The Hitachi-JHU DIHARD III System: Competitive End-to-End Neural Diarization and X-Vector Clustering Systems Combined by DOVER-Lap

Shota Horiguchi[1] Nelson Yalta[1] Paola García[2] Yuki Takashima[1] Yawen Xue[1]
Desh Raj[2] Zili Huang[2] Yusuke Fujita[1] Shinji Watanabe[2] Sanjeev Khudanpur[2]
[1]*Hitachi, Ltd. Research & Development Group, Tokyo, Japan*
[2]*Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, USA*
{shota.horiguchi.wk, nelson.yalta.wm}@hitachi.com, lgarci27@jhu.edu

***Abstract*—This paper provides a detailed description of the Hitachi-JHU system that was submitted to the Third DIHARD Speech Diarization Challenge. The system outputs the ensemble results of the five subsystems: two x-vector-based subsystems, two end-to-end neural diarization-based subsystems, and one hybrid subsystem. We refine each system and all five subsystems become competitive and complementary. After the DOVER-Lap based system combination, it achieved diarization error rates of 11.58% and 14.09% in Track 1 full and core, and 16.94% and 20.01% in Track 2 full and core, respectively. With their results, we won second place in all the tasks of the challenge.**

***Index Terms*—speaker diarization, x-vector, VBx, EEND, DOVER-Lap**

I. Notable highlights

This technical report describes the Hitachi-JHU system submitted to the Third DIHARD Speech Diarization Challenge [1]. We mainly focused our efforts on how we can pick the best of diarization based on x-vector clustering and end-to-end neural speaker diarization (EEND). The highlights of our systems are as follows:

- Two x-vector-based systems incorporated with VBx clustering and heuristic overlap assignment. One is based on a time-delay neural network (TDNN) based x-vector extractor following the winning system of the DIHARD II Challenge [2], [3]. The other is based on Res2Net-based x-vector extractors, which won the VoxCeleb Speaker Recognition Challenge 2020 [4].
- Two EEND-based subsystems, each of which is the extension of the original self-attentive EEND [5] to output diarization results of a variable number of speakers, with improved inference.
- A hybrid subsystem of x-vector clustering and EEND, in which update the results of x-vector clustering using EEND as post-processing [6].
- Modified DOVER-Lap [7] to combine the results from five subsystems above.
- Self-supervised adaptation of the EEND model.

II. Data resources

Table I summarizes the corpora we used to train our models which compose our diarization system. We briefly explain each corpus below.

- DIHARD III: focused on "hard" speaker diarization, contains 5-10 minute utterances selected from 11 conversational domains, each including approximately 2 hours of audio [1].
- VoxCeleb 1: a large-scale speaker identification dataset with 1,251 speakers and over 100,000 utterances, collected "in the wild" [8].
- VoxCeleb 2: a speaker recognition dataset that contains over a million utterances from over 6,000 speakers under noisy and unconstrained conditions [8].
- Switchboard-2 (Phase I, II, III), Switchboard Cellular (Part 1, 2): English telephone conversation datasets. Their LDC catalog numbers are LDC98S75, LDC99S79, LDC2002S06, LDC2001S13, and LDC2004S07, respectively.
- NIST Speaker Recognition Evaluation (2004, 2005, 2006, 2008): also telephone conversations but not limited to English, which are composed of the following LDC corpora: LDC2006S44, LDC2011S01, LDC2011S04, LDC2011S09, LDC2011S10, LDC2012S01, LDC2011S05, LDC2011S08.
- MUSAN: publicly available corpus that consists of music, speech, and noise [9]. The music and noise portions are sometimes used for data augmentation.

III. Detailed description of algorithm

A. Voice Activity Detector

We employed two voice activity detectors (VAD): SincNet-based VAD [10] and TDNN-based VAD.

SincNet-based VAD: Our SincNet-based VAD is implemented using the pyannote [11] framework. This VAD model learns to detect speech from the raw speech using a combination of a SincNet [12] followed by BiLSTM layers and fully connected layers. For our experiments, we employed the default configuration provided by pyannote: a SincNet with 80 channels and 251 dims of kernel size, two BiLSTM layers with 128 cell dimensions, and two fully connected layers of

TABLE I: Corpora we used to train the models in our system.

	VAD		X-vector extractor					
Corpus	SincNet	TDNN	TDNN	Res2Net	PLDA	Overlap detector	EEND-EDA	SC-EEND
DIHARD III development set [1]	✓	✓			✓	✓	✓	✓
DIHARD III evaluation set [1] (with pseudo labels)							✓	
VoxCeleb 1 [8]			✓	✓				
VoxCeleb 2 [8]			✓	✓				
Switchboard-2 Phase I, II, III							✓	✓
Switchboard Cellular Part 1, 2							✓	✓
NIST Speaker Recognition Evaluation 2004, 2005, 2006, 2008							✓	✓
MUSAN corpus		✓	✓	✓			✓	✓

TABLE II: VAD performance on the DIHARD III development set.

Method	False alarm (%)	Missed speech (%)
SincNet-based VAD	2.78	2.51
TDNN-based VAD	2.85	2.80
Posterior average	2.58	2.55

128 dimensions. We trained the model using the DIHARD III development set for 300 epochs.

TDNN-based VAD: Our TDNN VAD is based on the example from Kaldi [13] recipe. The acoustic feature we use is 40-dim MFCC, and the left and right 2 frames are appended to generate the 200-dim input features. The model first transforms the input features with linear discriminant analysis (LDA) estimated with the VAD labels. Then the transformed features pass through five TDNN blocks. Each TDNN block consists of a TDNN layer, a Rectified Linear Unit (ReLU), and a batch normalization layer. In the last two TDNN blocks, to capture long temporal contexts, the mean vector for neighboring frames is computed as an additional input. Finally, a linear layer is used to predict the probability for each frame. The model was trained on the DIHARD III development set for around 10 epochs. We augmented the training data with the noise, music, and babble from the MUSAN [9] corpus and created some reverberated speech with simulated room impulse responses [14].

The final results of VAD were calculated by averaging posterior probabilities from the two models, followed by thresholding and median filtering. As shown in the Table II, posterior averaging of the two systems achieved the best trade-off between false alarms and missed speech than the individuals.

B. X-vector-based subsystems

*1) TDNN (**System (1)**):* The TDNN x-vector-based system consists of two main parts: TDNN extractor and the VBx clustering.

TDNN-based extractor: It employs 40-dimensional filterbanks, with a $25\,\mathrm{ms}$ window and $15\,\mathrm{ms}$ frame shift. These features are used for the embedding extraction as in [15]. The x-vector was trained using a TDNN with a $1.5\,\mathrm{s}$ window with frame shift of $0.25\,\mathrm{s}$. The TDNN extractor consists of four TDNN-ReLU layers each of them followed by a dense-ReLU. Then, two dense-ReLU layers are incorporated before a pooling layer; a final dense-ReLU is included from which 512-dimension embeddings are computed. A dense-softmax concludes this TDNN architecture [16].

VBx clustering: To eliminate the need for a tuned agglomerative hierarchical clustering (AHC) stopping threshold, we perform VBx-clustering after AHC [15]. The VBx-clustering is a simplified variational Bayes diarization. It follows a hidden Markov model (HMM), in which the state represents a speaker, and the state transitions correspond to speaker turns. The state distributions, or emission probabilities, are Gaussian mixture models constrained by eigenvoice matrix. Each speaker has a probability of P_{loop} when the HMM ends up back in the same state. The initialization for this system is a probabilistic LDA (PLDA) model. For our experiments, this PLDA is the result of the interpolation of the VoxCeleb PLDA and the in-domain DIHARD III PLDA. Both PLDAs were centered and whitened using DIHARD III development set.

For the TDNN-based system, the x-vectors were projected from 512 dim to 220 using an LDA, the PLDA interpolation regulated by an alpha was set to 0.50, and the value for P_{loop} to 0.80.

We finally applied overlap assignment, which is described in Section III-B3, to obtain the final diarization results from this subsystem.

*2) Res2Net (**System (2)**):* Initially proposed for image recognition, Res2Net was applied to speaker embedding because it provides highly accurate speaker clustering [17].

The Res2Net-based extractor uses the default configuration described in [17]. The Res2Net uses 80 log-filterbank dimensions as input, a multi head-attentive pooling with attention heads set to 16 that learns to weight each frame, and additive angular margin Softmax (AAM) [18] with margin of 0.1 and scale of 30 as a training criterion. For our experiments, we employed four extractors:

i) **Res2Net-UN**: This extractor employs the default configuration of a Res2Net with 23 layers, utterance normalization, $\log_{10}$ compression, AAM margin of 0.1, and AAM scale of 30.
ii) **Res2Net-BN**: This extractor is similar to Res2Net-UN, with a batch normalization layer instead of utterance normalization and $\ln$ compression.
iii) **Res2Net-BN-Large**: This extractor uses a Res2Net with 50 layers with a similar configuration as Res2Net-BN.

iv) **Res2Net-UN-Large**: This extractor uses a Res2Net with 50 layers and a similar configuration as Res2Net-UN. Additionally, it uses SpecAugment [19] for data augmentation.

We employed the VoxCeleb 1 and VoxCeleb 2 sets [8] as training that provided 7323 speakers and over 1M of recordings. We augmented the data following similar data augmentation as the Kaldi recipe for VoxCeleb[1]. Each audio recording is randomly chunked into subsegments of length between $2.0\,\mathrm{s}$ and $4.5\,\mathrm{s}$ that are feed into the models.

Similarly to the TDNN-based system, the 128-dimension embeddings, were passed through and LDA without reduction and used a PLDA interpolation regulated by an alpha was set to 0.10, and the value for P_{loop} to 0.80.

Once the results from the four extractors were obtained, we combined the results using modified DOVER-Lap, which is explained in Section III-E.

3) Overlap detection and assignment: For the Res2Net and the TDNN x-vector subsystems, we used a similar approach to perform overlap detection like the one shown for the SincNet-based VAD, with the only difference that the classifier will distinguish between overlapping speech versus non-overlapping speech. We assigned the closest other speaker in the time axis as the second speaker for each detected frame.

Table III shows the diarization error rates (DERs) and Jaccard error rates (JERs) on the DIHARD III development set using the x-vector-based subsystems.

C. EEND-based subsystems

We employed EEND-EDA [20] and SC-EEND [21] as EEND-based subsystems, each of which can handle a flexible number of speakers. The inputs to the EEND-based models were based on log-Mel filterbanks but with different configurations for each model. For EEND-EDA, 23-dimensional log-Mel filterbanks was extracted with frame length of $25\,\mathrm{ms}$ and frame shift of $10\,\mathrm{ms}$ from $8\,\mathrm{kHz}$ recordings. Each filterbanks were then concatenated with those from the left and right seven frames to construct 345-dimensional features. We subsampled them by a factor of 10 to obtain input features for each $100\,\mathrm{ms}$ during training and that of five to obtain features for each $50\,\mathrm{ms}$ during inference. For SC-EEND, we used 40-dimensional log-Mel filterbanks from $16\,\mathrm{kHz}$ recordings and concatenated the left and right 14 frames to construct 1160-dimensional features. The subsampling factor was set to 20 during pretraining using simulated mixtures and 10 during adaptation and inference.

*1) EEND-EDA (**System (3)**):* EEND-EDA [20] calculates posteriors by dot products between time-frame-wise embeddings and speaker-wise attractors, which are calculated from the embeddings using encoder-decoder attractor calculation module (EDA). The training procedure depends on simulated mixtures summarized in Table IV and the DIHARD III corpus. We created them using the script provided in the EEND repository[2] with various β values shown in Table IV, which determines the average duration of silence between utterances. We first trained the model using Sim2spk for 100 epochs, then finetuned it on the concatenation of Sim1spk to Sim5spk for another 75 epochs, and finally adapted it on the DIHARD III development set for 200 epochs. We used Adam optimizer [22] for all the training, but with Noam scheduler [23] that set the warm-up steps to 100,000 iterations for training on simulated mixtures and with a fixed learning rate of 1×10^{-5} for adaptation.

During inference, we used the dereverberated audio using weighted prediction error (WPE) [24]. We estimated a dereverberation filter on Short Time Fourier Transform (STFT) spectrum using the entire audio recording as an input block. The STFT features are computed using a window of $32\,\mathrm{ms}$ (512 dims) and shifting of $8\,\mathrm{ms}$ (128 dims). Using 5 iterations, we set the prediction delay and the filter length to 3 and 30, respectively, for $16\,\mathrm{kHz}$.

Because EEND-based models conduct speaker diarization and voice activity detection simultaneously, they must be incorporated with oracle speech segments (for Track 1) or accurate external VAD (for Track 2) to fit the DIHARD tasks. Thus, once the diarization results were obtained using the EEND-EDA model, we filtered false alarms and recovered missed speech by assigning the speakers with the highest posterior probabilities using VAD. In this paper, we call these procedures VAD post-processing.

Even if the adaptation was based on the DIHARD III development set, which contains mixtures of at most 10 speakers, it is difficult to produce diarization results of more than five speakers because its pretraining was based on mixtures in which include at most five speakers. Therefore, we produce diarization results for more than five speakers using an iterative inference as follows:

i) decide the maximum number of speakers $K(\leq 5)$ to decode,
ii) decode at most K speaker's diarization results,
iii) stop inference if the estimated number of speakers is less than K otherwise continue to the next step,
iv) select frames in which all the decoded speakers are inactive and back to i),

We varied $K \in \{1, 2, 3, 4, 5\}$ at the first iteration and fixed it to 5 from the second iteration. Finally, the five estimated results are combined using the modified DOVER-Lap described in Section III-E to obtain the final results of the EEND-EDA-based system.

Table Va shows DERs and JERs of the EEND-EDA-based and SC-EEND-based subsystems. It clearly indicates that the VAD post-processing and the iterative inference improved the diarization performance.

*2) SC-EEND (**System (4)**):* SC-EEND is a model which estimates each speaker's speech activities one-by-one, conditioned on the previously estimated speech activities. We

[1]https://github.com/kaldi-asr/kaldi/blob/master/egs/voxceleb/v2

[2]https://github.com/hitachi-speech/EEND/blob/master/egs/callhome/v1/run_prepare_shared_eda.sh

TABLE III: DERs / JERs (%) of x-vector-based subsystems on the DIHARD III development set.

(a) TDNN (System (1))

Method	DER / JER (%)
x-vector + VBx	16.33 / 34.18
x-vector + VBx + OvlAssign	13.87 / 32.73

(b) Res2Net (System (2))

	DER / JER (%)			
Method	Res2Net-BN	Res2Net-UN	Res2Net-BN-Large	Res2Net-UN-Large
x-vector + VBx	17.24 / 37.12	17.04 / 36.17	16.85 / 35.86	17.08 / 35.95
x-vector + VBx + OvlAssign	14.89 / 35.64	14.72 / 34.65	14.56 / 34.31	14.74 / 34.40
Modified DOVER-Lap	14.04 / 34.29			

TABLE IV: Simulated mixtures used for EEND-EDA training. Sim1spk, Sim2spk, Sim3spk, and Sim4spk are the same as ones used in the EEND-EDA paper [20].

Dataset	#Spk	#Mixtures	β	Overlap ratio (%)
Sim1spk	1	100,000	2	0.0
Sim2spk	2	100,000	2	34.1
Sim3spk	3	100,000	5	34.2
Sim4spk	4	100,000	9	31.5
Sim5spk	5	100,000	13	30.3

TABLE V: DERs and JERs (%) on the DIHARD III development set using EEND-based models. FA: false alarm, MI: missed speech.

(a) EEND-EDA (System (3))

Method	DER	JER
EEND-EDA	18.77	38.98
+ filter FA	17.33	37.92
+ recover MI	13.08	35.38
+ iterative inference ($K = 5$)	13.35	34.19
+ iterative inference ($K \in \{1, \ldots, 5\}$) & DOVER-Lap	**12.92**	**33.85**

(b) SC-EEND (System (4))

Method	DER	JER
SC-EEND	18.61	39.19
+ filter FA	16.02	37.46
+ recover MI	**13.13**	**35.35**

used stacked Conformer encoders [25] instead of Transformer encoders that used in the original SC-EEND. The model was firstly trained on simulated mixtures, each of which contains at most four speakers, for 100 epochs using Adam optimizer with the same scheduler as in EEND-EDA. Then, the model was initialized with the average weights of the last 10 epochs and trained again on the simulated mixtures for additional 100 epochs. Finally, the model was adapted on the DIHARD III development set from the average weights of the last 10 epochs of the second-round pretraining for additional 200 epochs using Adam optimizer with the fixed learning rate of 1×10^{-5}. The details of the simulated mixtures are described in the SC-EEND paper [21].

For SC-EEND, we also used dereverberated audio and applied VAD post-processing (filtering false alarms and recovering missed speech) as described in Section III-C1. However, the Conformer encoders have order dependency so that we cannot conduct the decoding process only for the selected frames that are not always equally spaced along the time axis. Therefore, we did not apply the iterative inference for the SC-EEND model. The results of SC-EEND with step-by-step improvement by using VAD post-processing are shown in Table Vb.

*D. Hybrid subsystem (**System (5)**)*

We also used EEND as post-processing (EENDasP) [6] to refine diarization results obtained from the TDNN-based x-vectors described in Section III-B1. In EENDasP, two speakers from the results are iteratively selected and their results are updated using the EEND model. In the original paper, the EEND-EDA model was trained to output only two-speaker results, but we used the first two speakers' output from the model trained in Section III-C1 for our system. By applying EENDasP for TDNN-based x-vectors with VBx clustering but without heuristic overlap assignment, DER was improved from $16.33\,\%$ to $12.63\,\%$.

E. System fusion

To combine multiple diarization results, we used DOVER-Lap [7] with a modification. The original DOVER-Lap assigns uniformly-divided regions for each speaker if the multiple speakers are weighted equally in the label voting stage. However, we found that it leads to an increase in missed speech. This is obvious by considering the case when the same three hypotheses with overlaps are input to DOVER-Lap. The speakers included in the hypotheses are always tied in this case; thus, overlapped regions in the hypotheses are divided to be assigned for each speaker, which results in the combined hypothesis with no overlap. Thus, we assigned all the tied speakers to the regions without any division.

When we combine diarization results from various systems, we sometimes know that some systems are highly accurate and others are not so. Therefore, we introduced hypothesis-wise manual weighting to DOVER-Lap. The original DOVER [26] and DOVER-Lap, the input hypotheses are ranked by their average DER to all the other hypotheses. In other words, the hypotheses $H_1, \ldots, H_k, \ldots, H_K$ are ranked by following score s_k:

$$s_k = \frac{1}{K-1} \sum_{k' \in \{1,\ldots,K\}, k \neq k'} DER\left(H_k, H_{k'}\right), \qquad (1)$$

where $DER(H_k, H_{k'})$ is the function to calculate diarization error rate from the reference H_k and estimation $H_{k'}$. In our

TABLE VI: Comparison between the original and modified DOVER-Lap on the DIHARD III development set. MI: missed speech, FA: false alarm, CF: speaker confusion.

Method	MI	FA	CF	DER
(1) TDNN-based x-vector + VBx + OvlAssign	5.36	1.93	6.58	13.87
(2) Res2Net-based x-vector + VBx + OvlAssign	5.47	1.89	6.68	14.04
(3) EEND-EDA	6.54	1.36	5.02	12.92
(4) SC-EEND	4.85	1.96	6.32	13.13
(5) TDNN-based x-vector + VBx + EENDasP	6.53	1.32	4.79	12.63
DOVER-Lap	6.96	0.77	4.33	12.07
Modified DOVER-Lap (System (6))	5.53	0.93	4.27	10.73
Modified DOVER-Lap + manual weighting	5.54	0.93	4.21	10.68

system, we used $w_k s_k$ instead of s_k, where $w_k \in \mathbb{R}_+$ is a weighing value, to control the importance of each hypothesis.

Table VI shows DERs and breakdown on the DIHARD III development set. Note that the manual weighting was only used to combine five hypotheses in System (9) and not used for combine the Res2Net-results in System (1), EEND-EDA iterative inference in Systems (3) and (7), and the five-system fusion for System (6) due to time constraints. The weights to combine Systems (1)(2)(4)(7)(9) were set to $w_{(1)} = 2$, $w_{(2)} = 2$, $w_{(4)} = 1$, $w_{(7)} = 4$, $w_{(9)} = 3$, which were determined by using the development set.

F. Self-supervised adaptation

After the first system fusion, we applied self-supervised adaptation (SSA) for the EEND-EDA model. The estimated results were used as the pseudo labels for the DIHARD III evaluation set, We redid the adaptation step in Section III-C1 on the concatenation of the DIHARD III development set with the ground truth labels and the evaluation set with the pseudo labels. With the new model, we placed the results of the EEND-EDA (System (3)), EENDasP (System (5)), and DOVER-Lap (System (6)). Note that we used different pseudo labels for Track 1 and Track 2 because the oracle VAD was only available on Track 1.

IV. Results

Table VII shows the results on the DIHARD III development set and evaluation set. The results on the evaluation set are from the official scoring server. Every subsystem significantly outperformed the baseline system [1]. System (5) performed best as a single subsystem without self-supervised adaptation, but the other four subsystems showed the comparable performance. Our best system achieved $11.58\,\%$ and $14.09\,\%$ of DERs on the full and core evaluation set in Track 1, respectively. It also achieved $16.94\,\%$ and $20.01\,\%$ of DERs in Track 2.

V. Hardware requirements

We run our experiments using two different infrastructures. One is equipped with Intel® Xeon® CPU Gold 6123 @ $2.60\,\mathrm{GHz}$ using up to 56 threads with 750 GB of RAM, and up to eight NVIDIA® V100® GPUs with 32 GB of VRAM each and 15.7 single-precision TFLOPS. Using this infrastructure, we trained and processed the VAD models, the Res2Net models, the PLDA model, the EEND-based systems, and DOVER-Lap.

The other is the JHU's CLSP Cluster, which is equipped with Intel® Xeon® CPU E5-2680 v2 @ $2.80\,\mathrm{GHz}$ using up to 54 threads and 60GB of RAM, and up to four NVIDIA® GeForce GTX 1080 Ti® with 11 GB of VRAM each and 10.6 single-precision TFLOPS. The TDNN-based extractor, the VBx clustering, and the overlap detection and assignment model were trained on this cluster.

The processing time for WPE dereverberation is $2.54\,\mathrm{s}$ for 1 minute of audio.

Our framework's components were trained on PyTorch [27], except for the TDNN-based extractor that was trained on Kaldi [13].

The SincNet VAD was trained on a single NVIDIA® V100® GPU and required about 22 hours for training. The processing of the labels required $0.132\,\mathrm{s}$ for 1 minute of audio.

The TDNN VAD was trained with 3 to 8 NVIDIA® GeForce GTX 1080 Ti® (We gradually increased the number of GPU jobs during training) for 1 hour.

The TDNN x-vectors, VBx, and the overlap detector extraction were conducted on the CLSP Cluster. The overlap detector required 40 CPUs with a decoding time of 30 mins for all datasets including development and evaluation sets. The TDNN x-vector was trained on 4-8 GPUs and required approximately 48 hours. The PLDAs, trained using CPUs, required around 30 mins to train the VoxCeleb datasets, and 10 mins to train the DIHARD III dataset. The scoring for every file took around $0.25\,\mathrm{s}$ for each audio. All the procedures were parallelized using 30 to 40 jobs to reduce the computational time.

The Res2Net-based x-vector extractors were trained using four NVIDIA® V100® GPUs and required 54 hours approximately for training. The processing time for the x-vector extraction using this model is $1.52\,\mathrm{s}$ for 1 minute of audio.

The EEND-based models are trained using a single NVIDIA® V100® GPU. For EEND-EDA, it took 30 hours for training on Sim2spk, 325 hours for finetuning on the concatenation of Sim1spk to Sim5spk, and 1.5 hours for adaptation on the DIHARD III development set. The processing time of iterative inference and VAD post-processing was about 30 minutes. It takes about 3 hours for self-supervised adaptation, which was almost doubled from the adaptation on the development set because we additionally used the evaluation set with pseudo labels. For SC-EEND, it took 200 hours for training on simulated mixtures, 2 hours for adaptation, and 5 minutes for inference.

The processing time of EENDasP given the results of TDNN-based x-vectors + VBx was about 5 minutes for the entire development set.

DOVER-Lap of five systems was based on the official repository[3], which took about 3 minutes to process the development set.

[3]https://github.com/desh2608/dover-lap

TABLE VII: DERs / JERs (%) on Track 1 & 2.

	Track 1 (w/ oracle VAD)				Track 2 (w/o oracle VAD)			
	Dev		Eval		Dev		Eval	
System	full	core	full	core	full	core	full	core
Baseline [1]	19.41 / 41.66	20.25 / 46.02	19.25 / 42.45	20.65 / 47.74	21.71 / 43.66	22.28 / 47.75	25.36 / 46.95	27.34 / 51.91
(1) TDNN-based x-vector + VBx + OvlAssign	13.87 / 32.73	14.88 / 36.72	15.65 / 33.71	18.20 / 38.42	17.61 / 36.03	18.64 / 39.92	21.47 / 37.83	24.58 / 42.02
(2) Res2Net-based x-vector + VBx + OvlAssign	14.04 / 34.29	15.18 / 38.80	15.81 / 35.53	18.47 / 40.47	17.26 / 37.17	18.39 / 41.56	21.37 / 39.59	24.64 / 44.49
(3) EEND-EDA	12.92 / 33.85	13.95 / 35.37	13.95 / 35.37	17.28 / 41.97	15.90 / 35.94	18.50 / 41.71	19.04 / 38.89	22.84 / 45.27
(4) SC-EEND	13.13 / 35.35	16.05 / 41.80	15.16 / 38.62	19.14 / 46.04	16.16 / 37.52	19.00 / 43.74	20.30 / 42.19	24.75 / 49.36
(5) TDNN-based x-vector + VBx + EENDasP	12.63 / 31.52	14.61 / 36.28	13.30 / 33.02	15.92 / 38.29	15.94 / 34.11	18.09 / 38.97	18.13 / 35.82	21.31 / 40.78
(6) DOVER-Lap of (1)(2)(3)(4)(5)	10.73 / 31.39	12.56 / 36.88	11.83 / 32.85	14.41 / 38.81	14.13 / 34.32	16.06 / 39.75	17.21 / 37.64	20.34 / 43.40
(7) EEND-EDA (SSA)	12.95 / 33.98	15.69 / 40.03	12.74 / 34.08	15.86 / 40.44	15.03 / 33.64	17.52 / 39.15	17.81 / 38.32	21.31 / 44.32
(8) TDNN-based x-vector + VBx + EENDasP (SSA)	12.54 / 31.32	14.55 / **36.11**	12.74 / **32.20**	15.34 / **37.50**	15.45 / 33.61	17.77 / **38.67**	17.60 / **35.16**	20.84 / **40.18**
(9) DOVER-Lap of (1)(2)(4)(7)(8)	**10.65** / **30.82**	**12.47** / 36.21	**11.58** / 32.37	**14.09** / 38.25	**13.85** / **33.41**	**15.81** / 38.77	**16.94** / 36.31	**20.01** / 41.78

The trained models and the generated outputs had a total disk usage of 1.2 TB.

REFERENCES

[1] N. Ryant, P. Singh, V. Krishnamohan, R. Varma, K. Church, C. Cieri, J. Du, S. Ganapathy, and M. Liberman, "The third DIHARD diarization challenge," arXiv:2012.01477, 2020.

[2] F. Landini, S. Wang, M. Diez, L. Burget, P. Matějka, K. Žmolíková, L. Mošner, O. Plchot, O. Novotnỳ, H. Zeinali, and J. Rohdin, "BUT system description for DIHARD Speech Diarization Challenge 2019," arXiv:1910.08847, 2019.

[3] F. Landini, S. Wang, M. Diez, L. Burget, P. Matějka, K. Žmolíková, L. Mošner, A. Silnova, O. Plchot, O. Novotnỳ, H. Zeinali, and J. Rohdin, "BUT system for the Second DIHARD Speech Diarization Challenge," in *ICASSP*, 2020, pp. 6529–6533.

[4] X. Xiao, N. Kanda, Z. Chen, T. Zhou, T. Yoshioka, S. Chen, Y. Zhao, G. Liu, Y. Wu, J. Wu, S. Liu, J. Li, and Y. Gong, "Microsoft speaker diarization system for the VoxCeleb Speaker Recognition Challenge 2020," arXiv:2010.11458, 2020.

[5] 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.

[6] S. Horiguchi, P. García, Y. Fujita, S. Watanabe, and K. Nagamatsu, "End-to-end speaker diarization as post-processing," in *ICASSP*, 2021 (to appear).

[7] D. Raj, L. P. Garcia-Perera, Z. Huang, S. Watanabe, D. Povey, A. Stolcke, and S. Khudanpur, "DOVER-Lap: A method for combining overlap-aware diarization outputs," in *SLT*, 2021, pp. 881–888.

[8] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "VoxCeleb: Large-scale speaker verification in the wild," *Computer Speech & Language*, vol. 60, p. 101027, 2020.

[9] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," arXiv:1510.08484, 2015.

[10] M. Lavechin, M.-P. Gill, R. Bousbib, H. Bredin, and L. P. Garcia-Perera, "End-to-end domain-adversarial voice activity detection," in *INTERSPEECH*, 2020, pp. 3685–3689.

[11] H. Bredin, R. Yin, J. M. Coria, G. Gelly, P. Korshunov, and et al., "pyannote.audio: neural building blocks for speaker diarization," in *ICASSP 2020*, 2020, pp. 7124–7128.

[12] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with SincNet," in *SLT*, 2018, pp. 1021–1028.

[13] 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," in *ASRU*, 2011.

[14] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *ICASSP*, 2017, pp. 5220–5224.

[15] M. Diez, L. Burget, and P. Matejka, "Speaker diarization based on Bayesian HMM with eigenvoice priors," in *Odyssey*, 2018, pp. 102–109.

[16] 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, pp. 5796–5800.

[17] T. Zhou, Y. Zhao, and J. Wu, "ResNeXt and Res2Net structures for speaker verification," in *SLT*, 2021, pp. 301–307.

[18] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *CVPR*, 2019, pp. 4685–4694.

[19] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *INTERSPEECH*, 2019, pp. 2613–2617.

[20] S. Horiguchi, Y. Fujita, S. Wananabe, Y. Xue, and K. Nagamatsu, "End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors," in *INTERSPEECH*, 2020, pp. 269–273.

[21] Y. Fujita, S. Watanabe, S. Horiguchi, Y. Xue, J. Shi, and K. Nagamatsu, "Neural speaker diarization with speaker-wise chain rule," arXiv:2006.01796, 2020.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 5998–6008.

[24] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. Juang, "Speech dereverberation based on variance-normalized delayed linear prediction," *IEEE TASLP*, vol. 18, no. 7, pp. 1717–1731, 2010.

[25] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *INTERSPEECH*, 2020, pp. 5036–5040.

[26] A. Stolcke and T. Yoshioka, "DOVER: A method for combining diarization outputs," in *ASRU*, 2019, pp. 757–763.

[27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019, pp. 8024–8035.