¤rinen and E. Oja, "Independent component analysis: algorithms and applications," vol. 13, pp. 411–430, 2000.

[8] J.-F. Cardoso, "Source separation using higher order moments," in *International Conference on Acoustics, Speech, and Signal Processing,*, pp. 2109–2112 vol.4, 1989.

[9] A. J. R. Simpson, G. Roma, and M. D. Plumbley, "Deep karaoke: Extracting vocals from musical mixtures using a convolutional deep neural network," 2015.

[10] Z. Chen, J. Li, X. Xiao, T. Yoshioka, H. Wang, Z. Wang, and Y. Gong, "Cracking the cocktail party problem by multi-beam deep attractor network," 2018.

[11] J. Zhu, M. Hasegawa-Johnson, and L. Sari, "Identify speakers in cocktail parties with end-to-end attention," 05 2020.

[12] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: from features to supervectors," vol. 52, pp. 12–40, 01 2010.

[13] M. Liu, Y. Wang, J. Wang, J. Wang, and X. Xie, "Speech enhancement method based on lstm neural network for speech recognition," pp. 245–249, 2018.

[14] L. Lina and F. Ilona, "Integrating fast-ica and mfcc methods for voice recognition with noise splitter," 01 2018.

[15] R. Rotili, E. Principi, S. Squartini, and B. Schuller, "A real-time speech enhancement framework for multi-party meetings," pp. 80–87, 11 2011.

[16] T. Kinnunen, E. Chernenko, M. Tuononen, and H. Li, "Voice activity detection using mfcc features and support vector machine," vol. 2, 03 2012.

[17] P. Sertsi, S. Boonkla, V. Chunwijitra, N. Kurpukdee, and C. Wutiwiwatchai, "Robust voice activity detection based on lstm recurrent neural networks and modulation spectrum," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 342–346, 2017.

[18] S. H. Yella, A. Stolcke, and M. Slaney, "Artificial neural network features for speaker diarization," in *2014 IEEE Spoken Language Technology Workshop (SLT)*, pp. 402–406, 2014.

[19] A. Zewoudie, J. Luque, and J. Hernando, "The use of long-term features for gmm- and i-vector-based speaker diarization systems," vol. 2018, 09 2018.

[20] H. Ning, M. Liu, and H. Tang, "A spectral clustering approach to speaker diarization," vol. 5, 01 2006.

[21] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," 2017.

[22] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with permutation-free objectives," 2019.

[23] M. HrÃºz and Z. ZajÃc, "Convolutional neural network for speaker change detection in telephone speaker diarization system," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4945–4949, 2017.