

Federated Learning with Feature Space Separation for Speaker Recognition

Ying Meng^{1,3,*}, Zhihua Fang^{1,3,*}, Liang He^{1,2,3,4}

¹School of Computer Science and Technology, Xinjiang University, China

²School of Intelligence Science and Technology, Xinjiang University, China

³Xinjiang Key Laboratory of Signal Detection and Processing, China

⁴Department of Electronic Engineering, Tsinghua University, China

mengying@stu.xju.edu.cn, fangzhihua@stu.xju.edu.cn, heliang@mail.tsinghua.edu.cn

Abstract

The performance of deep speaker models relies on large-scale, high-quality datasets. To improve the generalization of speaker models, federated learning can be used to share the knowledge learned by local models in private data. However, traditional federated learning methods have mapping conflict problems between local models. In this paper, we propose a federal speaker recognition method with feature space separation. Specifically, we introduce feature anchors for each speaker and transmit them when the client and server communicate. Local models use anchor loss to constrain the feature distribution of private data to avoid mapping conflict. In addition, we introduce a dynamic weight aggregation method when the server aggregates the local models, which amplifies the weights of well-performed local models to achieve a better aggregation effect. Extensive experiments and analyses in VoxCeleb and CN-Celeb demonstrate the effectiveness of our proposed method.

Index Terms: Speaker recognition, federated learning, feature space separation, dynamic weight aggregation

1. Introduction

Speaker recognition [1] has been widely used in many important scenarios. The speaker recognition system consists of two main components, the front-end speaker embedding network and the back-end scoring system. Currently, the speaker recognition task is mainly implemented by Deep Neural Networks (DNNs) [2, 3, 4, 5, 6, 7, 8], whose typical application model is to train and deploy the model independently by a single data-holding organization.

The construction of efficient speaker recognition models relies on large-scale, diverse, and high-quality data sets to ensure the robustness and generalizability of the model [9, 10]. Existing speaker recognition methods [11, 12, 13, 14] have achieved promising performance, but they always require centralized training. However, accumulating such data resources independently by a single organization is usually challenging in terms of high cost and technical complexity. Meanwhile, global increased public awareness of privacy protection and stringent personal data protection regulations (e.g., the EU's General Data Protection Regulation [15]) require organizations to follow strict security compliance standards when handling sensitive personal information.

Federated learning [16] allows multiple devices or organizations to collaboratively train models without sharing data. It achieves this by sharing model parameters. However, widely

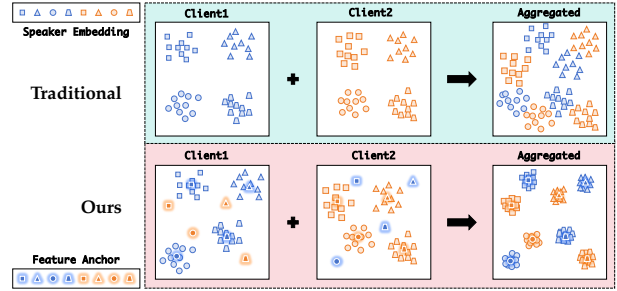


Figure 1: **Top:** In traditional approaches, local models map private data to the entire feature space, and the aggregated global model has weak discriminative ability. **Bottom:** Our approach shares feature anchors between clients, allowing local models to learn private data while reserving space in the feature space for other speakers.

used federated learning algorithms [17, 18, 19, 20] assume that clients share the same data categories, which is unlikely in multi-user federated speaker recognition scenarios. In federated speaker recognition research [21, 22], user devices are used as clients for training, which not only infringes on computational resources but also yields poor collaborative learning outcomes. In this study, we explore a more realistic federated speaker recognition scenario: two or more organizations holding speaker databases collaborate through federated learning.

However, the traditional federated learning method has a problem. As shown in Figure 1, when multi-organization collaborative learning. The local model can classify private data well, but there is a mapping conflict problem in the feature space of different clients, which affects the recognition performance of the aggregated model. In addition, it is not an optimal strategy for most federated learning algorithms to assign aggregation weights based on the proportion of client data volume during aggregation [23, 24, 25].

To address the above problems, in this paper, we propose a **Federated speaker recognition method with Feature Space Separation (FedFSS)**. Specifically, we introduce feature anchors for each speaker identity and pass these feature anchors when the client communicates with the server so that the local model can access the feature anchors of other clients and avoid the mapping conflict problem by constraining the feature distribution of private data through anchor loss. In addition, we introduce the dynamic weight aggregation method when the server is aggregating local models, which achieves a better aggregation effect by amplifying the weights of local models with good performance. In summary, the contributions of this paper are as follows:

*These authors contributed equally.

Corresponding author: Liang He. Code and data are available at <https://github.com/xju-cs/FedFSS>.

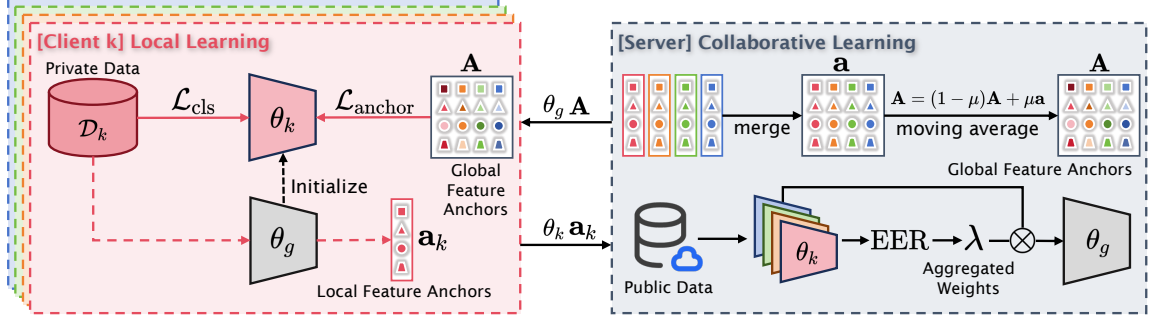


Figure 2: **Overview of our proposed method.** Compared to the general method, global feature anchors are incorporated to constrain the local model θ_k , in addition to introducing dynamic aggregation weights to compute the global model θ_g . The client communicates with the server every T_l epochs of training, for a total of T_c communications. For details, see Algorithm 1.

- We propose a federated learning approach with *Feature Space Separation* in the field of speaker recognition, which solves the data mapping conflict problem by providing corresponding feature anchors for different speakers.
- We introduce a *Dynamic Weight Aggregation* approach to assign higher weights for more robust local models, increasing model performance.
- Extensive experiments and analyses have demonstrated the effectiveness of the proposed method.

2. Methodology

Federated learning for speaker recognition typically comprises two phases: local learning and collaborative learning. Local learning refers to the process where clients train local models on their private datasets after receiving the global model from the server. Collaborative learning involves collecting local models from all clients, aggregating them into an updated global model, and redistributing the refined model back to clients. In this work, we enhance both phases: *Feature Space Separation* mechanism (Section 2.2) is integrated into the local learning phase to strengthen the model’s discriminative capability for unseen speakers. *Dynamic Weight Aggregation* mechanism (Section 2.3) is introduced during collaborative learning to optimize global model performance. The framework we proposed is shown in Figure 2.

2.1. Problem Definition and Notation

Under the federated learning speaker recognition framework, we consider K clients and a server. The server has a global model θ_g and a public dataset $\mathcal{D}_0 = \{(x_i^0, y_i^0)\}_{i=1}^{N_0}$. For the k -th client c_k contains a private dataset $\mathcal{D}_k = \{(x_i^k, y_i^k, \text{id}_i^k)\}_{i=1}^{N_k}$, $|\mathcal{D}_k| = N_k$, where x_i^k is the speech data, $y_i^k \in \{0, 1\}^{C_k}$ indicates the one-hot vector of the speaker label, id_i^k is the speaker string ID, and C_k is the number of speakers in client c_k . Client c_k holds the embedding extractor $f(\cdot)$ parameterized by θ_k and the classifier $h(\cdot)$ parameterized by ψ_k , $\mathbf{e} = f(x; \theta_k)$ and $h(\mathbf{e}; \psi_k)$ represents speaker embedding and logits outputs, respectively. In addition, collaborative learning includes T_c epochs, and local learning includes T_l epochs.

2.2. Local Learning with Feature Space Separation

In local learning, each client can achieve good classification results when training the local model in their private dataset, but since the local model does not have access to the data of other

clients, different local models may map the speaker data with different identities to the same feature space location (as shown in Figure 1), and this **mapping conflict** affects the aggregated model’s recognition ability.

To solve the mapping conflict problem, we introduce a *Feature Space Separation* mechanism. Specifically, a feature anchor $\mathbf{a}_k = \{\{\text{id}_{d_1} : \bar{\mathbf{e}}_1\}, \dots, \{\text{id}_{d_{C_k}} : \bar{\mathbf{e}}_{C_k}\}\}$ is introduced for each client c_k , where id_{d_i} is the speaker ID and $\bar{\mathbf{e}}_i$ the average embedding belonging to speaker id_{d_i} . Since the parameters of the local model are always updated during the training process, the average embedding is extracted using the obtained global model:

$$\bar{\mathbf{e}}_i = \frac{1}{|\mathcal{D}_{\text{id}_i}|} \sum_{j=1}^{|\mathcal{D}_{\text{id}_i}|} \mathbf{e}_j \quad (1)$$

where $\mathbf{e}_j = f(x_j; \theta_g)$ and $\mathcal{D}_{\text{id}_i}$ is the set of data belonging to speaker id_{d_i} . In the collaborative learning process, feature anchors $\mathbf{a}_k$ are sent to the server along with the local model θ_k , and the server merges the feature anchors from different clients together $\mathbf{A} = \{\mathbf{a}_1, \dots, \mathbf{a}_K\}$, and then through moving average updating the old global feature anchors:

$$\mathbf{A} = (1 - \mu)\mathbf{A} + \mu\mathbf{a} \quad (2)$$

where a moving average is computed between feature anchors with the same speaker ID, and μ is the smoothing factor, noting that the initial global feature anchors are randomly parameterized.

The global feature anchors $\mathbf{A}$ are updated and then delivered to each client, and for each client c_k , the anchor loss between the embedding of the private dataset and $\mathbf{A}$ is computed to enhance the local model’s discriminative properties for unseen data:

$$\mathcal{L}_{\text{anchor}} = \frac{1}{N_k * |\mathbf{A}|} \sum_{(x, \text{id}) \in \mathcal{D}_k} \sum_{\{\text{id} : \bar{\mathbf{e}}\} \in \mathbf{A}} \text{CE}(\mathbf{s}, \mathbb{I}(\text{id} == \text{id})) \quad (3)$$

where $\mathbf{s} = \text{cosine}(f(x; \theta_k), \bar{\mathbf{e}})$ is the cosine similarity between speaker embedding and feature anchor, $\text{CE}(\cdot, \cdot)$ is the cross-entropy loss function, and $\mathbb{I}(\cdot)$ is the indicator function, with a value of 1 when the condition is true and 0 otherwise.

The loss of local learning is jointly supervised by the classification loss and the anchor loss:

$$\mathcal{L}(\mathcal{D}_k; \theta_k, \psi_k) = (1 - \alpha)\mathcal{L}_{\text{cls}} + \alpha\mathcal{L}_{\text{anchor}} \quad (4)$$

where $\mathcal{L}_{\text{cls}} = -\frac{1}{N_k} \sum_{(x, y) \in \mathcal{D}_k} \text{AM}(h(f(x; \theta_k); \psi_k), y)$, $\text{AM}(\cdot, \cdot)$ is the additive margin softmax (AM-Softmax) loss [26] and α is the balance factor between the two losses.

Algorithm 1: The FedFSS framework pseudocode.

Input: Public dataset $\mathcal{D}_0$, private dataset $\{\mathcal{D}_k\}_{k=1}^K$,
local model $\{(\theta_k, \psi_k)\}_{k=1}^K$, global model θ_g ,
number of collaborative learning epochs T_c ,
number of local learning epochs T_l , global
feature anchors $\mathbf{A}$, learning rate η ,
hyperparameters α, γ, μ .
Output: $\{(\theta_k, \psi_k)\}_{k=1}^K$
// Local Learning
1 **Client** ($k, \theta_g, \mathbf{A}$):
2 $\theta_k = \theta_g$
3 **for** e **in** $1, \dots, T_l$ **do**
4 Obtain $\mathcal{L}(\mathcal{D}_k; \theta_k, \psi_k)$ by Eq. (4)
5 $(\theta_k, \psi_k) = (\theta_k, \psi_k) - \eta \nabla \mathcal{L}(\mathcal{D}_k; \theta_k, \psi_k)$
6 Obtain $\mathbf{a}_k$ by Eq. (1)
7 **return** $\mathbf{a}_k, \theta_k$
// Collaborative Learning
8 **Server** ($\theta_g, \mathbf{A}$):
9 **for** e **in** $1, \dots, T_c$ **do**
10 **for** k **in** $1, \dots, K$ **do**
11 $\mathbf{a}_k, \theta_k = \text{Client}(k, \theta_g, \mathbf{A})$
12 Merge $\{\mathbf{a}_k\}_{k=1}^K$ into $\mathbf{a}$
13 Update $\mathbf{A}$ by Eq. (2)
14 Obtain θ_g by Eq. (6)
15 **main**():
16 Randomly initialize $\theta_g, \mathbf{A}$
17 **Server** ($\theta_g, \mathbf{A}$)

2.3. Collaborative Learning with Dynamic Weight Aggregation

The process of collaborative learning usually involves the server aggregating the local models of the clients into a global model and then redistributing the global model to each client. The aggregation algorithm is usually defined as $\theta_g = \sum_{k=1}^K \lambda_k \theta_k$, where $\lambda_k = \frac{N_k}{\sum_{k=1}^K N_k}$, and the client's aggregation weight is fixedly set as a percentage of the client's data volume.

Compared to fixed weights, we want better-performing local models to have higher weights when aggregated, so we introduce a *Dynamic Weight Aggregation* approach. Specifically, we place a smaller public speaker dataset $\mathcal{D}_0$ on the server, which is feasible in practice. All local models are tested for equal error rates (EERs) on $\mathcal{D}_0$, and the resulting EER are then used to compute normalized vectors as aggregation weights:

$$\lambda_k = \frac{-\log \text{EER}_k}{\sum_{k=1}^K -\log \text{EER}_k} \quad (5)$$

In addition, it is found that the shrinking effect produced by the sum of aggregation weights less than 1 can improve the generalization performance of federated learning [24], so we introduce the contraction factor in the local model aggregation, and the final aggregation formula is as follows:

$$\theta_g = \gamma \sum_{k=1}^K \lambda_k \theta_k \quad (6)$$

where γ is the shrinkage factor, details of collaborative learning and local learning are given in Algorithm 1.

3. Experiments

3.1. Datasets

We have experimentally validated our methods on popular public speaker datasets. The development set of VoxCeleb1 [27] contains 148,642 utterances from 1,211 speakers, while the test set (Vox-O) includes over 4,000 utterances from 40 speakers. The development set of VoxCeleb2 [28] contains 1,092,009 utterances from 5,994 speakers. Vox-E and Vox-H are test pairs extracted from the development set of VoxCeleb1. There is no overlap of speakers between VoxCeleb1 and VoxCeleb2. The development set of CN-Celeb1 [29] contains 111,260 utterances from 800 speakers, while the test set (CN-Celeb.Eval) includes 18,849 utterances from 200 speakers.

3.2. Implementation Details

Data Preparation. To simulate different scales of federated speaker recognition scenarios, we randomly and uniformly divide VoxCeleb1 and VoxCeleb2 into K sub-datasets by speaker ID, respectively.

Training Details. For all experiments, we used ECAPA-TDNN with a channel size of 1024 as the embedding extractor θ , the input is the 80-dimensional log mel-spectrogram, the speaker embedding is 512-dimensional, and the loss function is AM-Softmax Loss ($m = 0.2, s = 30$). For local learning, the initial learning rate is set to 0.001 with 0.97 decay per epoch, the optimizer uses Adam [30], and the batch size is 128, $T_l = 5$, $\mu = 0.9, \alpha = 0.2$. For collaborative learning, the CMU Arctic [31] was used as $\mathcal{D}_0$, $T_c = 20, \gamma = 0.99$.

Evaluation Details. Vox-O is used to evaluate local models trained using VoxCeleb1 or VoxCeleb2. Vox-E and Vox-H are used to evaluate local models trained using VoxCeleb2. CN-Celeb.Eval is used to evaluate local models trained using CN-Celeb1. The evaluation metrics use equal error rate (EER).

Baselines. We compare FedFSS with the following popular federated learning methods: FedAvg [16], FedProx [17], and MOON [32]. Additionally, local models trained only on private datasets are used as a simple baseline (named Standard).

3.3. Results and Analysis

Results on VoxCeleb. We conducted four sets of experiments on the VoxCeleb dataset to explore the performance of the different methods, as shown in Table 1. It can be seen that our proposed FedFSS method achieved the best EER results across datasets of varying sizes and with different evaluation datasets, showing its effectiveness and stability in improving the performance of local models in federated learning scenarios. In contrast, the standard approach performs relatively poorly, demonstrating that using federated learning for collaborative learning can help clients improve the performance of local models. These results collectively show that our proposed method has significant advantages in federated speaker recognition.

Results on VoxCeleb1 and CN-Celeb1. To simulate a more realistic scenario, we set the number of clients K to 2, with each client using VoxCeleb1 and CN-Celeb1 as their private datasets, respectively. As shown in Table 2, it can be observed that even though the private data characteristics of the clients differ, the performance of local models improved through collaborative learning using federated learning. Furthermore, our proposed FedFSS achieved the best overall performance.

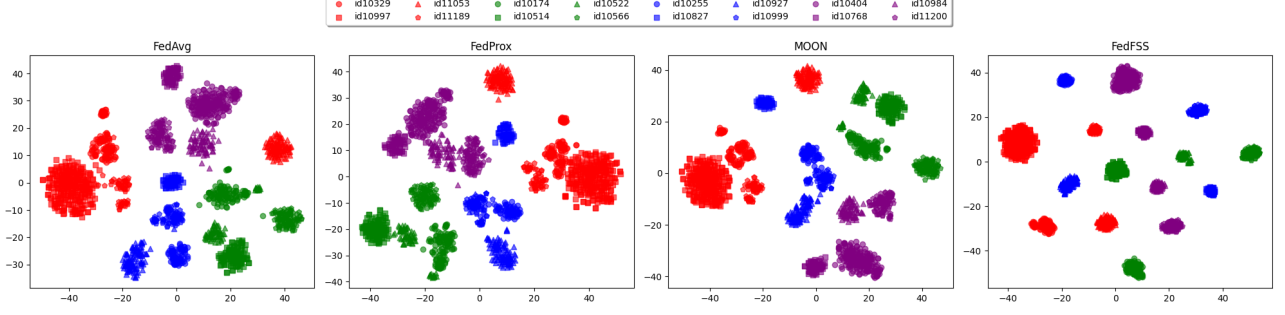


Figure 3: The t-SNE visualization comparison of speaker embeddings extracted by our proposed method and baseline methods, with different colors representing different clients.

Table 1: Comparison of the EER(%) of each local speaker model using different federated learning methods, the local datasets of Client1-4 randomly and uniformly partitioned by the development sets of VoxCeleb1 or VoxCeleb2 based on speaker ID. “Standard” indicates that federated learning is not used.

Method	Client1	Client2	Client3	Client4	Average
Exp-1: Training Data: VoxCeleb1, Dev, Evaluation Data: Vox-O.					
Standard	6.45	6.79	6.29	6.99	6.63
FedAvg [16]	4.86	4.94	4.98	5.62	5.10
FedProx [17]	5.04	5.17	4.84	5.57	5.15
MOON [32]	5.04	4.85	4.82	5.22	4.98
FedFSS (ours)	4.72	4.65	4.57	5.07	4.75
Exp-2: Training Data: VoxCeleb2, Dev, Evaluation Data: Vox-O.					
Standard	3.49	3.24	3.74	3.34	3.45
FedAvg [16]	2.56	2.50	2.59	2.40	2.51
FedProx [17]	3.14	2.87	2.99	2.92	2.98
MOON [32]	2.72	2.59	2.66	2.51	2.62
FedFSS (ours)	2.16	2.34	2.44	2.18	2.28
Exp-3: Training Data: VoxCeleb2, Dev, Evaluation Data: Vox-E.					
Standard	3.77	3.53	3.73	3.71	3.68
FedAvg [16]	2.75	2.70	2.72	2.68	2.71
FedProx [17]	3.20	3.08	2.97	2.97	3.05
MOON [32]	2.91	2.84	2.89	2.85	2.87
FedFSS (ours)	2.59	2.45	2.61	2.50	2.53
Exp-4: Training Data: VoxCeleb2, Dev, Evaluation Data: Vox-H.					
Standard	6.52	6.20	6.28	6.43	6.35
FedAvg [16]	4.98	4.78	4.92	4.81	4.87
FedProx [17]	5.25	5.08	4.86	4.94	5.03
MOON [32]	5.33	5.01	5.19	5.08	5.15
FedFSS (ours)	4.92	4.48	4.77	4.66	4.70

3.4. Visualization Analysis

We use t-SNE [33] to show the comparison between our proposed FedFSS method and baseline methods in terms of extracted speaker embeddings, as shown in Figure 3. It can be seen that FedFSS exhibits significant advantages in both clustering and inter-class separation compared to other methods. This implies that it can better adapt to the data heterogeneity in federated learning environments. This advantage is attributed to our proposed feature space separation method, which reserves locations in the feature space of the local model for speakers from other clients, thereby improving the generalization of the local model to unseen speakers.

3.5. Ablation Study

Table 3 presents the results of the ablation study. The results indicate that using *Feature Space Separation* (FSS) or *Dynamic Weight Aggregation* (DWA) individually can reduce the EER to

Table 2: Comparing the EER(%) of using VoxCeleb1 (**Client1**) and CN-Celeb (**Client2**) with the “Standard” method and different federated learning methods. **Client1** is evaluated on Vox-O, and **Client2** is evaluated on CN-Celeb.Eval.

Method	VoxCeleb1	CN-Celeb1	Average
Standard	3.24	15.51	9.37
FedAvg [16]	3.14	13.46	8.30
FedProx [17]	3.16	13.28	8.22
MOON [32]	3.24	13.62	8.43
FedFSS (ours)	3.14	12.35	7.74

some extent, with FSS showing a better effect than DWA. When FSS and DWA are used in combination, the model achieves the lowest average EER of 4.75%, demonstrating that the integration of both methods can effectively enhance the performance of clients in federated learning.

Table 3: EER(%) results of ablation experiments with the same experimental settings as **Exp-1** in Table 1.

FSS	DWA	Client1	Client2	Client3	Client4	Average
		4.86	4.94	4.98	5.62	5.10
	✓	4.82	4.93	4.65	5.32	4.93
✓		4.67	4.72	4.64	5.38	4.85
✓	✓	4.72	4.65	4.57	5.07	4.75

4. Conclusions

In this paper, we experimentally demonstrate that using federated learning to allow multiple clients to collaborate in learning can enhance the performance of each client. In addition, we propose a feature space separation method to solve the mapping conflict between local models and then introduce a dynamic weight aggregation method to adjust the aggregation weights of local models to improve the performance of the global model. Extensive experimental analysis on VoxCeleb and CN-Celeb verifies the effectiveness of our proposed method. In the future, we plan to explore more complex federated learning scenarios for speaker recognition.

5. Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 62366051).

6. References

- [1] S. Wang, Z. Chen, K. A. Lee, Y. Qian, and H. Li, "Overview of speaker modeling and its applications: From the lens of deep speaker representation learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 4971–4998, 2024.
- [2] S.-H. Liou, P.-C. Chan, C.-P. Chen, T.-C. Lin, C.-L. Lu, Y.-H. Cheng, H.-F. Chuang, and W.-Y. Chen, "Enhancing ecapadttn with feature processing module and attention mechanism for speaker verification," in *Interspeech 2024*, 2024, pp. 2120–2124.
- [3] Y. Sun, C. Li, and B. Li, "Branchformer-based tdnn for automatic speaker verification," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 10 981–10 985.
- [4] H.-J. Heo, U.-H. Shin, R. Lee, Y. Cheon, and H.-M. Park, "Nexttdnn: Modernizing multi-scale temporal convolution backbone for speaker verification," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 11 186–11 190.
- [5] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.
- [6] Desplanques, Brecht and Thienpondt, Jenthe and Demuynck, Kris, "ECAPA-TDNN : Emphasized Channel Attention, Propagation and Aggregation in TDNN based speaker verification," in *INTERSPEECH 2020*, 2020, pp. 3830–3834.
- [7] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, J. Borgstrom, L. P. García-Perera, F. Richardson, R. Dehak, P. A. Torres-Carrasquillo, and N. Dehak, "State-of-the-art speaker recognition with neural network embeddings in nist sre18 and speakers in the wild evaluations," *Computer Speech & Language*, vol. 60, p. 101026, 2020.
- [8] H. Wang, S. Zheng, Y. Chen, L. Cheng, and Q. Chen, "Cam++: A fast and efficient network for speaker verification using context-aware masking," in *Interspeech 2023*, 2023, pp. 5301–5305.
- [9] Z. Fang, L. He, H. Ma, X. Guo, and L. Li, "Robust training for speaker verification against noisy labels," in *Interspeech 2023*, 2023, pp. 3192–3196.
- [10] A. Q. Ohi, M. F. Mridha, M. A. Hamid, and M. M. Monowar, "Deep speaker recognition: Process, progress, and challenges," *IEEE Access*, vol. 9, pp. 89 619–89 643, 2021.
- [11] L. He, Z. Fang, Z. Chen, M. Xu, Y. Meng, and P. Wang, "Multi-view speaker embedding learning for enhanced stability and discriminability," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 10 081–10 085.
- [12] J. Cho, S. Park, H. Park, H. Park, S. Yang, and S. Yun, "Balanced learning for multi-domain long-tailed speaker recognition," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 7465–7469.
- [13] E. Rosello, A. M. Gomez, I. López-Espejo, A. M. Peinado, and J. M. Martín-Doñas, "Anti-spoofing ensembling model: Dynamic weight allocation in ensemble models for improved voice biometrics security," in *Interspeech 2024*, 2024, pp. 497–501.
- [14] Z. Fang, L. He, L. Li, and Y. Hu, "Improving speaker verification with noise-aware label ensembling and sample selection: Learning and correcting noisy speaker labels," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 2988–3001, 2024.
- [15] E. Union, "General data protection regulation (GDPR)," 2018. [Online]. Available: <https://gdpr-info.eu/>
- [16] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.
- [17] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proceedings of Machine Learning and Systems*, 2020, pp. 429–450.
- [18] L. Qu, Y. Zhou, P. P. Liang, Y. Xia, F. Wang, E. Adeli, L. Fei-Fei, and D. Rubin, "Rethinking architecture design for tackling data heterogeneity in federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10 061–10 071.
- [19] M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao, "Heterogeneous federated learning: State-of-the-art and research challenges," *ACM Comput. Surv.*, vol. 56, 2023.
- [20] Z. Yang, Y. Zhang, Y. Zheng, X. Tian, H. Peng, T. Liu, and B. Han, "Fedfed: Feature distillation against data heterogeneity in federated learning," in *Advances in Neural Information Processing Systems*, 2023, pp. 60 397–60 428.
- [21] A. Woubie and T. Bäckström, "Federated learning for privacy preserving on-device speaker recognition," in *2021 ISCA Symposium on Security and Privacy in Speech Communication*, 2021, pp. 1–5.
- [22] F. Granqvist, M. Seigel, R. van Dalen, Áine Cahill, S. Shum, and M. Paulik, "Improving on-device speaker verification using federated learning with privacy," in *Interspeech 2020*, 2020, pp. 4328–4332.
- [23] H. Wu and P. Wang, "Fast-convergent federated learning with adaptive weighting," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, pp. 1078–1088, 2021.
- [24] Z. Li, T. Lin, X. Shang, and C. Wu, "Revisiting weighted aggregation in federated learning with neural networks," in *Proceedings of the 40th International Conference on Machine Learning*, 2023, pp. 19 767–19 788.
- [25] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1142–1154, 2022.
- [26] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, pp. 926–930, 2018.
- [27] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Interspeech 2017*, 2017, pp. 2616–2620.
- [28] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Interspeech 2018*, 2018, pp. 1086–1090.
- [29] Y. Fan, J. Kang, L. Li, K. Li, H. Chen, S. Cheng, P. Zhang, Z. Zhou, Y. Cai, and D. Wang, "Cn-celeb: A challenging chinese speaker recognition dataset," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7604–7608.
- [30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *The Third International Conference on Learning Representations*, 2015, pp. 1–11.
- [31] J. Kominek and A. W. Black, "The cmu arctic speech databases," in *5th ISCA Workshop on Speech Synthesis (SSW 5)*, 2004, pp. 223–224.
- [32] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 10 708–10 717.
- [33] M. Balamurali, *t-Distributed Stochastic Neighbor Embedding*. Cham: Springer International Publishing, 2020, pp. 1–9.