LEAP Submission for Third DIHARD Diarization Challenge

Prachi Singh
Learning and Extraction of Acoustic Patterns (LEAP) Lab
Electrical Engineering
Indian Institute of Science
Bangalore, India
prachisingh@iisc.ac.in

Rajat Varma
LEAP Lab
Electrical Engineering
Indian Institute of Science
Bangalore, India
rajatvarma@iisc.ac.in

Venkat Krishnamohan
LEAP Lab
Electrical Engineering
Indian Institute of Science
Bangalore, India
venkat201097@gmail.com

Srikanth Raj Chetupalli
LEAP Lab
Electrical Engineering
Indian Institute of Science
Bangalore, India
sraj@iisc.ac.in

Sriram Ganapathy
LEAP Lab
Electrical Engineering
Indian Institute of Science
Bangalore, India
sriramg@iisc.ac.in

***Abstract*—The LEAP submission for DIHARD-III challenge is described in this report. The LEAP system involves the use of End-to-End speaker diarization system for the two-speaker conversational telephone speech recordings. For the wideband multi-speaker recordings, the proposed approach for diarization uses embeddings from a time-delay neural network (called x-vectors) followed by a graph based clustering approach called the path integral clustering. The LEAP system showed 24% and 18% relative improvements for track1 and track2 respectively over the baseline system provided by the organizers. This report provides details of the model and the experimental results on the DIHARD-III dataset.**

***Index Terms*—speaker diarization, End-to-End system, x-vectors, path integral clustering**

I. Notable Highlights

The DIHARD dataset has a mix of narrowband and wideband speech recordings. In the development set, 24% of the recordings are narrowband from CTS domain, and the remaining have wideband speech. The narrowband recordings have only two speakers in each recording. We use a combination of models optimized separately for narrowband and wideband speech, in combination with a bandwidth classifier, to design the diarization system. For wideband speech, the pipeline consists of an embedding extractor based on extended time delay neural network (ETDNN), a graph based clustering scheme called path integral clustering (PIC) and variational Bayes - hidden Markov model (VB-HMM) re-segmentation. In addition, we also use an overlap detection based on a separate overlap detection model which is combined with the VB-HMM diarization system. For narrowband speech, we explore the supervised End-to-End model architecture [1], with known (two) number of speakers.

II. Data Resources

The following datasets are used for the two different system configurations.

A. Wideband system datasets

- VoxCeleb1 [2]: It contains over 100,000 utterances from 1,251 celebrities, extracted from videos uploaded to YouTube. The dataset is gender balanced, with 55% of the speakers male. The speakers span a wide range of different ethnicities, accents, professions and ages.
  Link to dataset: http://www.robots.ox.ac.uk/~vgg/data/voxceleb
- Voxceleb2 [3]: It contains over 1 million utterances from over 6,000 celebrities, extracted from videos uploaded to YouTube. The dataset is fairly gender balanced, with 61% of the speakers male. The speakers span a wide range of different ethnicities, accents, professions and ages. Videos included in the dataset are shot in a large number of challenging visual and auditory environments.
  Link to dataset: http://www.robots.ox.ac.uk/~vgg/data/voxceleb2

B. Narrowband system datasets

- Telephone recordings: Switchboard-2 (Phase I, II, III), Switchboard Cellular (Part 1, Part 2) and NIST SRE datasets 2004-2008.
- CALLHOME (CH) : It is a collection of multi-lingual telephone call recordings sampled at 8kHz, containing 500 recordings. The duration of each recording ranges from 2-5 minutes. The number of speakers in each recording varies from 2 to 7, and majority of the files have two speakers. The CH dataset is divided equally into two

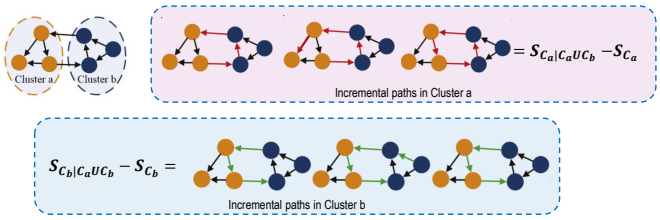

Fig. 1. $Cluster\ a$ and $Cluster\ b$ are represented in orange and blue colours respectively. First rectangular block highlights incremental paths of $Cluster\ a$ in magenta colour after merging $cluster\ a$ and $Cluster\ b$. Similarly second block highlights incremental paths of $Cluster\ b$ in green colour.

different sets, CH1 and CH2, with similar distribution of number of speakers.

III. Detailed description of the systems

A. Baseline system

The baseline systems for both the tracks are implemented as described in [4]. We have used the best system configuration obtained for track1 along with the pre-trained baseline SAD model for track2.

B. Wideband-Narrowband classifier

A two layer fully-connected feed-forward neural network with x-vectors as input features is used as the wideband-narrowband classifier. The input x-vectors are 512-dimensional and extracted every 5 s using segments of duration 10 s. During evaluation, majority voting of the segment-wise prediction of the classifier neural network is used to decide on the bandwidth of the recording.

C. Wideband PIC system

The diarization system for wideband speech recordings is inspired by the multi-stage baseline system, which consists of neural embedding extraction, followed by pair-wise similarity scoring and clustering, of short speech segments, and VB-HMM based re-segmentation with overlap processing. In our setup, we have explored two different models for embedding extraction as described below.

1) Embedding extraction and Scoring: We use x-vectors as the embeddings. The embeddings are extracted using two variants (i) extended-TDNN (ETDNN), and (ii) factorized-TDNN (FTDNN). For training both models we used 30-d mel frequency cepstral coefficient (MFCC) features extracted every 10 ms using a 25 ms window. Datasets used are also common to both models.

ETDNN: The 13-layer ETDNN model follows the architecture described in [5]. ETDNN model is trained on the VoxCeleb1 and VoxCeleb2 datasets, for speaker identification task, to discriminate among $7,232$ speakers. It has 9 TDNN layers before pooling layer which provides temporal context of ± 11 neighbouring frames. The 512 dimensional output of the affine component of the 11^{th} layer is taken as the x-vector embedding. We extract the embeddings using a segment size of 1.5 s and a temporal shift of 0.25 s.

FTDNN: The architecture of the 14-layer FTDNN model is similar to that of ETDNN, with factorized TDNN layers [6] in place of the TDNN layers.The model is trained in a manner similar to ETDNN. It reduces the number of parameters of the network by factorizing the weight matrix of each TDNN layer into a product of two low-rank matrices. The network has a larger temporal context of ± 16 frames.

We extract 512-dimensional output from the 12^th affine layer of the model as the x-vector embedding, using the same resolution and segment-size as x-vectors from ETDNN.

We consider (i) cosine score, and (ii) PLDA score, to compute the similarity between segments. A separate probabilistic linear discriminant analysis (PLDA) model is trained for both ETDNN and FTDNN x-vectors. The similarity score between two segments is then obtained using a binary hypothesis testing framework. To compute the cosine score, the x-vectors are projected to a 30 dimensional space using principle component analysis (PCA), computed using the development dataset, and the score is computed in the projected space.

2) Path integral clustering: We perform clustering of PLDA/Cosine scores to get the diarization output. We have explored a graph-structural based agglomerative clustering algorithm known as path integral clustering (PIC) [7].The clustering process involves creation of a directed graph $G = (V, E)$ where input features are the vertices (V) and E is the set of edges connecting the vertices. Similar to the agglomerative hierarchical clustering (AHC), PIC also merges two clusters at each time step based on maximum affinity, but the affinity is computed using the path integral as defined in [7]. Let $\boldsymbol{X} = \{x_1, x_2, ..., x_{N_r}\}$ be the embeddings extracted from a recording r, where N_r is the total number of embeddings present. Steps involved in the clustering process are described below:

1) Computing adjacency matrix $\boldsymbol{W} \in \mathbb{R}^{N_r \times N_r}$ of graph $G = (V, E)$ defined as,

$$[W]_{ij} = \begin{cases} s(i,j), & \text{if } \boldsymbol{y}_j \epsilon N_i^K. \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

where, N_i^K is the set of K-nearest neighbors of $\boldsymbol{x}_i$. $s(i,j)$ is the pairwise similarity score between x-vectors of (i,j)th segments of a recording. We use sigmoid of PLDA/Cosine similarity scores to ensure positive edge weights. The transition probability matrix $\boldsymbol{P}$ is obtained from the adjacency matrix $\boldsymbol{W}$ by normalizing each row with its sum.

2) Cluster initialization: we group the x-vectors $\boldsymbol{X}$ with nearest neighbour to form $N_r/2$ clusters. We further merge the clusters which have common elements, and use it as the initialization for PIC.

3) Path integral: It is used as the structural descriptor of a cluster. We define path integral $\mathcal{S}_{\mathcal{C}_a}$ of the cluster $\mathcal{C}_a$ as the weighted sum of probabilities of all possible paths from any vertex i to any other vertex j, where

i, j vertices belong to the cluster $\mathcal{C}_a$ and all the vertices along the path also belong to cluster $\mathcal{C}_a$.
We define the conditional path integral $\mathcal{S}_{\mathcal{C}_a|\mathcal{C}_a\cup\mathcal{C}_b}$ as the path integral of all paths in $\mathcal{C}_a \cup \mathcal{C}_b$ such that the paths start and end with vertices belonging to $\mathcal{C}_a$.
As shown in [7], the normalized path integral and the normalized conditional path integral is given as,

$$\mathcal{S}_{\mathcal{C}_a} = \frac{1}{|\mathcal{C}_a|^2}\mathbf{1}^T\left(\boldsymbol{I} - z\boldsymbol{P}_{\mathcal{C}_a}\right)^{-1}\mathbf{1} \quad (2)$$

$$\mathcal{S}_{\mathcal{C}_a|\mathcal{C}_a\cup\mathcal{C}_b} = \frac{1}{|\mathcal{C}_a|^2}\mathbf{1}_{\mathcal{C}_a}^T\left(\boldsymbol{I} - z\boldsymbol{P}_{\mathcal{C}_a\cup\mathcal{C}_b}\right)^{-1}\mathbf{1}_{\mathcal{C}_a} \quad (3)$$

where, $\boldsymbol{P}_{\mathcal{C}_a}$ and $\boldsymbol{P}_{\mathcal{C}_a\cup\mathcal{C}_b}$ are the sub-matrices of the transition probability matrix $\boldsymbol{P}$ obtained by selecting vertices that belong to $\mathcal{C}_a$ and $\mathcal{C}_a \cup \mathcal{C}_b$ respectively. Here, $|\mathcal{C}_a|$ denotes the cardinality (# of vertices) of cluster $\mathcal{C}_a$, $\mathbf{1}$ is a column vector of all ones of size $|\mathcal{C}_a|$ and $\mathbf{1}_{\mathcal{C}_a}$ is a binary column vector of size $|\mathcal{C}_a \cup \mathcal{C}_b|$ with ones at all locations corresponding to the vertices of $\mathcal{C}_a$ and zeros at all locations corresponding to the vertices of $\mathcal{C}_b$. Parameter $0 < z < 1$ ensures higher weightage to shorter paths as compared to longer paths. Note that the identity matrix $\boldsymbol{I}$ used in Equation (2) and (3) are of dimensions $|\mathcal{C}_a|$ and $|\mathcal{C}_a \cup \mathcal{C}_b|$ respectively.

4) Cluster merging: The cluster affinity measure for the PIC algorithm is computed as,

$$\mathcal{A}\left(\mathcal{C}_a, \mathcal{C}_b\right) = \mathcal{S}_{\mathcal{C}_a|\mathcal{C}_a\cup\mathcal{C}_b} - \mathcal{S}_{\mathcal{C}_a} + \mathcal{S}_{\mathcal{C}_b|\mathcal{C}_a\cup\mathcal{C}_b} - \mathcal{S}_{\mathcal{C}_b} \quad (4)$$

where, $\mathcal{S}_{\mathcal{C}_a|\mathcal{C}_a\cup\mathcal{C}_b} - \mathcal{S}_{\mathcal{C}_a}$ is the incremental path integral of $\mathcal{C}_a$. It is illustrated in Figure 1. The clusters to be merged at each step are determined using,

$$\{\mathcal{C}_a, \mathcal{C}_b\} = \underset{C_i, C_j \in \mathbf{C}, i\neq j}{\operatorname{argmax}} \mathcal{A}\left(\mathcal{C}_i, \mathcal{C}_j\right) \quad (5)$$

We perform merging based on above criterion till we reach the required number of speakers. In our approach, clusters are treated as dynamical systems. Higher path integral indicates high stability. Therefore, the algorithm encourages merging towards higher stability by maximising the affinity.

3) VB resegmentation and Overlap detection (VB-overlap): For further refinement of segment boundaries, we apply VB-HMM resegmentation [8] with posterior scaling [9] as described in baseline [4].

For overlap detection, we use the overlap detection module available in the pyannote.audio python toolkit [10]. The architecture of the neural overlap detection model is described in [11]; it consists of sincNet filter layers followed by recurrent and fully-connected layers. We use the pre-trained network, trained on DIHARD-I dataset, to compute the frame-level overlap scores. The segments identified as overlap by the detector are then used to refine the segments obtained after VB-HMM re-segmentation, similar to approach described in [11].

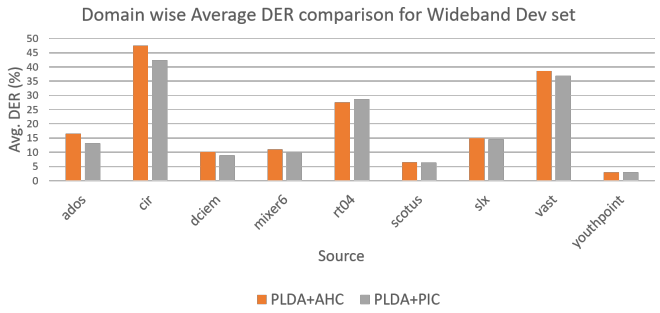

Fig. 2. Performance comparison of AHC and PIC algorithms across domains in Dihard dev wideband set.

TABLE I
TRACK1 DER (JER) OF INDIVIDUAL AND FUSED SYSTEMS. WPS IS WIDEBAND PIC SYSTEM AND NES IS NARROWBAND END-TO-END SYSTEM.

Individual System	Set	Dev DER(JER)	Eval DER(JER)
Baseline [4]	full	19.10 (41.10)	19.68 (44.32)
	core	19.97 (45.52)	21.35 (48.89)
WPS (ETDNN)+NES	full	14.45 (37.09)	14.93 (37.09)
	core	16.43 (42.45)	18.2 (43.28)
WPS (FTDNN)+NES	full	14.34 (37.31)	14.88 (36.73)
	core	16.26 (42.75)	18.07 (42.82)

D. Narrowband End-to-End system

The architecture of the model is similar to SA-EEND [12] combined with the encoder-decoder based attractor calculation (EDA) [1]. The model uses 4 stacked Transformer encoder blocks; each block consists of 256 attention units with 4 attention heads.

23-dimensional log-Mel-filterbank features, extracted every 10 ms using a frame length of 25 ms are used as input features, similar to [1]. A context of ± 7 frames is applied, and the resulting 345 dimensional vectors are sub-sampled by a factor of 10 and used as input to the SA-EEND+EDA model.

Training: We simulated 1,00,000 two-speaker mixtures to train the Narrowband End-to-End system from Switchboard-2 (Phase I, II, III), Switchboard Cellular (Part 1, Part 2) and NIST SRE datasets 2004-2008, using the algorithm proposed in [13]. The model was trained on the simulated mixtures for 100 epochs using utterance-level permutation-invariant training (PIT) [14] criterion. This was followed by model adaptation on CALLHOME dataset two-speaker files.

Evaluation: For the narrowband audio files, we hard-set the number of attractors to be generated to 2 and obtained the frame wise posteriors. A threshold was applied on the posteriors to detect the presence of the speakers. If for any frame the model failed , we assign the speaker with the maximum posterior in that frame. The silence frames are then removed based on ground truth SAD for track1 and pre-trained model SAD for track2.

IV. EXPERIMENTS & RESULTS

We have applied different strategies for wideband and narrowband subset of DIHARD Dev set. For wideband, we

TABLE II
TRACK2 DER (JER) OF INDIVIDUAL AND FUSED SYSTEMS. WPS IS WIDEBAND PIC SYSTEM AND NES IS NARROWBAND END-TO-END SYSTEM.

Individual System	Set	Dev DER (JER)	Eval DER (JER)
Baseline [4]	full	21.35 (42.97)	25.76 (47.64)
	core	22.31 (47.28)	28.31 (52.44)
WPS(ETDNN)+NES	full	16.77 (37.15)	21.04 (39.68)
	core	18.64 (41.93)	24.92 (45.32)
WPS(FTDNN)+NES	full	16.53 (38.50)	21.09 (39.54)
	core	18.34 (43.62)	24.99 (45.13)

TABLE III
DER(JER) PERFORMANCE FOR WIDEBAND SYSTEM CONFIGURATIONS USING ETDNN X-VECTOR MODEL AND FOR NARROWBAND USING SA-EEND MODEL INDICATING THE IMPROVEMENTS FROM THE PROPOSED APPROACHES FOR TRACK1. * INDICATES BASELINE WITH ORACLE NUMBER OF SPEAKERS.

Wideband System config.	Dev DER(JER)
PLDA+AHC (S1)	20.09 (43.86)
PLDA + PIC (S2)	**19.06** (42.44)
Cosine+ PIC	19.78 (43.61)
S1+VB-overlap	17.70 (42.93)
S2+VB-overlap	17.03 (41.92)
Narrowband System config.	**Dev DER(JER)**
Baseline w Oracle*	16.03 (20.21)
SA-EEND V1	9.84 (12.00)
SA-EEND V2	9.34 (11.19)

experiment with different scoring and clustering techniques. Table III shows DER(JER) performance of wideband system using ETDNN x-vectors. PLDA+AHC is the baseline approach. As discussed in Section III-C2, we have implemented PLDA and Cosine with path integral clustering (PIC). From the table, we can see that PLDA with PIC algorithm gives 1% absolute improvement compared to PLDA with AHC algorithm. The stopping criteria of PIC is based on number of speakers predicted by PLDA+AHC threshold obtained after fine tuning on Dev set. Comparison of PIC and AHC is done in Figure 2. It shows domain wise average DER for PLDA-AHC and PLDA-PIC using the ETDNN embedding extractor. The plot shows improvement of PIC algorithm over AHC in 7 out of 9 domains in wideband. We further improved the DER and JER using VB-HMM refinement along with overlap detection as described Section III-C3. The **miss-rate and false-alarm** of the overlap-detection module for DIHARD dev set are **17.1% and 5.0 %** respectively.

For narrowband system, we use SA-EEND system with attractor. For evaluation, we subsample the frame-level features by different factor to avoid abrupt speaker change and to make it more memory efficient. Table III also shows results on narrowband recordings from cts domain. SA-EEND V1 involves subsampling by 10 and whereas SA-EEND V2 involves subsampling by 5. We see a significant improvement of 40% (relative). As we increase the resolution by reducing subsampling factor, we see marginal improvement.

Table I and II shows results of our systems along with baseline for track1 and track2 respectively. We have combined wideband PIC system (WPS) with ETDNN/FTDNN for wideband domain files and narrowband end-to-end system (NES) for narrowband (cts domain) files. We also explored weighted combination of PLDA scores from ETDNN and FTDNN models for wideband PIC system (WPS) which only gave marginal improvement.

V. HARDWARE REQUIREMENTS

The hardware requirements reported were common for both, the training and testing phase.

A. CPU information

Model - Acer F380 series
Number of cores - 64
Memory - 256 GB

B. GPU information

Model - NVIDIA Quadro P5000
CUDA cores - 2560
GPU memory - 16 GB
GPU used - 2

C. Toolkits used

For training and extraction of x-vectors, PLDA training, and resegmentation we used Kaldi [15] framework. The data were prepared according to the data formats required by kaldi. Python 3.7 is used for similarity scoring and clustering. For the narrowband system we use the Chainer framework. Other toolkits used are librosa, soundfile, and pyannote python libraries.

D. Execution times

The wall clock time corresponding to the execution time is reported here. For ETDNN and FTDNN systems training requires 72 hours and 144 hours respectively. PLDA training requires 30 minutes. For testing (all wideband eval files) which involves extraction of features, embeddings, clustering and resegmentation, took around 30 minutes. Narrowband end-to-end system training was completed in 96 hours.

REFERENCES

[1] S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, and K. Nagamatsu, "End-to-End Speaker Diarization for an Unknown Number of Speakers with Encoder-Decoder Based Attractors," in *Interspeech*, 2020.

[2] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," *Proc. of Interspeech*, pp. 2616–2620, 2017.

[3] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Proc. of Interspeech*, 2018, pp. 1086–1090.

[4] N. Ryant, P. Singh, V. Krishnamohan, R. Varma, K. Church, C. Cieri, J. Du, S. Ganapathy, and M. Liberman, "The Third DIHARD Diarization Challenge," 2020.

[5] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5796–5800.

[6] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks," in *Proc. Interspeech 2018*, 2018, pp. 3743–3747. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1417

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

[8] M. Diez, L. Burget, and P. Matejka, "Speaker diarization based on bayesian hmm with eigenvoice priors," in *Proceedings of Odyssey*, 2018, pp. 147–154.

[9] P. Singh, H. Vardhan, S. Ganapathy, and A. Kanagasundaram, "LEAP diarization system for the second dihard challenge," in *Proc. of INTERSPEECH*, 2019, pp. 983–987.

[10] H. Bredin, R. Yin, J. M. Coria, G. Gelly, P. Korshunov, M. Lavechin, D. Fustes, H. Titeux, W. Bouaziz, and M.-P. Gill, "pyannote.audio: neural building blocks for speaker diarization," in *ICASSP 2020, IEEE International Conference on Acoustics, Speech, and Signal Processing*, Barcelona, Spain, May 2020.

[11] L. Bullock, H. Bredin, and L. P. Garcia-Perera, "Overlap-aware diarization: Resegmentation using neural end-to-end overlapped speech detection," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7114–7118.

[12] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-End Neural Speaker Diarization with Self-attention," in *ASRU*, 2019, pp. 296–303.

[13] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-End Neural Speaker Diarization with Permutation-free Objectives," in *Interspeech*, 2019, pp. 4300–4304.

[14] M. Kolbæk, D. Yu, Z. Tan, and J. Jensen, "Multi-talker Speech Separation with Utterance-level Permutation Invariant Training of Deep Recurrent Neural Networks," in *IEEE/ACM Trans. on ASLP*, vol. 25, no. 10, 2017, pp. 1901–1913.

[15] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," IEEE Signal Processing Society, Tech. Rep., 2011.