Hindawi
Computational Intelligence and Neuroscience
Volume 2021, Article ID 6151651, 8 pages
https://doi.org/10.1155/2021/6151651

*Research Article*

# Utterance Clustering Using Stereo Audio Channels

**Yingjun Dong**[1,2] **Neil G. MacLaren**[1,3] **Yiding Cao**[1,2] **Francis J. Yammarino**[1,3] **Shelley D. Dionne**[1,3] **Michael D. Mumford**[4] **Shane Connelly**[4] **Hiroki Sayama**[1,2,3] **and Gregory A. Ruark**[5]

[1]*Center for Collective Dynamics of Complex Systems, Binghamton University, State University of New York, Binghamton, NY 13902-6000, USA*
[2]*Department of Systems Science and Industrial Engineering, Binghamton University, State University of New York, Binghamton, NY 13902-6000, USA*
[3]*Bernard M. and Ruth R. Bass Center for Leadership Studies, School of Management, Binghamton University, State University of New York, Binghamton, NY, USA*
[4]*Department of Psychology, University of Oklahoma, Norman, OK, USA*
[5]*U.S. Army Research Institute for the Behavioral and Social Sciences, Fort Belvoir, VA, USA*

Correspondence should be addressed to Yingjun Dong; ydong25@binghamton.edu

Received 15 April 2021; Revised 12 August 2021; Accepted 13 September 2021; Published 26 September 2021

Academic Editor: Carlos M. Travieso-González



Utterance clustering is one of the actively researched topics in audio signal processing and machine learning. This study aims to improve the performance of utterance clustering by processing multichannel (stereo) audio signals. Processed audio signals were generated by combining left- and right-channel audio signals in a few different ways and then by extracting the embedded features (also called *d*-vectors) from those processed audio signals. This study applied the Gaussian mixture model for supervised utterance clustering. In the training phase, a parameter-sharing Gaussian mixture model was obtained to train the model for each speaker. In the testing phase, the speaker with the maximum likelihood was selected as the detected speaker. Results of experiments with real audio recordings of multiperson discussion sessions showed that the proposed method that used multichannel audio signals achieved significantly better performance than a conventional method with mono-audio signals in more complicated conditions.

## 1. Introduction

With artificial intelligence (AI) development, many techniques are applied in our daily life, such as automatic speech recognition (ASR) [1] and speaker recognition. Studies and products in speech processing are widely used in our daily life, such as Apple's Siri, Amazon's Alexa, Google Assistant, and Microsoft's Cortana. As more studies developed in speech processing, it will likely see further increases in popularity. Utterance clustering is a popular topic in speech processing that can be used for speaker diarization [2] and ASR. However, most studies are based on laboratory data sets, and those cannot process the real-world problem very well. Both formal and informal meetings have more segments with overlapping speaking than segments with only one speaker [3]. In the laboratory data sets, people speak one by one, but it is hard to ask people not to interrupt others' speech in the real world. The issue of overlapping speech segments has received considerable attention [4]. To expand the application of speech processing, it is necessary to have better performance in overlapping utterance clustering.

A key aspect of performance improvement in utterance clustering is audio feature embeddings. Feature embedding plays a vital role in ensuring the performance of utterance clustering. There are many studies that focus on the enhancement of audio feature embeddings, such as mel-frequency cepstral coefficient (MFCC) [5], *i*-vector [6], *x*-vector [7], and *d*-vector [8]. However, a significant bottleneck towards the widespread adoption of speech processing applications in daily life is high-quality audio data requirements. Besides, audio signal processing could be a contributing factor to feature embeddings. There are needs

for a better speaker diarization method using low-quality audio recording data in many social science experiments. This study initially tried several different published methods for our own experimental research, but their results were not as good as we had hoped.

To address this problem, here, a new method of audio signal processing was proposed for utterance clustering [9]. The challenge this study aims to address is how to handle low-quality audio data recorded in real-world discussion settings. The audio data set was recorded using an ordinary video camcorder in a noisy environment without a professional microphone. This study contributes to the advance of utterance clustering when the recording conditions are limited.

This study aims to improve clustering performance by processing multichannel (stereo) audio signals. Mono-audio signals are typically used in audio processing studies because they can be obtained easily by downmixing stereo audio signals. Then the *d*-vector of each audio segment was obtained using pretrained neural networks as the audio feature representation.

Gaussian mixture model (GMM) was used as a supervised clustering method. The error rate (ER) of the clustering was compared, and the results showed that using the processed multichannel audio signal for utterance clustering was significantly better than using the original mono-audio signal. The structure of the paper is as follows. Section 2 of this paper introduces some related work. In Section 3, the method in feature processing and the Gaussian mixture model are described. Section 4 shows the details of the data set and the details of the experiments. The results are discussed in Section 5, and conclusion and plans for future work are described in Section 6.

## 2. Related Work

Numerous researchers have made significant advances in utterance clustering and related fields over the last few decades. Certain studies place a greater emphasis on feature embeddings; historically, the most common feature representation was the MFCC [5], which is a method based on the Fourier spectrum. Then, as factor analysis developed, Dehak et al. [6] proposed a factor analysis called *i*-vector. Their factor analysis took into account the variability of speakers and channels without distinction. Lei and Kun [10] proposed wavelet packet entropy (WPE) to extract short vectors from utterances and then used the *i*-vector as a feature embedding. As with *i*-vectors, *d*-vectors [8] also have fixed sizes regardless of the length of the input utterance. Wan et al. [11] trained speakers' utterances using a deep neural network, and the lengths of these utterances varied, resulting in fixed-length embeddings, namely *d*-vector. The distinction between *i*-vector and *d*-vector is that the former is generated using GMM, while the latter is trained using deep neural networks. Similar to the *d*-vector, the *x*-vector [7] is also trained with deep neural networks. Ma et al. [12] proposed an *E*-vector, which was obtained by minimizing Euclidean metric to improve the performance of speaker identification. All of the feature embeddings aforementioned are commonly utilized, and the *d*-vector was used in this study. There are also some works that focus on the improvement of feature extraction. Lin et al. [13] introduced a novel feature extraction approach that combines multiresolution analysis with chaotic feature extraction to improve the performance of utterance features. Daqrouq et al. [14] proposed a feature extraction method based on wavelet packet transform (WPT). They removed the silence parts from the audio data and decomposed the audio signal into wavelet packet tree nodes.

In some research, the clustering algorithms are given greater consideration. Delacourt and Wellekens [15] applied Bayesian information criterion (BIC) to measure the distances among utterances and conducted the agglomerative hierarchical clustering (AHC) based on the BIC metrics. Li et al. [16] conducted GMM on MFCC to classify the speakers' gender. Algabri et al. [17] applied Gaussian mixture model with the universal background model (GMM-UBM) to recognize speakers according to the MFCC of utterances. Shum et al. [18] used the resegmentation algorithm of Bayesian GMM clustering model based on *i*-vector to contribute to improving the speech clustering. Zajíc et al. [19] proposed a model for applying convolutional neural network (CNN) on *i*-vector to detect speaker changes. Wang et al. [20] developed the LSTM model on *d*-vector for the speaker diarization. Zhang et al. [21] constructed a supervised speaker diarization system on the extracted *d*-vector, called unbounded interleaved-state recurrent neural networks (UIS-RNNs).

In comparison to the previous efforts, this study used processed audio signals rather than mono-audio samples. The processed audio signals are derived from multichannel (stereo) audio signals, and the proposed method attempted to preserve more representative audio characteristics.

## 3. Methods

In this section, the proposed method of audio feature processing is discussed. The details of processing multichannel audio features are shown, and the tool which was employed to extract audio feature embeddings is described. Also, the clustering method is presented.

*3.1. Feature Processing.* This study operated the left-channel audio signals and the right-channel audio signals to obtain speech-only audio features in the present work. The details of feature processing are visualized in Figure 1. This example shows that after removing the nonspeech part, the speaker's speaking time is 27 seconds. In this work, 27 seconds of stereo audio were divided into 54 stereo audio segments, each of which is 0.5 seconds in length. After that, mono-audio files were extracted, left-channel audio files and right-channel audio files from the 0.5-second-long stereo audio files. The Python package librosa [22] was used to obtain the left and right audio signals in the time series.

Horizontal stacking of the original left and right audio signals (hstack) and horizontal stacking of the sum and the difference of the left- and right-channel signals (sumdif)

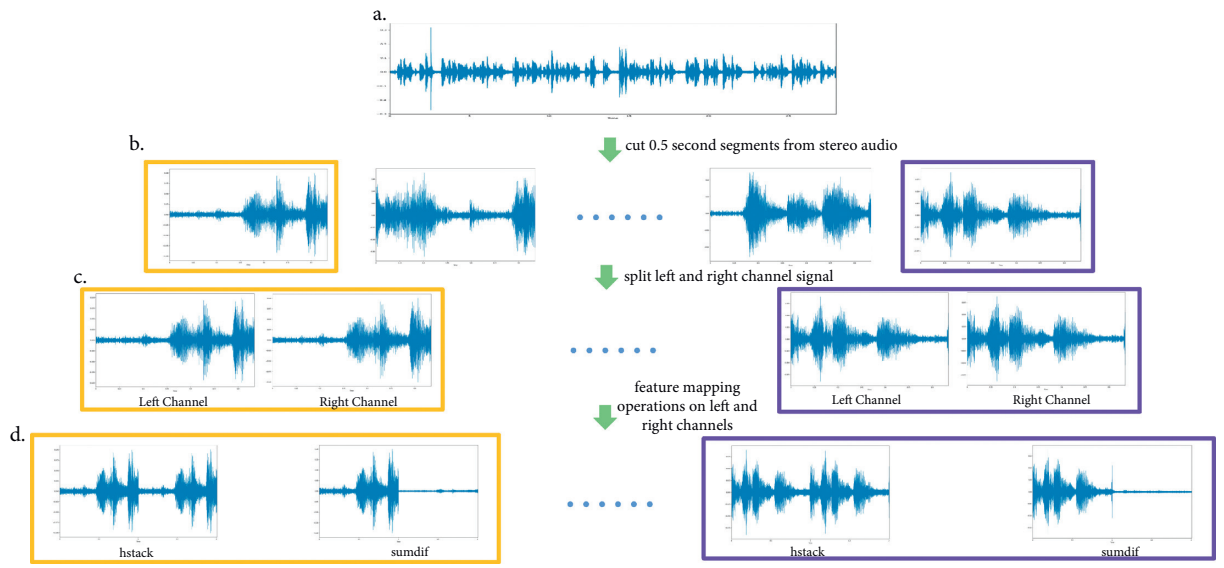


FIGURE 1: Visualization of audio signal processing for each speaker. The same color box represents the waveforms from the same speech segment. (a) A stereo waveform of a speaker's speaking audio, (b) stereo waveforms in 0.5 second, (c) mono waveforms of extracted left- and right-channel audio signal for every 0.5 seconds, and (d) the processed waveforms for every 0.5 seconds.

were performed. The computational complexity of the proposed method is still $O(L)$, where $L$ is the length of audio signal, which is the same in order as the traditional methods (although the actual computation takes about twice as much because our method processes two channels of audio signals).

For the training set, all speakers' utterances $S = (s_1, \ldots, s_i, \ldots, s_N)$ were acquired, where $N$ represents the number of speakers in the audio data set, and $s_i$ represents the sequence of all speaking segments of the $i$th speaker. Specifically, $s_i = (x_{i,1}, \ldots, x_{i,t})$, where $x_{i,t}$ represents the $i$th speaker's audio signal at the $t$th segment. Then left- and right-channel audio signals from each segment in $s_i$ were extracted: $x_{i,t}^L$ for the $i$th speaker's left-channel audio signal at the $t$th segment and $x_{i,t}^R$ for the $i$th speaker's right-channel audio signal at the $t$th segment were obtained. Using the left and right channels, the following two combined audio segments were created: $x_{i,t,\text{hstack}} = (x_{i,t}^L, x_{i,t}^R)$ and $x_{i,t,\text{sumdif}} = (x_{i,t}^L + x_{i,t}^R, x_{i,t}^L - x_{i,t}^R)$. For the $i$th speaker's all audio segments, $s_{i,W} = (x_{i,1,W}, \ldots, x_{i,t,W})$ was obtained, where $W \in \{\text{hstack}, \text{sumdif}\}$. For a fair comparison, a mono stack was created, which is called mstack. It is a stack result of repeated mono signals, represented as $x_{i,t,\text{mstack}} = (x_{i,t}^{\text{mono}}, x_{i,t}^{\text{mono}})$.

*3.2. Feature Embeddings.* After feature processing, the $d$-vector [11] was extracted as the feature representation of the audio signals. The pretrained model called real-time voice cloning [23] was used to extract the $d$-vector. The pretrained model was trained using three data sets: one data set is LibriSpeech ASR corpus [24], which contains 292,000 utterances for more than 2,000 speakers in English, and others are VoxCeleb 1 and 2 [25, 26], which contain more than 1 million utterances for more than 7,000 speakers in multiple languages.

A $d$-vector from each $s_{i,W}$ was extracted, to obtain $D_{i,W} = (d_{i,1,W}, \ldots, d_{i,t,W})$, where $D_{i,W}$ represents $d$-vectors of the $i$th speaker's all audio segments, and $W \in \{\text{mono}, \text{mstack}, \text{hstack}, \text{sumdif}\}$. Then, GMM clustering on the extracted $d$-vectors was conducted.

*3.3. Gaussian Mixture Model.* Gaussian mixture model (GMM) was used as the clustering method. GMM is one of the most frequently used tools for speakers clustering. In this study, separate GMM models for individual speakers were built, defined as

$$p(y) = \sum_{m=1}^{M} \alpha_m \mathcal{N}(y; \mu_m, \Sigma_m), \tag{1}$$

where $y$ represents the feature vector of audio signal, $p(y)$ is the probability that the input audio signal belongs to specific cluster, $\alpha_m$ represents the mixing proportions, $\mu_m$ represents mean, and $\Sigma_m$ represents covariance matrix [27]. The expectation maximization (EM) algorithm [28] was used to estimate the model parameters in GMM. GMM has significant advantages in acoustic modeling [27].

## 4. Experiments

The details of the experiments are described in this section. The details of the data set used in this study are presented first. Then, the tool used in this study to perform audio processing is introduced. Also, the details of clustering experiments are shown. To ensure that the comparison between the proposed method and comparative methods is fair, the same audio data and the same audio processing method to extract multichannel audio signals and mono-audio signals were used in the proposed experiments for both the proposed method and comparative methods. Last

but not least, a parameter sharing GMM was conducted for both the proposed method and the comparative methods.

*4.1. Data set.* A data set [29] containing 11 video files of discussions by multiple participants in a real-world physical environment was used in the proposed work. The number of speakers in 11 videos ranged from 4 to 10, the number of female speakers varies from 1 to 6, the number of male speakers varies from 1 to 6, and all speakers spoke English. Each speaker's speaking time ranged from 1 to 130.5 seconds. The total speaking time for all 11 videos is 31.6 minutes, and the average speaking time for each speaker is 26.7 seconds. The data set was manually annotated with the ground-truth speaker labels.

In the proposed experiments, two comparison groups were set. Audio files in one group contain overlapping speeches, and in the other one, audio files do not contain overlapping speeches. The audio files in these two groups are from the same audio. Speakers were in a real-world free discussion scenario, and an ordinary video camcorder was used to record all videos and audios with a built-in stereo microphone.

*4.2. Audio Processing.* FFmpeg [30] was used to extract stereo audio files and mono-audio files from the video files. Based on the manually annotated speaking time data, audio segments for each of the different speakers were cut. Then, each audio segment of different speakers was cut into shorter segments of a length of 0.5 seconds. The audio files that were shorter than 0.5 seconds were deleted. Then, stereo signals were split into left- and right-channel signals, and $d$-vectors from processed signals were obtained. After the audio signal processing, the clustering experiments were conducted.

*4.3. Clustering by the Gaussian Mixture Model.* The proposed work applied scikit-learn [31] for GMM training and testing. In the initial experiment, a small part of the data set and traditional methods (mono-audio signal) were used to adjust the parameters to obtain better accuracy. Then, for fair comparison, the same parameters were set for all the proposed methods. The full covariance type and $K$-means were used to initialize the model.

The input for the clustering model is the $d$-vector, and clustering experiments were conducted 50 times; for each time, a 10-fold cross-validation test was conducted.

In the training phase, there are $d$-vectors $D_{i,W}^{\text{train}} = (d_{i,1,W}^{\text{train}}, \ldots, d_{i,t_1,W}^{\text{train}})$, where $t_1$ is 70% of speaker $i$'s total speaking time. To train the model, the speakers' label sequence $Y_i = (y_{i,1}, \ldots, y_{i,t_1})$ for speaker $i$. This study trained GMM models for each speaker, and then model set $M = (m_1, \ldots, m_N)$ was obtained, where $m_N$ represents the $N$th speaker's trained model. For the testing, there are $d$-vectors of audio segment for each speaker $D_{i,W}^{\text{test}} = (d_{i,1,W}^{\text{test}}, \ldots, d_{i,t_2,W}^{\text{test}})$, where $t_2$ is 30% of speaker $i$'s total speaking time. For $N$ speakers, $D^{\text{test}} = (D_{1,W}^{\text{test}}, \ldots, D_{N,W}^{\text{test}})$. The elements of the test set $D^{\text{test}}$ were put into GMM to generate the maximum likelihood prediction results Yhat. Then, $Y$^hat was compared with the ground truth $Y_{\text{test}}$ to obtain the error rate.

## 5. Results

This part will show the results of the proposed experiments. The visualization of feature vectors will be displayed to show the results of feature processing, then the results of GMM clustering and the results of significance tests will be shown.

*5.1. Feature Processing.* Figures 2 and 3 show the visualizations of processed feature vectors using t-SNE [32], and the proposed algorithms (hstack and sumdif) show better clustering results. The data points show manifest clusters in the proposed methods. It can be seen from Figures 2 and 3 that in the proposed methods, the data points of Speakers 04, 05, 06, and 07 are clustered more closely. This implies that the $d$-vectors contain more information when the processed multichannel audio signal is used than the one extracted using the mono-audio signal.

This study improves the performance of feature embeddings by processing multichannel audio signals. The proposed method extracts more useful features from the audio signals. Although the improvement is apparent, there are still some differences between the audio with overlap group and the audio without overlap group. Compared with the audio without overlap group, the proposed method enhances the performance of feature embedding in the audio with overlap group. The audio with overlap group is more intricate than the audio without overlap group. Extracting more useful features helps more in the complicated scenario than in the simple scenario.

*5.2. Clustering.* Table 1 shows the comparison of the $z$-scores of GMM error rates in different algorithms. From Table 1, the sumdif algorithm works better than other algorithms in the audio with overlap group. In the audio without overlap group, the mstack algorithm works better. However, hstack and sumdif work better than the mono. The overall performance of hstack and sumdif is better than mono and mstack.

One-way ANOVA tests with Tukey HSD test were performed to determine whether there were differences among the error rates of algorithms compared. Results are shown in Table 2 for both the audio with overlap group and the audio without overlap group. Both groups had statistically very significant difference among the algorithms.

Results of Tukey HSD test are shown in Table 3. Results showed that the proposed algorithms (hstack and sumdif) are significantly different from traditional algorithms (mono and mstack) when the audio signals contained overlaps between speeches. The difference was less clear when the audio signals had no overlaps.

Results of clustering signify that even if the traditional GMM is applied instead of the deep learning model, using the processed audio signals in utterance clustering can achieve a higher-accuracy score than mono-audio signals.

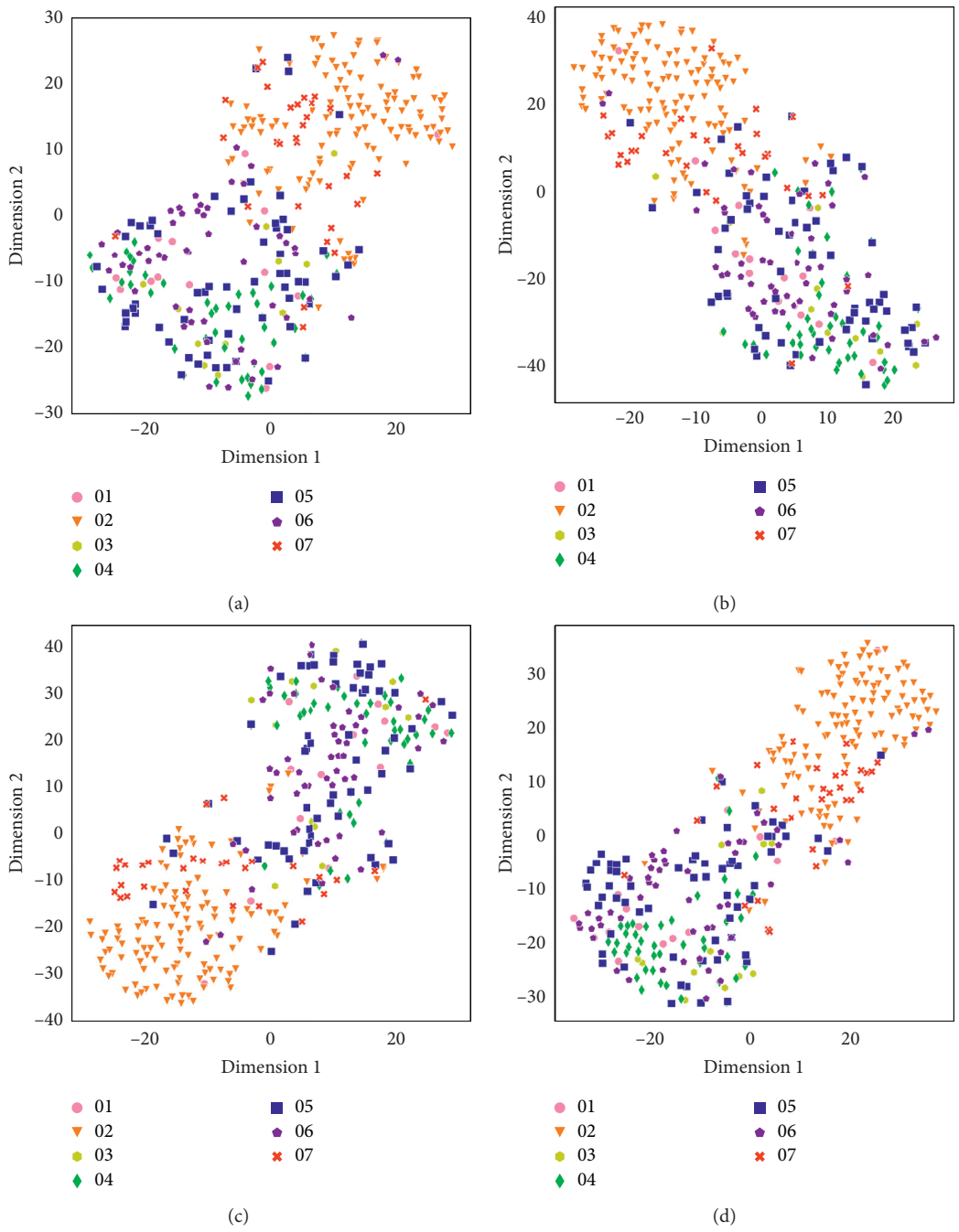


Figure 2: t-SNE visualization for seven speakers' feature vectors in the condition in which audio contains overlapping. Different colors represent different speakers. (a) t-SNE visualization of *d*-vectors' clusters for speakers' mono signals, (b) t-SNE visualization of *d*-vectors' clusters for speakers' mstack processed signals, (c) t-SNE visualization of *d*-vectors' clusters for speakers' hstack processed signals, and (d) t-SNE visualization of *d*-vectors' clusters for speakers' sumdif processed signals.

The data set used in this study represents a real-world discussion setting. The proposed method shows significant improvements in a complicated discussion scenario, and the performance could be further improved by implementing deep learning models. The average of difference in means also shows that compared with simple condition (audio without overlap), the proposed method extracted more features from audio, which is more conducive to the utterance clustering in the complicated scenario (audio with overlap).

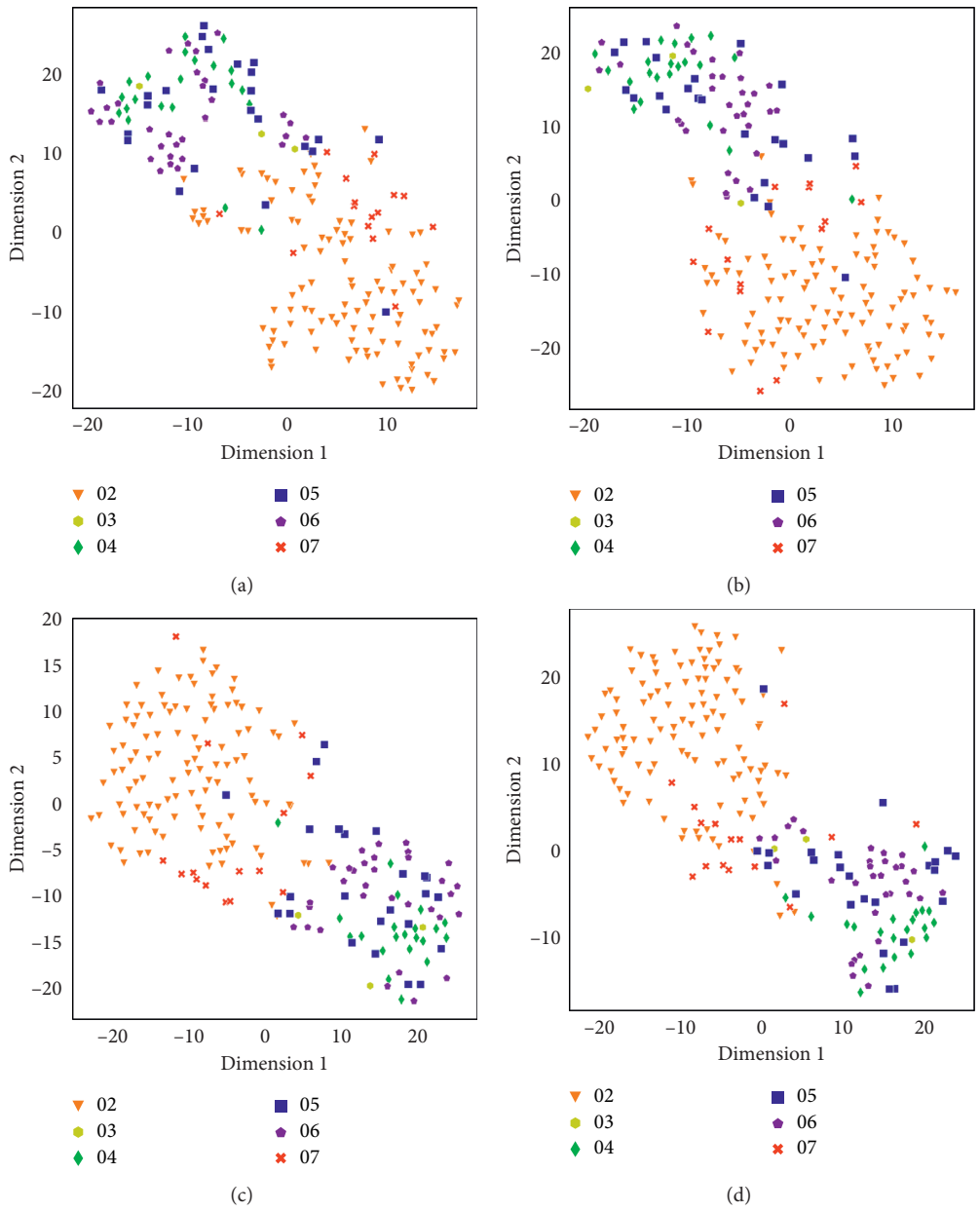


Figure 3: t-SNE visualization for seven speakers' feature vectors in the condition in which audio does not contain overlapping. Different colors represent different speakers. (a) t-SNE visualization of *d*-vectors' clusters for speakers' mono signals, (b) t-SNE visualization of *d*-vectors' clusters for speakers' mstack processed signals, (c) t-SNE visualization of *d*-vectors' clusters for speakers' hstack processed signals, and (d) t-SNE visualization of *d*-vectors' clusters for speakers' sumdif processed signals.

Table 1: Mean *z*-scores of error rates on different methods.

| Data sets | Mono | mstack | hstack | Sumdif |
|---|---|---|---|---|
| With overlap | 0.2520 | 0.1861 | −0.1477 | **−0.2905** |
| Without overlap | 0.2764 | **−0.1419** | −0.0872 | −0.0473 |

Table 2: One-way ANOVA test results

| | *df* | Sum of squares | Mean square | *F* | *p* |
|---|---|---|---|---|---|
| (a) With overlap | | | | | |
| Group | 3.0 | 2.2976 | 0.7659 | 49.1991 | 1.1082*e*−31 |
| Residual | 21,996.0 | 342.3983 | 0.0156 | | |
| (b) Without overlap | | | | | |
| Group | 3.0 | 1.8214 | 0.6071 | 40.7833 | 2.8427*e*−26 |
| Residual | 21,996.0 | 327.4508 | 0.0149 | | |

Table 3: Tukey HSD (FWER = 0.05).

| Group 1 | Group 2 | Meandiff | *p*-adj | Lower | Upper | Reject |
|---|---|---|---|---|---|---|
| (a) With overlap | | | | | | |
| Mono | mstcak | −0.0018 | 0.8611 | −0.0079 | 0.0043 | False |
| Mono | Sumdif | −0.0234 | 0.001 | −0.0295 | −0.0173 | True |
| mstack | Sumdif | −0.0216 | 0.001 | −0.0277 | −0.0155 | True |
| hstack | Mono | 0.0187 | 0.001 | 0.0126 | 0.0248 | True |
| hstack | mstack | 0.0169 | 0.001 | 0.0108 | 0.023 | True |
| hstack | Sumdif | −0.0047 | 0.1973 | −0.0108 | 0.0014 | False |
| (b) Without overlap | | | | | | |
| Mono | mstack | −0.0237 | 0.001 | −0.0297 | −0.0178 | True |
| Mono | Sumdif | −0.0143 | 0.001 | −0.0203 | −0.0084 | True |
| mstack | Sumdif | 0.0094 | 0.001 | 0.0034 | 0.0154 | True |
| hstack | Mono | 0.0205 | 0.001 | 0.0146 | 0.0264 | True |
| hstack | mstack | −0.0033 | 0.4938 | −0.0093 | 0.0027 | False |
| hstack | Sumdif | 0.0061 | 0.0425 | 0.0001 | 0.0121 | True |

Positive/negative meandiff means the average error rate of Group 2 is more/less than the average error rate of Group 1.

## 6. Conclusion

This study generated processed audio signals by combining left- and right-channel audio signals in two different ways. *d*-vectors were extracted as embedded features from those processed audio signals. GMM was conducted for supervised utterance clustering. Based on the results obtained from the supervised clustering experiment, the proposed method works better in complicated conditions than traditional methods. Namely, the proposed method can achieve a higher accuracy score than using traditional algorithms in the speech that contains overlapping. This is because the stereo audio signals contain information about spatial location of the sound source (in a left-right direction space). In a typical real-world discussion setting, speakers tend to sit in a fixed location, so using spatial information can help speaker identification and utterance clustering. This study successfully demonstrated this idea.

One limitation of the proposed method is the computational cost. Even though the theoretical computational complexity of the proposed method is the same as the traditional methods, in the actual experiments, the run time of our proposed method is greater than that of the traditional methods. Moreover, stereo audio signals were used in this study, so another limitation is that the input data must be multichannel audio signals that involve spatial information.

In this study, GMM was applied as a clustering method. An innovative clustering model using deep learning will be proposed for future works. After applying different clustering methods, there are more comprehensive comparisons between the proposed algorithms and traditional algorithms.

## Data Availability

The audio data used to support the findings of this study have not been made available because Institutional Review Board permissions do not accommodate their release.

## Disclosure

A preprint version of this work is also available from https://arxiv.org/abs/2009.05076. The views expressed in this presentation are those of the authors and do not reflect the official policy or position of the Department of the Army, DOD, or the U.S. Government.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This research was sponsored by the U.S. Army Research Institute for the Behavioral and Social Sciences, Department of the Army (grant no. W911NF-17-1-0221).

## References


[1] T. Menne, I. Sklyar, R. Schlüter, and H. Ney, "Analysis of deep clustering as preprocessing for automatic speech recognition of sparsely overlapping speech," in *Proceedings of the Interspeech*, Graz, Austria, 2019.

[2] X. Anguera Miro, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: a review of recent research," *IEEE Transactions on Audio Speech and Language Processing*, vol. 20, no. 2, pp. 356–370, 2012.

[3] T. von Neumann, K. Kinoshita, M. Delcroix, S. Araki, T. Nakatani, and R. Haeb-Umbach, "All-neural online source separation, counting, and diarization for meeting analysis," in *Proceedings of the ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 91–95, IEEE, Brighton, UK, 2019.

[4] Z. Chen, X. Xiao, T. Yoshioka, H. Erdogan, J. Li, and Y. Gong, "Multi-channel overlapped speech recognition with location guided speech extraction network," in *Proceedings of the 2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 558–565, IEEE, Athens, Greece, 2018.

[5] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[6] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio Speech and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[7] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: robust DNN embeddings for speaker recognition," in *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5329–5333, IEEE, Calgary, Canada, 2018.

[8] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4052–4056, IEEE, Florence, Italy, 2014.

[9] Y. Dong, N. G. MacLaren, Y. Cao et al., "Utterance clustering using stereo audio channels," 2021, http://arxiv.org/abs/2009.05076.

[10] L. Lei and S. Kun, "Speaker recognition using wavelet packet entropy, i-vector, and cosine distance scoring," *Journal of Electrical and Computer Engineering*, vol. 2017, Article ID 1735698, 9 pages, 2017.

[11] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4879–4883, IEEE, Calgary, Canada, 2018.

[12] H. Ma, Y. Zuo, T. Li, and C. L. P. Chen, "Data-driven decision-support system for speaker identification using e-vector system," *Scientific Programming*, vol. 2020, Article ID 4748606, 13 pages, 2020.

[13] J. Lin, Y. Yumei, Z. Maosheng, C. Defeng, W. Chao, and W. Tonghan, "A multiscale chaotic feature extraction method for speaker recognition," *Complexity*, vol. 2020, Article ID 8810901, 9 pages, 2020.

[14] K. Daqrouq, R. Al-Hmouz, A. S. Balamash, N. Alotaibi, and E. Noeth, "An investigation of wavelet average framing LPC for noisy speaker identification environment," *Mathematical Problems in Engineering*, vol. 2015, Article ID 598610, 10 pages, 2015.

[15] P. Delacourt and C. J. Wellekens, "Distbic: a speaker-based segmentation for audio data indexing," *Speech Communication*, vol. 32, no. 1-2, pp. 111–126, 2000.

[16] D. Li, Y. Yang, and W. Dai, "Cost-sensitive learning for emotion robust speaker recognition," *Science World Journal*, vol. 2014, Article ID 628516, 9 pages, 2014.

[17] M. Algabri, H. Mathkour, M. A. Bencherif, M. Alsulaiman, and M. A. Mekhtiche, "Automatic speaker recognition for mobile forensic applications," *Mobile Information Systems*, vol. 2017, Article ID 6986391, 6 pages, 2017.

[18] 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.

[19] Z. Zajíc, M. Hrúz, and L. Müller, "Speaker diarization using convolutional neural network for statistics accumulation refinement," in *Proceedings of the INTERSPEECH*, pp. 3562–3566, Stockholm, Sweden, 2017.

[20] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, "Speaker diarization with LSTM,," in *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5239–5243, IEEE, Calgary, Canada, 2018.

[21] A. Zhang, Q. Wang, Z. Zhu, J. Paisley, and C. Wang, "Fully supervised speaker diarization," in *Proceedings of the ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6301–6305, IEEE, Brighton, UK, 2019.

[22] B. McFee, C. Raffel, D. Liang et al., "librosa: audio and music signal analysis in python," in *Proceedings of the 14th Python in Science Conference*, pp. 18–25, Austin, TX, USA, 2015.

[23] C. Jemine, "*Real-time-voice-cloning*," Master's thesis, https://github.com/CorentinJ/Real-Time-Voice-Cloning, University of Liége, Liége, Belgium, 2019.

[24] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books,," in *Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, IEEE, Brisbane, Australia, 2015.

[25] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *Proceedings of the INTERSPEECH*, Stockholm, Sweden, 2017.

[26] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: deep speaker recognition," in *Proceedings of the INTERSPEECH*, Hyderabad, India, 2018.

[27] D. Yu and L. Deng, *Automatic Speech Recognition*, Springer, Berlin, Germany, 2016.

[28] J. A. Bilmes, "A gentle tutorial of the em algorithm and its application to parameter estimation for Gaussian mixture and hidden markov models," *International Computer Science Institute*, vol. 4, no. 510, p. 126, 1998.

[29] N. G. MacLaren, F. J. Yammarino, S. D. Dionne et al., "Testing the babble hypothesis: speaking time predicts leader emergence in small groups," *The Leadership Quarterly*, vol. 31, no. 5, Article ID 101409, 2020.

[30] FFmpeg Developers, FFmpeg Tool, https://ffmpeg.org/.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort et al., "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[32] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.