FedSP: Federated Speaker Verification with Personal Privacy Preservation

Yangqian Wang[1], Yuanfeng Song[2], Di Jiang[2], Ye Ding[3], Xuan Wang[1], Yang Liu[1], and Qing Liao[1,4(✉)]

[1] Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China
19s051041@stu.hit.edu.cn, wangxuan@cs.hitsz.edu.cn,
{liu.yang,liaoqing}@hit.edu.cn
[2] AI Group, WeBank Co., Ltd., Shenzhen, China
{yfsong,dijiang}@webank.com
[3] Dongguan University of Technology, Dongguan, China
dingye@dgut.edu.cn
[4] Peng Cheng Laboratory, Shenzhen 518055, China

Abstract. Automatic speaker verification (ASV) has been widely applied in a variety of industrial scenarios. In ASV, the universal background model (UBM) needs to be trained with a large variety of speaker data so that the UBM can learn the speaker-independent distribution of speech features for all speakers. However, the sensitive information contained in raw speech data is important and private for the speaker. According to the recent European Union privacy regulations, it is forbidden to upload private raw speech data to the cloud server. Thus, a new ASV model needs to be proposed to alleviate data scarcity and protect data privacy simultaneously in the industry. In this work, we propose a novel framework named *Federated Speaker Verification with Personal Privacy Preservation*, or FedSP, which enables multiple clients to jointly train a high-quality speaker verification model and provide strict privacy preservation for speaker. For data scarcity, FedSP is based on the federated learning (FL) framework, which keeps raw speech data on each device and jointly trains the UBM to learn the speech features well. For privacy preservation, FedSP provides more strict privacy preservation than traditional basic FL framework by selecting and hiding sensitive information from raw speech data before jointly training the UBM. Experimental results on two pair speech datasets demonstrate that FedSP has superior performances in terms of data-utility and privacy preservation.

Keywords: Speaker verification · Federated learning · Privacy preservation · Sensitive information

1 Introduction

Automatic speaker verification (ASV) aims to verify whether a speech belongs to a specific speaker based on the speaker's known utterances. ASV has become a

Y. Lai et al. (Eds.): ICA3PP 2021, LNCS 13157, pp. 462–478, 2022.
https://doi.org/10.1007/978-3-030-95391-1_29

common verification way in terms of forensics and security. For instance, many commercial smart devices such as mobile phones, AI speakers, and automotive infotainment systems have adopted machine learning-based ASV for unlocking the system or providing a user-specific service. In universal background model (UBM) based ASV, UBM needs to be trained with a large variety of speech data so that the model can learn the speaker-independent feature distribution of all speakers [4,13,22,25]. However, the raw speech data that contains sensitive information is not allowed to upload to the server [3,5,23] according to European Union General Data Protection Regulation (GDPR) [24], which is a privacy preservation regulation. For example, sharing the raw speech data with the server will not only disclose the identity of the speaker but also make the ASV systems vulnerable to spoofing attacks [27,28]. Thus, a new ASV model needs to be proposed to alleviate data scarcity and protecting data privacy simultaneously in the industry.

ASV systems based on federated learning (FL) [9,11,29] can jointly train a high-quality speaker verification model with multiple clients to alleviate the data scarcity problem via repeatedly communicating model parameters, e.g., model weights or certain statistics, between client and server. However, sharing the parameters of ASV in the FL framework is still not secure enough due to it may also disclose user's privacy when using the raw speech data including sensitive information to training model [30]. For example, through model inversion attack [6] the malicious attackers can reconstruct the speaker biometric templates when getting the parameters uploading from the client, thereby leaking the privacy of speaker. Recently, several cryptographic techniques are merged with the FL framework to overcome the above security and privacy issues [1,18]. However, cryptographic based FL frameworks bring heavy overhead and significantly slow down the verification process, which makes it is unsuitable for ASV. Therefore, it is necessary to develop new privacy preservation technologies for FL based speaker verification systems.

In this work, we propose a novel framework named *Federated Speaker Verification with Personal Privacy Preservation*, or FedSP, that can alleviate data scarcity and provide strict privacy preservation at the same time. For data scarcity, FedSP is based on federated learning (FL) framework that can jointly train the UBM well with multiple clients while retaining the raw speech data on each device. For privacy preservation, except for the default privacy preservation provided by the FL framework, FedSP provides more strict privacy preservation by selecting and hiding the sensitive information from the raw speech data before jointly training the UBM. Especially, sharing the parameters of the model during the model training with sensitive information related to the identity of the speaker may disclose the privacy of speaker. Hence, we try to select and hide the sensitive information of raw speech data and then we formulate the selecting process as an NP-hard integer programming problem. Furthermore, a heuristic greedy search algorithm is proposed to obtain a suboptimal solution for the integer programming problem. In summary, the main contributions of this paper as follows:

- FedSP is the first large-scale distributed speaker verification framework based on GMM-UBM, which can simultaneously overcome data scarcity and privacy leakage issues.
- In order to protect the privacy of each client, FedSP tries to select and hide the sensitive information from raw speech data. FedSP formulates the sensitive information selection process as an NP-hard integer programming problem. In this work, we propose a heuristic greedy search algorithm to solve it.
- Experiments on two pair speech datasets validate the effectiveness of FedSP in data-utility and privacy preservation.

The rest of this paper is organized as follows. In Sect. 2, we briefly review the related literature. In Sect. 3, we detail our proposed FedSP framework, followed by experimental results and analyses in Sect. 4. We finally conclude the work in Sect. 5.

2 Related Work

ASV arises unique privacy concerns, because speech data that are used to training the speaker verification model is closely related to the identity of speakers. Several works have been proposed to overcome the privacy challenges in ASV. And these works can be divided into two categories.

For the first category, they mainly focus on corporating cryptographic encryption or salting techniques with existing ASV methods to protect the privacy of speaker. For example, Pathak and Raj et al. [19] merges GMM with Homomorphic encryption (HE) and Secure Multi-Party Computation (SMPC), and summarize how to perform inference and classification on the encrypted GMM-based speaker verification. Manas et al. [20] applied locality sensitive hashing (LSH) transformation to convert and protect speech signals. Yogachandran et al. [21] cooperated randomization technique to propose an i-vector [4] based verification model to verify speakers' voice in the randomized domain. However, all these methods assume the server needs to collect a large variety of speech data to train the universal background model, which is forbidden by GDPR.

For the second category, these methods are based on the FL framework to jointly training the ASV model with local speech data of users by repeatedly communicating the model weights between a server and a group of users. Recently, FL is widely used in many applications such as topic modeling [12], mobile keyboard prediction [15], and visual object detection [17]. However, there only a few work research the problem of using FL framework to preserve the speakers' privacy in ASV. For example, Filip Granqvist et al. [9] using the side information of local speaker, e.g., gender and emotion, to construct an auxiliary model to enriching the speaker embedding network based on the FL framework. However, this method focuses on protecting the privacy of side information rather than the speech data. Hossein Hosseini et al. [11] proposed a framework for training user authentication models named FedUA, which adopts FL framework and random binary embedding to protect the privacy of raw inputs and embedding vectors based on neural networks, respectively. However, FedUA doesn't provide

any privacy preservation on the model weights, which make it is vulnerable to model inversion attack. In general, the above works can't simultaneously alleviate data scarcity and protect data privacy for ASV. In this paper, we give an in-depth consideration of how to jointly training ASV model while protecting user's privacy in the FL framework.

3 Framework

As demonstrated in Fig. 1, FedSP is composed of three computational modules: client computation, server merging, and client verification. (1) In the client computation module, we propose a heuristic greedy search algorithm to select and hide the sensitive information from the raw speech data, so that the Baum-Welch statistics [4] leaning in the local client will not contain sensitive information. (2) In the server merging module, the server collects the Baum-Welch statistics of each client and assign updated parameters of UBM to all clients. The Baum-Welch Statistics contain acoustic and phonetic variations in speech data, so merging and using it to update the parameters of UBM can make the UBM fits the acoustic channels of the training data. And the privacy of the speaker will not be disclosed via server and transmission attacks, because the Baum-Welch statistics are calculated based on data that does not contain sensitive information. (3) In the client verification module, each client receives the UBM parameters from the server and derives the hypothesized speaker model based on the local raw speech data contained sensitive information. The sensitive information contained in raw speech data is closely related to the identity of the speaker, so each client can achieve a satisfactory verification performance.

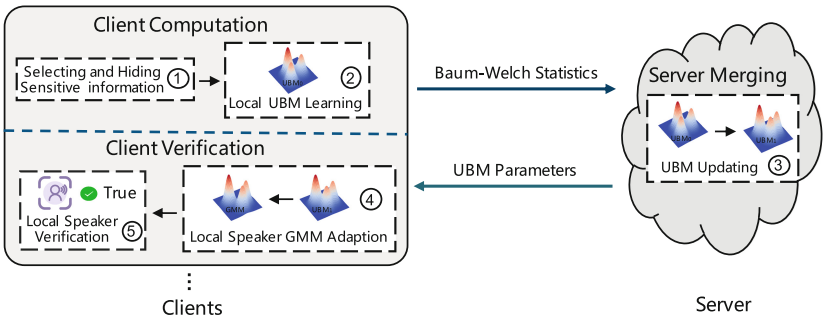


Fig. 1. Federated speaker verification with personal privacy preservation pipeline.

3.1 Client Computation

In this section, we first describe the algorithm for selecting and hiding sensitive information from the raw speech data to protect the privacy of each client. Second, each client executes local UBM learning to get the Baum-Welch statistics without containing sensitive information. At last, clients transmit the local

Baum-Welch statistics without sensitive information to update the parameters of UBM on the server.

Selecting and Hiding Sensitive Information. In this subsection, we present how to select and hide sensitive information (SHS) in detail. In SHS, the sensitive sample is the training vector that closely relates to the sensitive information [5,14,23]. The goal of SHS is to select and hide the sensitive samples from the raw speech data to calculate the Baum-Welch statistics without sensitive information of speaker on local client. Therefore, the speaker's privacy will not be disclosed, when the malicious attackers intercept the Baum-Welch statistics. In GMM-UBM, each training vector of speech data can be divided into different gaussian components in GMM. So we perform sensitive sample selection at the gaussian components level. SHS selects f (i.e., selected fraction) percentage of components containing sensitive information, and then delete the sensitive samples based on that components. In SHS, we first divide the training vectors into different gaussain components according to the posterior probabilities of each training vector. Then we define the *personal confidence score* of each gaussian component and the *distance* between two gaussian components. Finally, the sensitive components selecting process is formulated as an integer programming problem, which is to select a set of gaussian components with maximizing total *personal confident score* and *distance*.

Personal Confidence Score of Gaussian Component. The *personal confidence score* (*pcs*) of a gaussian component is defined as the log-likelihood difference after the vectors belonging to the component are deleted. Therefore, *pcs* can reflect the importance of the component for the speaker. For example, the larger value of *pcs* the component has a higher sensitive level. The personal confidence score pcs_i of the i-th component can be calculated as follows:

$$\begin{aligned} pcs_i &= \frac{\Lambda(\boldsymbol{X}) - \Lambda(\bar{\boldsymbol{X}})}{\Lambda(\boldsymbol{X})}, \\ \Lambda(\boldsymbol{X}) &= log(p(\boldsymbol{X}|\lambda_{spk1})), \\ \Lambda(\bar{\boldsymbol{X}}) &= log(p(\bar{\boldsymbol{X}})|\bar{\lambda}_{spk1})), \end{aligned} \tag{1}$$

where $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_T\}$ is the raw training vectors of each client and $\bar{\boldsymbol{X}}$ is the training vectors that do not include the vectors belong to the i-th component. And $p(\cdot)$ is the probability density function of GMM. λ_{spk1} and $\bar{\lambda}_{spk1}$ are the GMM model derived from UBM_0 based on $\boldsymbol{X}$ and $\bar{\boldsymbol{X}}$, respectively.

Distance between Gaussian Components. It is not enough to select sensitive components based on *pcs*, because we may select similar components with large *pcs* values. For instance, if the selected components are in the same area of the acoustic space, sensitive components located in other areas can still cause privacy leakage. In order to ensure the diversity of the selected sensitive components,

we design a new indicator. In SHS, we recommend ensuring that the distance between all selected components is as large as possible. The *distance* $d_{i,j}$ between two components c_i and c_j is calculated below:

$$d_{i,j} = \sqrt{\sum_{h=1}^{m}(\mu_{i,h} + \mu_{j,h})}, \tag{2}$$

where $\mu_{i,h}$ is the h-th elements of $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_i$ is the mean vector of components c_i in λ_{spk1}. $\mu_{j,h}$ is the h-th elements of $\boldsymbol{\mu}_j$ and $\boldsymbol{\mu}_j$ is the mean vector of components c_j in λ_{spk1}.

Optimization Formulation and Solution. As aforementioned, we want to select those gaussian components with maximizing personal confident score and total distance. The objective function of SHS is defined as below:

$$\max \sum_i pcs_i s_i + \alpha \sum_i \sum_{j \neq i} d_{i,j} s_i s_j, \tag{3}$$

where $s_i = 1$ (or 0) indicates that component c_i is selected (or not) and $\sum_i s_i = f{*}M$, pcs_i denotes the personal confidence score of component c_i and $d_{i,j}$ denotes the distance between component c_i and c_j. α is a trade off parameter and M is the number of gaussian components in GMM. The optimization in Eq. 3 is a typical 0-1 integer programming problem [8,26], which has been shown as a NP-hard problem with $O(2^M)$ search space in exhaustive search. In this work, we propose a greedy component search algorithm (GCS) to solve this problem. Next, we will introduce the GCS in detail.

In GCS, the selected component subset starts from $\boldsymbol{S}_0 = \emptyset$ and adds one component for each step. Supposing that l' is the index of gaussian component selected in the *l-th* step, and component $c_{l'}$ will be added in the component set $\boldsymbol{S}_l$.

$$\boldsymbol{S}_l = \boldsymbol{S}_{l-1} \cup c_{l'}, \tag{4}$$

the components in $\boldsymbol{S}_l = \{c_{i'} | i = 1, ..., l\}$ should meet Eq. (5):

$$\max(\sum_{i=1}^{l} pcs_{i'} + \alpha \sum_{i=1}^{l} \sum_{j \neq i} d_{i',j'}), \tag{5}$$

since $d_{i,j} = d_{j,i}$, $\boldsymbol{S}_l \supset \boldsymbol{S}_{l-1}$ and $\boldsymbol{S}_{l-1} = \{c_{i'} | i = 1, ..., l-1\}$, we can rewrite Eq. (5) as:

$$\max_{l'}((\sum_{i=1}^{l-1} pcs_{i'} + 2\alpha \sum_{i=1}^{l-2} \sum_{j=i+1}^{l-1} d_{i',j'}) + (pcs_{l'} + 2\alpha \sum_{i=1}^{l-1} d_{i',l'})), \tag{6}$$

note that the first part in Eq. (6) is a constant concerning the components selected in the previous $l-1$ steps and the goal becomes to select the component maximizing the second part. So the component $c_{l'}$ selected in l-th step is

$$c_{l'} = \underset{c_i \in \boldsymbol{S}_{M-l-1}}{\arg\max}\ pcs_i + 2\alpha \sum_{j=1}^{l-1} d_{i,j'}, \tag{7}$$

where $\boldsymbol{S}_{M-l-1} = \{c_i | c_i \notin \boldsymbol{S}_{l-1}, i = 1, ..., M\}$.

Selecting and Hiding Sensitive Samples. After getting the component set $\boldsymbol{S}_{f*M}$ that contains sensitive information, we can use it to select sensitive samples. The posterior probability of component c generating the vector $\boldsymbol{x}_t \in \boldsymbol{X}$ is the indicator to select the sensitive samples. The training vector is recognized as a sensitive sample when the posterior probability of the training vector on component c_i is greatest and $c_i \in \boldsymbol{S}_{f*M}$. The sensitive samples will be deleted from the raw training vector $\boldsymbol{X}$ to hide the sensitive information. $\hat{\boldsymbol{X}}$ is the vectors, which delete the sensitive samples from $\boldsymbol{X}$. The posterior probability $r_{c,t}$ of component c generating the training vector $\boldsymbol{x}_t$ is as follows:

$$r_{c,t} = \frac{w_{0,c}\mathcal{N}(\boldsymbol{x}_t | \boldsymbol{\mu}_{0,c}, \boldsymbol{\sigma}_{0,c})}{\sum_{j=1}^{M} w_{0,j}\mathcal{N}(\boldsymbol{x_t} | \boldsymbol{\mu}_{0,j}, \boldsymbol{\sigma}_{0,j})}, \tag{8}$$

where $w_{0,j}, \boldsymbol{\mu}_{0,j}$, and $\boldsymbol{\sigma}_{0,j}$ are the mixture weight, mean vector, and diagonal covariance matrix of j-th component of UBM_0, respectively. M is the number of Gaussian components of UBM_0. the parameters of UBM_0 are collectively represented by the notation $\lambda_0 = \{w_{0,j}, \boldsymbol{\mu}_{0,j}, \boldsymbol{\sigma}_{0,j}\}$, where $j = 1, ..., M$.

The detail of SHS is shown in Algorithm 1.

Algorithm 1. SHS

1: **Input:** raw training vector $\boldsymbol{X}$, pre-trained model UBM_0. $\boldsymbol{PCS} = \{pcs_i | i = 1, ..., M\}$, $\boldsymbol{D} = \{d_{i,j} | i \neq j, i, j \in \{1, ..., M\}\}$, and selected fraction f;
2: **Output:** $\hat{\boldsymbol{X}}$;
3: Initially $\boldsymbol{S}_0 = \emptyset$ and $\hat{\boldsymbol{X}} = \emptyset$;
4: **for** $i = 1$ to $f * M$ **do**
5: search for the new component $c_{i'}$ according to Eq.(7);
6: update the other components according to their distance with $c_{i'}$:

$$pcs_j \leftarrow pcs_j + 2\alpha \times d_{i',j}, \quad j \neq i'$$

7: add $c_{i'}$ to the set $\boldsymbol{S}_i$, $\boldsymbol{S}_i = \boldsymbol{S}_{i-1} \cup c_{i'}$;
8: **end for**
9: **for** $t = 1$ to T **do**
10: get the component c', which has the greatest probability of $\boldsymbol{x}_t$ come from it:

$$c' = \underset{c \in \{1, ..., M\}}{\arg\max}\ r_{c,t}$$

11: **if** $c' \in \boldsymbol{S}_{f*M}$ **then**
12: $\hat{\boldsymbol{X}} = \hat{\boldsymbol{X}} \cup \boldsymbol{x}_t$;
13: **end if**
14: **end for**

Local UBM Learning. To alleviate the problem of privacy leakage, each local client is the workhorse for UBM learning in FedSP. Especially, FedSP calculates the Baum-Welch statistics on local clients, and that statistics will be uploaded to the server. Thus the Baum-Welch Statistics contain acoustic and phonetic variations in local speech data. At the same time, to prevent the privacy leakage by the Baum-Welch statistics each client using the training vectors processed by SHS, i.e., $\hat{\boldsymbol{X}}$, instead of raw training vector $\boldsymbol{X}$ to do UBM learning. Based on the global UBM_0 received from the server and $\hat{\boldsymbol{X}} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_{\hat{T}}\}$, we first compute the posterior probability $r_{c,t}$ of $\boldsymbol{x}_t$ on the c-th component of UBM_0 as Eq. (8). Then $r_{c,t}$ is used to calculate the Baum-Welch statistics $\hat{\boldsymbol{r}}_c^i$ and $\hat{\boldsymbol{z}}_c^i$ as follows:

$$\hat{\boldsymbol{r}}_c^i = \sum_{t=1}^{\hat{T}} r_{c,t}[\mathbf{1}], \quad \hat{\boldsymbol{z}}_c^i = \sum_{t=1}^{\hat{T}} r_{c,t} \tag{9}$$

where $[\mathbf{1}]$ is the vector created by filling unit elements and the dimension of it is m. i is the index of client. $\hat{\boldsymbol{r}}_c^i$ and $\hat{\boldsymbol{z}}_c^i$ will be sent to the server for UBM updating. The Baum-Welch statistics do not contain sensitive information, so the privacy of the client will not be disclosed.

3.2 Server Merging: UBM Updating

To alleviate the problem of data scarcity, the server updates the parameters of UBM to model the general acoustic features well. And the UBM updating is based on the Baum-Welch statistics uploaded from local clients. UBM updating on the server needs the help of $\hat{\boldsymbol{r}}_c^i$ and $\hat{\boldsymbol{z}}_c^i$, which are uploaded from the client i. The Baum-Welch statistics are firstly pooled together to form the pooled statistics:

$$\bar{\boldsymbol{r}}_c = \sum_{i=1}^{K} \hat{\boldsymbol{r}}_c^i, \quad \bar{\boldsymbol{z}}_c = \sum_{i=1}^{K} \hat{\boldsymbol{z}}_c^i, \tag{10}$$

where $\hat{\boldsymbol{r}}_c^i$ and $\hat{\boldsymbol{z}}_c^i$ denote the Baum-Welch statistics transmitted from client i with privacy preservation provided by SHS. And K is the number of training clients participating in UBM updating.

Then the server uses the pooled statistics to update the parameters of UBM_0 to get UBM_1, $\lambda_1 = \{w_{1,j}, \boldsymbol{\mu}_{1,j}, \boldsymbol{\sigma}_{1,j}\}$ and $j = 1, ..., M$, according to the following formula:

$$\boldsymbol{\mu}_{1,c} = \frac{\bar{\boldsymbol{z}}_c + \frac{\boldsymbol{\sigma}_{0,c}}{\hat{\sigma}_{\text{UBM}}} \boldsymbol{\mu}_{0,c}}{\bar{\boldsymbol{r}}_c + \frac{\boldsymbol{\sigma}_{0,c}}{\hat{\sigma}_{\text{UBM}}}}, \tag{11}$$

where $\hat{\sigma}_{\text{UBM}}$ represents the covariance prior to indicate that the prior is used for UBM update.

3.3 Client Verification

Local Speaker GMM Adaption. In FedSP, each client derives the speaker's GMM model by adapting the parameters of the UBM_1 based on the raw training vectors $\boldsymbol{X}$. The speaker's GMM model contains sensitive information, which is closely related to the identity of speaker, so each client can verify the identity of a new speech accurately. $\boldsymbol{r}_c$ and $\boldsymbol{z}_c$ calculated based on $\boldsymbol{X}$ according Eq. (9) are the key parameters for adapting the speaker's GMM. Then, $\boldsymbol{r}_c$ and $\boldsymbol{z}_c$ are used to adapt the c-th component of the speaker's GMM, $\lambda_{\text{s}} = \{w_{s,j}, \boldsymbol{\mu}_{s,j}, \boldsymbol{\sigma}_{s,j}\}$ and $j = 1, ..., M$, with the following equation:

$$\boldsymbol{\mu}_{s,c} = \frac{\boldsymbol{z}_c + \frac{\boldsymbol{\sigma}_{1,c}}{\hat{\sigma}_{\text{spk}}}\boldsymbol{\mu}_{1,c}}{\boldsymbol{r}_c + \frac{\boldsymbol{\sigma}_{1,c}}{\hat{\sigma}_{\text{spk}}}}, \tag{12}$$

where $\hat{\sigma}_{\text{spk}}$ indicates the prior for speaker model adaptation.

Local Speaker Verification. Each client can use the global UBM_1 and their GMM model λ_{s} to compute the log-likelihood ratio to verify the identity of new speeches [22]. Given a new segment of speech $\boldsymbol{X_{test}}$ and the hypothesized speaker S, the task of ASV is to determine if $\boldsymbol{X_{test}}$ is spoken by S. Mathematically, speaker verification can be formulated as:

$$\begin{aligned} \Lambda(\boldsymbol{X_{test}}) &= \log p(\boldsymbol{X_{test}}|\lambda_s) - \log p(\boldsymbol{X_{test}}|\lambda_1) \\ \begin{cases} \Lambda(\boldsymbol{X_{test}}) \geq \theta & \boldsymbol{X_{test}} \text{ is from the hypothesized speaker } S \\ \Lambda(\boldsymbol{X_{test}}) < \theta & \boldsymbol{X_{test}} \text{ is not from the hypothesized speaker } S, \end{cases} \end{aligned} \tag{13}$$

3.4 FedSP Workflow

The algorithm of FedSP is presented in Algorithm 2. During the client computation stage, each client first selects and hides sensitive information from the raw speech data according to Algorithm 1. Then each client calculates the Baum-Welch statistics based on the training vectors without sensitive information and uploads the statistics to the server. After that, the server collects the Baum-Welch statistics of each client and assigns update parameters of UBM to all clients. Finally, each client receives the parameters of UBM and uses it to derive the speaker's GMM model based on the raw speech data.

Algorithm 2. FedSP

function ServerMergingUBMUpdating( ):
  initialize UBM_0;
  for each Guassian Component c from 1 to M in UBM_0 **do**
    $\bar{\boldsymbol{r}}_c = 0$, $\bar{\boldsymbol{z}}_c = 0$;
    for each local speaker i from 1 to K **do**
      $\hat{r}_c^i, \hat{z}_c^i$ = ClientComputation(UBM_0,f,c);
      $\bar{\boldsymbol{r}}_c = \bar{\boldsymbol{r}}_c + \hat{r}_c^i$, $\bar{\boldsymbol{z}}_c = \bar{\boldsymbol{z}}_c + \hat{z}_c^i$;
    end for
    update UBM from UBM_0 to UBM_1 according Eq.(11);
  end for
  return UBM_1;
end function

function ClientComputation(UBM_0,f,c):
  get the training speech of speaker i, $\boldsymbol{X}_i$, and processed by SLPP to get $\hat{\boldsymbol{X}}_i$;
  calculate $\boldsymbol{PCS}$ and $\boldsymbol{D}$ according Eq.(1) and Eq.(2), respectively;
  $\hat{\boldsymbol{X}}$ =SHS($\boldsymbol{X}$,UBM_0,$\boldsymbol{PCS}$,$\boldsymbol{D}$,f);
  calculate $\hat{\boldsymbol{r}}_c$ and $\hat{\boldsymbol{z}}_c$ based on $\hat{\boldsymbol{X}}$ just as Eq.(9);
  return $\hat{\boldsymbol{r}}_c$, $\hat{\boldsymbol{z}}_c$;
end function

4 Experiments

In this section, we first introduce datasets and corresponding settings used in experiments. Second, we introduce the evaluation metrics used in this work. Third, we conduct experiments to verify the effectiveness of FedSP in solving data scarcity and privacy preservation problems. Finally, we state the privacy preservation capability and do the parameter study of FedSP.

4.1 Dataset and Configurations

We conduct our experiments using four real-world speech datasets which include SUD12 [16], ST-AEDS-20180100_1[1], Aishell [2] and TIMIT [7]. The SUD12 dataset contains 61 Chinese speakers and each speaker produces 100 utterances. The ST-AEDS-20180100_1 dataset contains 10 native English speakers and each speaker records about 350 utterances under a silent in-door environment using cellphones. The Aishell dataset contains 400 Chinese speakers under 11 domains such as science & technology, finance, and sport. The TIMIT speech dataset contains 630 speakers of eight major dialects of American English and each speaker reads ten phonetically rich sentences.

The four datasets are divided into two pairs. The first pair of datasets consists of SUD12 and Aishell datasets, which are Chinese speech datasets. In this pair,

[1] http://www.openslr.org/45/.

we use the SUD12 to pre-training the UBM on the server and each client gets one speaker's data of Aishell. The second pair of datasets consists of ST-AEDS-20180100_1 and TIMIT datasets, which are English speech datasets. In this pair, we use the ST-AEDS-20180100_1 to pre-training the UBM on the server and each client gets one speaker's data of TIMIT.

In this experiment, we first use Mel Frequency Cepstral Coef-ficients (MMFCs) method to extract features of speakers from speech dataset. Specially, the dimension of MMFCs is 60 (20 basic + first order + second order) using a 25 ms Hamming window with 10 ms shift. First order and second order are calculated using a 2-frame window. Second, FedSP trains UBM with 256 Gaussian components, i.e., $M = 256$. The value of $\hat{\sigma}_{\mathrm{SPK}} = 0.5$ and $\hat{\sigma}_{\mathrm{UBM}} = 0.07$ in local speaker GMM adapting module and server merging module, respectively. $\alpha = 0.005$ is used to balance the two terms in SHS. Third, $K = 10,\ 100$ and 200 speakers of different accents are selected randomly from Aishell and TIMIT, respectively, and assigned to different clients. For each client, three sentences were used to jointly training the UBM. At last, to test the performance of the FedSP, we randomly select other 200 speakers that are different from the K speakers used to jointly the UBM.

4.2 Evaluation Metrics

Verification Metric. The metric employed for performance evaluation is the Equal Error Rate (EER). EER is the point that the False Acceptance Rate (FAR) and False Rejection Rate (FRR) become equal. FRR and FAR measure the classification error for target and non-target trials, respectively.

Privacy Metric. The metric employed for measuring the capability of privacy preservation is Kullback-Leibler Distance (KLD) [10]. On the one hand, selecting and hiding the sensitive samples from raw speech data inevitably reduce the amount of data used, thereby increasing EER. Thus using EER to measure the effectiveness of privacy preservation is not rational. On the other hand, FedSP changes the real distribution of the speaker's GMM model by hiding the sensitive samples from the raw training vectors. Therefore, the speaker's real biometric template cannot be reconstructed, when the malicious attackers intercept the Baum-Welch statistics. The smaller the KLD between different speaker's model reconstructed by the server, the smaller the difference between the Baum-Welch statistics calculated by different clients. And the Baum-Welch statistics calculated on local clients contain less sensitive information about the speaker.

4.3 Verification Performance

In FedSP, we jointly train the UBM with multiple clients. In this section, we compare two classical framework learning scenarios in two pair of speech datasets for demonstrating the effectiveness of FedSP in data utility. The details are as follows:

- **Baseline**: This baseline is straightforward. Participants derive their own speaker model based on the pre-trained model without any collaboration.
- **Center**: It is a classical training process for ASV. In this case, the speaker's raw speech data are centrally collected and trained in the server first. Then, the server transmits the parameters of well-trained UBM to the clients and clients derive the speaker's model based on it locally.

The comparative results over two datasets between FedSP and the above method for speaker verification are summarized in Table 1. For the Baseline, we found it is the worst since it cannot handle the data scarcity problem and local clients directly apply the pre-training UBM to derive the speaker's model. For Center and FedSP, they remarkably outperform the Baseline and have lower EER scores as the number of training clients increases. This is because they solve the data scarcity problem by using the users' speech data to jointly train UBM. The performance of FedSP is slightly worse than Center. However Center needs to upload all raw speech data of each client to the central server and FedSP only uploads the Baum-Welch statistics without sensitive information to the central server. Hence, only FedSP can protect the privacy of participants with remain satisfied speaker verification performances.

Table 1. EER scores of three approaches on Aishell and TIMIT datasets.

Method	Baseline	Center			FedSP		
Training Clients K	N/A	10	100	200	10	100	200
Aishell	5.67	2.36	0.57	0.58	2.25	1.5	1.39
TIMIT	7.26	4.84	3.13	2.61	4.88	3.58	3.36

4.4 Effectiveness of SHS in FedSP

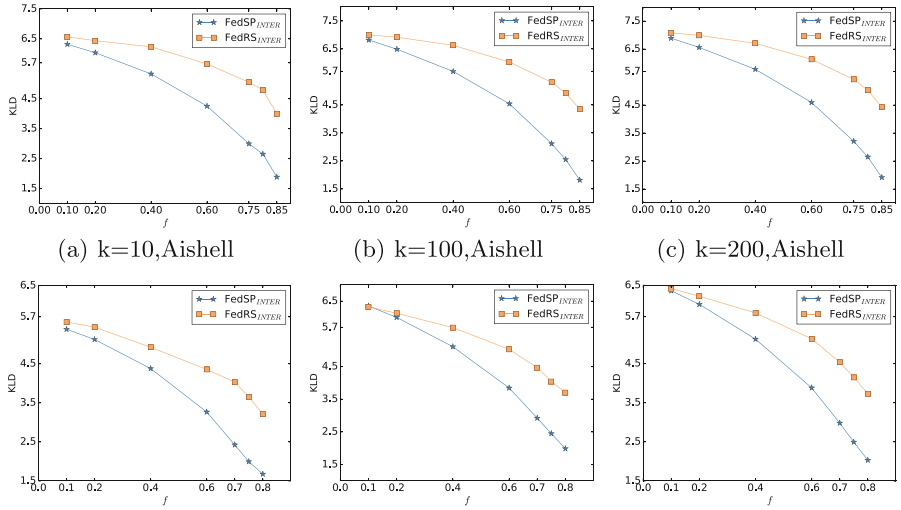
The main contribution of this work is that FedSP can protect the privacy of speaker, which is done by hiding the sensitive information from the raw training vectors. The way to show the effectiveness of SHS is to verify whether the Baum-Welch statistics collected from clients will disclose the privacy feature of speakers. When SHS is effective, the Baum-Welch statistics learning by different clients should be more and more alike, and vice versa. And we design the following experiment to verify the effectiveness of SHS:

- **FedSP**$_{INTER}$: In this method, we first use the Baum-Welch statistics that are calculated on the training vectors processed by SHS to reconstruct the GMM model for each client. Then, the KLD is calculated between two different clients' GMM models.
- **FedRS**$_{INTER}$: When the SHS of FedSP is replaced by RS (random sampling), the new framework is named as FedRS. So, in FedSR we randomly delete the same number of vectors as SHS no matter whether the vectors are sensitive samples or not. And the KLD of **FedRS**$_{INTER}$ is calculated as **FedSP**$_{INTER}$.

Figure 2 compares the KLD of the above two methods on Aishell and TIMIT datasets. It can be observed that the KLD of FedSP$_{INTER}$ and FedRS$_{INTER}$ decreases as the selected fraction f increases. This is because that the more sensitive samples are deleted, the fewer sensitive information is included. So the Similarity between the reconstructed GMM models becomes larger. Except for that, the gap between FedSP$_{INTER}$ and FedRS$_{INTER}$ increases, which indicates that SHS is more effective than RS in selecting and hiding sensitive information.

4.5 Privacy Preservation Capability

In this section, we need to define the privacy preservation capability of FedSP and determine when FedSP can provide strict privacy preservation. The stricter the privacy protection, the less sensitive information contained in the Baum-Welch statistics, which are calculated on the training vectors processed by SHS. We say that FedSP can provide strict privacy preservation for speakers when malicious attackers can't intercept sensitive information about the speakers from the Baum-Welch statistics. Except for FedSP$_{INTER}$, which calculates the KLD of different speakers, we also design a new method named FedSP$_{INTRA}$ to calculate the KLD of the same speaker.



(a) k=10,Aishell (b) k=100,Aishell (c) k=200,Aishell

(d) k=10,TIMIT (e) k=100,TIMIT (f) k=200,TIMIT

Fig. 2. The effectiveness of FedSP on Aishell and TIMIT datasets.

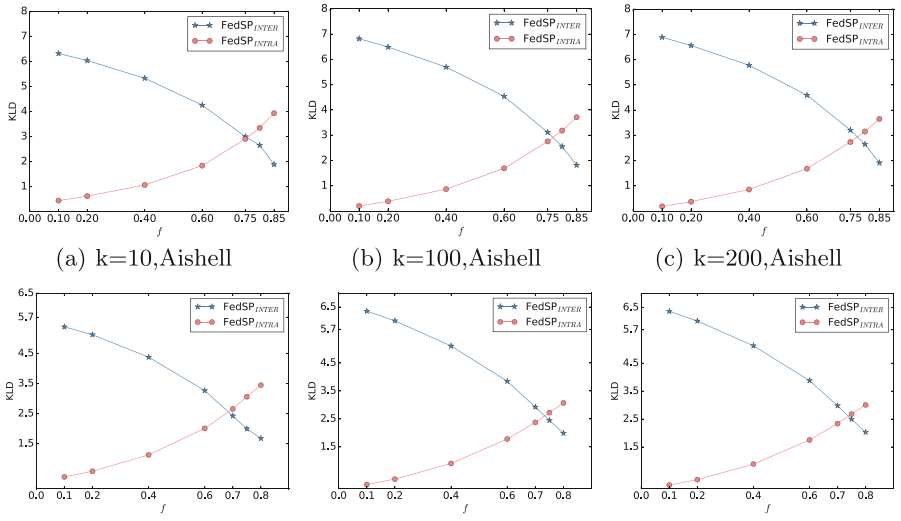
– **FedSP**$_{INTRA}$: For each client, we calculate the KLD between the real speaker's GMM model get by Eq. (12) and the reconstructed GMM model.

Figure 3 compares FedSP$_{INTER}$ and FedSP$_{INTRA}$ on Aishell and TIMIT datasets. As shown in the Fig. 3, the KLD of FedSP$_{INTER}$ decreases and the

KLD of FedSP$_{INTRA}$ increases as the selected fraction f increase. This is because that as f increase, SHS select and hiding more and more sensitive information for each client. So the sensitive information contained in the statistics which are upload from clients gradually decreases. Besides that, it also can be found that the KLD of FedSP$_{INTER}$ and FedSP$_{INTRA}$ overlap for Aishell and TIMIT dataset, when f is around 0.75. And We state that FedSP can provide strict privacy preservation when the f is greater than or equals to 0.75 for Aishell and TIMIT dataset. In this case, the Baum-Welch statistics of different speakers that are upload to the server are similar and do not contain sensitive information. So the Baum-Welch statistics can not disclose the privacy of speakers.

4.6 Parameter Study

FedSP has two main parameters training clients K and selected fraction f. The verification performance of FedSP is determined by these two parameters together. So in this subsection, we investigate the verification performance of FedSP under different values of these two parameters in Aishell and TIMIT datasets. As shown in Fig. 4, the EER score of FedSP decreases with more training clients and increases with the value of selected fraction increases in both Aishell and TIMIT datasets. In each plot, we show the EER score for Baseline, which is independent of the x-axis. We state that FedSP can provide strict privacy preservation when the f is greater than or equals to 0.75 for Aishell and TIMIT dataset In Sect. 4.5. And in Fig. 4, we also find that $f = 0.75$ for Aisell and TIMIT is a turning point of verification performance. Before the turning point, the EER increases slowly. After the turning point, the EER increases

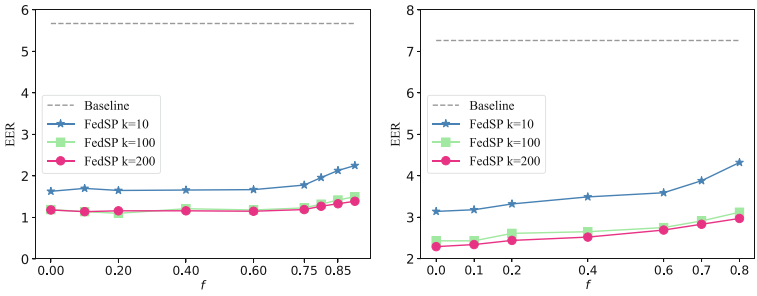


(a) k=10,Aishell (b) k=100,Aishell (c) k=200,Aishell

(d) k=10,TIMIT (e) k=100,TIMIT (f) k=200,TIMIT

Fig. 3. The privacy preservation capability of FedSP on Aishell and TIMIT datasets.

rapidly. What's more, FedSP still outperforms Baseline when FedSP provides strict privacy preservation. So FedSP can offer an attractive trade-off between data utility and privacy.



(a) Aishell (b) TIMIT

Fig. 4. Verification performance of FedSP with different selection fraction f and training clients K on Aishell and TIMIT datasets.

5 Conclusion

This paper presents a novel framework named FedSP that allows multiple mobile clients to cooperatively train the UBM to solve data scarcity problem while providing strict privacy preservation for participants. In the jointly training procedure, each client executes local learning on the training vectors without sensitive information so that the Baum-Welch statistics learning in local clients will not disclose the privacy of speaker. The sensitive information selection process is formulated as an integer programming problem, and we proposed a heuristic greedy search algorithm to tackling the problem. With the federated architecture in FedSP, a server and a series of clients jointly train a UBM that can model the speaker-independent distribution feature well. To justify our model, we conducted extensive experiments over two speech datasets. We notice that the experimental results support the following points: first, FedSP can alleviate the data scarcity problem; second, through selecting and hiding the sensitive information, FedSP can provide strict privacy preservation.

Acknowledgement. This work is supported in part by the National Natural Science Foundation of China under grant No. 62076079 and No. 61976051, the Guangdong Major Project of Basic and Applied Basic Research under grant No. 2019B030302002.

References

1. Aono, Y., Hayashi, T., Wang, L., Moriai, S., et al.: Privacy-preserving deep learning via additively homomorphic encryption. IEEE Trans. Inf. Forensics Secur. **13**(5), 1333–1345 (2017)

2. Bu, H., Du, J., Na, X., Wu, B., Zheng, H.: Aishell-1: an open-source mandarin speech corpus and a speech recognition baseline. In: 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment, pp. 1–5 (2017)
3. Chowdhury, M.F.R., Selouani, S.A., O'Shaughnessy, D.: Text-independent distributed speaker identification and verification using GMM-UBM speaker models for mobile communications. In: 10th International Conference on Information Science, Signal Processing and their Applications, pp. 57–60 (2010)
4. Dehak, N., Kenny, P.J., Dehak, R., Dumouchel, P., Ouellet, P.: Front-end factor analysis for speaker verification. IEEE Trans. Audio Speech Lang. Process. **19**(4), 788–798 (2010)
5. Faltlhauser, R., Ruske, G.: Improving speaker recognition using phonetically structured gaussian mixture models. In: Seventh European Conference on Speech Communication and Technology (2001)
6. Fredrikson, M., Jha, S., Ristenpart, T.: Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 1322–1333 (2015)
7. Garofolo, J.S.: Timit acoustic phonetic continuous speech corpus. Linguistic Data Consortium (1993)
8. Geng, X., Liu, T.Y., Qin, T., Li, H.: Feature selection for ranking. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 407–414 (2007)
9. Granqvist, F., Seigel, M., van Dalen, R.C., Cahill, Á., Shum, S., Paulik, M.: Improving on-device speaker verification using federated learning with privacy. In: Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25–29 October 2020, pp. 4328–4332 (2020)
10. Hershey, J.R., Olsen, P.A.: Approximating the kullback leibler divergence between gaussian mixture models. In: IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 4, pp. IV–317 (2007)
11. Hosseini, H., Yun, S., Park, H., Louizos, C., Soriaga, J., Welling, M.: Federated learning of user authentication models. arXiv preprint arXiv:2007.04618 (2020)
12. Jiang, D., et al.: Federated topic modeling. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 1071–1080 (2019)
13. Kumari, T.R.J., Jayanna, H.S.: i-vector-based speaker verification on limited data using fusion techniques. J. Intell. Syst. **29**(1), 565–582 (2020)
14. Lei, Y., Scheffer, N., Ferrer, L., McLaren, M.: A novel scheme for speaker recognition using a phonetically-aware deep neural network. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1695–1699 (2014)
15. Leroy, D., Coucke, A., Lavril, T., Gisselbrecht, T., Dureau, J.: Federated learning for keyword spotting. In: ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6341–6345 (2019)
16. Li, L., Wang, D., Zhang, C., Zheng, T.F.: Improving short utterance speaker recognition by modeling speech unit classes. IEEE/ACM Trans. Audio Speech Lang. Process. **24**(6), 1129–1139 (2016)
17. Liu, Y., et al.: FedVision: an online visual object detection platform powered by federated learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 13172–13179 (2020)

18. Liu, Y., Kang, Y., Xing, C., Chen, T., Yang, Q.: A secure federated transfer learning framework. IEEE Intell. Syst. **35**(4), 70–82 (2020)
19. Pathak, M.A., Raj, B.: Privacy-preserving speaker verification and identification using gaussian mixture models. IEEE Trans. Audio Speech Lang. Process. **21**(2), 397–406 (2012)
20. Pathak, M.A., Raj, B.: Privacy-preserving speaker verification as password matching. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1849–1852 (2012)
21. Rahulamathavan, Y., Sutharsini, K.R., Ray, I.G., Lu, R., Rajarajan, M.: Privacy-preserving ivector-based speaker verification. IEEE/ACM Trans. Audio Speech Lang. Process. **27**(3), 496–506 (2019)
22. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker verification using adapted gaussian mixture models. Digital Signal Process. **10**(1–3), 19–41 (2000)
23. Reynolds, D.A., Rose, R.C.: Robust text-independent speaker identification using gaussian mixture speaker models. IEEE Trans. Speech Audio Process. **3**(1), 72–83 (1995)
24. Voss, W.G.: European union data privacy law reform: general data protection regulation, privacy shield, and the right to delisting. Bus. Lawyer **72**(1), 221–234 (2016)
25. Wang, J., Wang, D., Wu, X., Zheng, T.F., Tejedor, J.: Sequential model adaptation for speaker verification. In: 14th Annual Conference of the International Speech Communication Association, pp. 2460–2464 (2013)
26. Wolsey, L.A., Nemhauser, G.L.: Integer and Combinatorial Optimization, vol. 55. Wiley, Hoboken (1999)
27. Wu, Z., Evans, N., Kinnunen, T., Yamagishi, J., Alegre, F., Li, H.: Spoofing and countermeasures for speaker verification: a survey. Speech Commun. **66**, 130–153 (2015)
28. Wu, Z., et al.: ASVspoof: the automatic speaker verification spoofing and countermeasures challenge. IEEE J. Sel. Top. Signal Process. **11**(4), 588–604 (2017)
29. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: concept and applications. ACM Trans. Intell. Syst. Technol. **10**(2), 1–19 (2019)
30. Zhu, L., Liu, Z., Han, S.: Deep leakage from gradients. In: Advances in Neural Information Processing Systems, pp. 14774–14784 (2019)