# PRIVACY ATTACKS FOR AUTOMATIC SPEECH RECOGNITION ACOUSTIC MODELS IN A FEDERATED LEARNING FRAMEWORK

*Natalia Tomashenko[1], Salima Mdhaffar[1], Marc Tommasi[2], Yannick Estève[1], Jean-François Bonastre[1]*

[1] LIA, Avignon Université, France
[2] Université de Lille, CNRS, Inria, Centrale Lille, UMR 9189 - CRIStAL, Lille, France

## ABSTRACT

This paper investigates methods to effectively retrieve speaker information from the personalized speaker adapted neural network acoustic models (AMs) in automatic speech recognition (ASR). This problem is especially important in the context of federated learning of ASR acoustic models where a global model is learnt on the server based on the updates received from multiple clients. We propose an approach to analyze information in neural network AMs based on a neural network footprint on the so-called *Indicator* dataset. Using this method, we develop two attack models that aim to infer speaker identity from the updated personalized models without access to the actual users' speech data. Experiments on the TED-LIUM 3 corpus demonstrate that the proposed approaches are very effective and can provide equal error rate (EER) of 1–2%.

***Index Terms***— Privacy, federated learning, acoustic models, attack models, speech recognition, speaker verification

## 1. INTRODUCTION

Federated learning (FL) for automatic speech recognition (ASR) has recently become an active area of research [1–6]. To preserve the privacy of the users' data in the FL framework, the model is updated in a distributed fashion instead of communicating the data directly from clients to a server.

Privacy is one of the major challenges in FL [7, 8]. Sharing model updates, i.e. gradient information, instead of raw user data aims to protect user personal data that are processed locally on devices. However, these updates may still reveal some sensitive information to a server or to a third party [9, 10]. According to recent research, FL has various privacy risks and may be vulnerable to different types of attacks, i.e. membership inference attacks [11] or generative adversarial network (GAN) inference attacks [12]. Techniques to enhance the privacy in a FL framework are mainly based on two categories [8]: secure multiparty computation [13] and differential privacy [14]. Encryption methods [15, 16] such as fully homomorphic encryption [16] and secure multiparty computation perform computation in the encrypted domain. These methods increase computational complexity. In a FL framework, this increase is not so significant compared to standard centralized training, since only the transmitted parameters need to be encrypted instead of large amounts of data, however with an increased number of participants, computational complexity becomes a critical issue. Differential privacy methods preserve privacy by adding noise to users' parameters [14, 17], however such solutions may degrade learning performance due to the uncertainty they introduce into the parameters.

Alternative methods to privacy protection for speech include deletion methods [18] that are meant for ambient sound analysis, and anonymization [19, 20] that aims to suppress personally identifiable information in the speech signal keeping unchanged all other attributes. These privacy preservation methods can be combined and integrated in a hybrid fashion into a FL framework.

Despite the recent interest in FL for ASR and other speech-related tasks such as keyword spotting [21, 22], emotion recognition [23], and speaker verification [24], there is a lack of research on vulnerability of ASR acoustic models (AMs) to privacy attacks in a FL framework. In this work, we make a step towards this direction by analyzing speaker information that can be retrieved from the personalized AM locally updated on the user's data. To achieve this goal, we developed two privacy attack models that operate directly on the updated model parameters without access to the actual user's data. Parameters of neural network (NN) personalized AMs contain a wealth of information about the speakers [25]. In this paper, we propose novel methods to efficiently and easily retrieve speaker information from the adapted AMs. The main idea of the proposed methods is to use an external *Indicator* dataset to analyze the footprint of AMs on this data. Another important contribution of this work is understanding how the speaker information is distributed in the adapted NN AMs.

This paper is structured as follows. Section 2 briefly introduces a considered FL framework for AM training. Section 3 describes the privacy preservation scenario and proposes two attack models. Experimental evaluation is presented in Section 4. We conclude in Section 5.

## 2. FEDERATED LEARNING FOR ASR ACOUSTIC MODELS

We consider a classical FL scenario where a global NN AM is trained on a server from the data stored locally on multiple remote devices [7]. The training of the global model is performed under the constraint that the training speech data are stored and processed locally on the user devices (clients), while only model updates are transmitted to the server from each client. The global model is learnt on the server based on the updates received from multiple clients. The FL in a distributed network of clients is illustrated in Figure 1. First, an initial global speech recognition AM $W_g$ is distributed to the group of devices of $N$ users (speakers). Then, the initial global model is run on every user $s_i$ ($i \in 1..N$) device and updated locally on the private user data. The updated models $W_{s_i}$ are then transmitted to the server where they are aggregated to obtain a new global model $W_g^*$. Typically, the personalized updated models are aggregated using federated averaging and its variations [13, 26]. Then, the updated global model $W_g^*$ is shared with the clients. The

process restarts and loops until convergence or after a fixed number of rounds. The utility and training efficiency of the FL AMs have been successfully studied in recent works [1–6], and these topics are beyond the scope of the current paper. Alternatively, we focus on the privacy aspect of this framework.
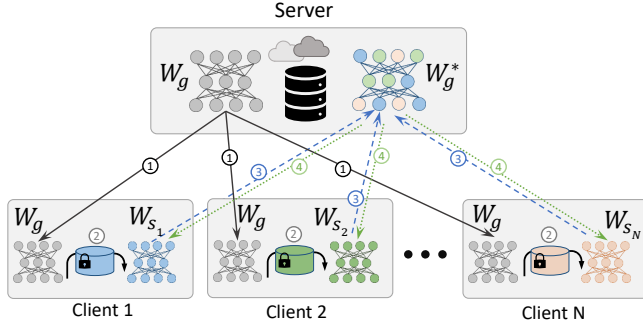


**Fig. 1**. Federated learning in a distributed network of clients: 1) Download of the global model $W_g$ by clients. 2) Speaker adaptation of $W_g$ on the local devices using user private data. 3) Collection and aggregation of multiple personalized models $W_{s_1},...,W_{s_N}$ on the server. 4) Sharing the resulted model $W_g^*$ with the clients.

# 3. ATTACK MODELS

In this section, we describe the privacy preservation scenario and present two attack models.

## 3.1. Privacy preservation scenario

Privacy preservation is formulated as a game between *users* who share some data and *attackers* who access this data or data derived from it and aim to infer information about the users [19]. To preserve the user data, in FL, there is no speech data exchange between a server and clients, only model updates are transmitted between the clients and server (or between some clients). Attackers aim to attack users using information owned by the server. They can get access to some updated personalized models.

In this work, we assume that an attacker has access to the following data:

- An initial global NN AM $W_g$;
- A personalized model $W_s$ of the target speaker $s$ who is enrolled in the FL system. The corresponding personalized model was obtained from the global model $W_g$ by fine-tuning $W_g$ using speaker data. We consider this model as *enrollment* data for an attacker.
- Other personalized models of non-target and target speakers: $W_{s_1},...,W_{s_N}$. We will refer to these models as *test trial* data.

The attacker's objective is to conduct an automatic speaker verification (ASV) task by using the enrollment data model in the form of $W_s$ and test trial data in the form of models $W_{s_1},...,W_{s_N}$.

## 3.2. Attack models

The motivation of the proposed approaches is based on the hypothesis that we can capture information about the identity of speaker $s$ from the corresponding speaker-adapted model $W_s$ and the global model $W_g$ by comparing the outputs of these two neural AMs taken

from hidden layers $h$ on some speech data. We will refer to this speech data as *Indicator* data. Note, that the *Indicator* data is not related to any test or AM training data and can be chosen arbitrarily from any speakers.

### 3.2.1. Attack model A1

The ASV task with the proposed attack model is performed in several steps as illustrated in Figure 2. Let denote a set of utterances in the *Indicator* dataset $\mathbb{I}$ as $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_J \in \mathbb{I}$; a sequence of vectors in utterance $\boldsymbol{u}_j$ as $\{u_j^1,\ldots,u_j^{T_j}\}$; a set of personalized models as $W_{s_1},\ldots,W_{s_N} \in \mathbb{W}$; and an identifier of a hidden layer in the global or personalized AM as $h$.

1. $\forall\, W_{s_i} \in \mathbb{W}, \forall\, \boldsymbol{u}_j \in \mathbb{I}$ we compute activation values from the layer $h$ for model pairs: $W_{s_i}^h(\boldsymbol{u}_j) = \{w_{s_i,j}^{h,t}\}_{t=1}^{T_j}$ and $W_g^h(\boldsymbol{u}_j) = \{w_{g,j}^{h,t}\}_{t=1}^{T_j}$, and per-frame differences between corresponding outputs:

$$\boldsymbol{\Delta}_{s_i}^h(\boldsymbol{u}_j) = \{\Delta_{s_i,j}^{h,t}\}_{t=1}^{T_j}, \tag{1}$$

where $\Delta_{s_i,j}^{h,t} = w_{s_i,j}^{h,t} - w_{g,j}^{h,t}, t \in 1..T_j$.

2. For each personalized model, we compute mean and standard deviation vectors for $\Delta_{s_i,j}^{h,t}$ over all speech frames in the *Indicator* dataset $\mathbb{I}$:

$$\boldsymbol{\mu}_{s_i}^h = \frac{\sum_{j=1}^{I}\sum_{t=1}^{T_j}\Delta_{s_i,j}^{h,t}}{\sum_{j=1}^{I}T_j}, \tag{2}$$

$$\boldsymbol{\sigma}_{s_i}^h = \left(\frac{\sum_{j=1}^{I}\sum_{t=1}^{T_j}(\Delta_{s_i,j}^{h,t} - \boldsymbol{\mu}_{s_i}^h)^2}{\sum_{j=1}^{I}T_j}\right)^{\frac{1}{2}} \tag{3}$$

3. For a pair of personalized models $W_{s_i}$ and $W_{s_k}$, we compute a similarity score $\rho$ at hidden level $h$ on the *Indicator* dataset based on the $\boldsymbol{L}_2$-normalised Euclidean distance between the corresponding vector pairs for means and standard deviations:

$$\rho(W_{s_i}^h, W_{s_k}^h) = \alpha_\mu\frac{\|\boldsymbol{\mu}_{s_i}^h - \boldsymbol{\mu}_{s_k}^h\|_2}{\|\boldsymbol{\mu}_{s_i}^h\|_2\|\boldsymbol{\mu}_{s_k}^h\|_2} + \alpha_\sigma\frac{\|\boldsymbol{\sigma}_{s_i}^h - \boldsymbol{\sigma}_{s_k}^h\|_2}{\|\boldsymbol{\sigma}_{s_i}^h\|_2\|\boldsymbol{\sigma}_{s_k}^h\|_2}, \tag{4}$$

where $\alpha_\mu, \alpha_\sigma$ are fixed parameters in all experiments.

4. Given similarity scores for all matrix pairs, we can complete a speaker verification task based on these scores.
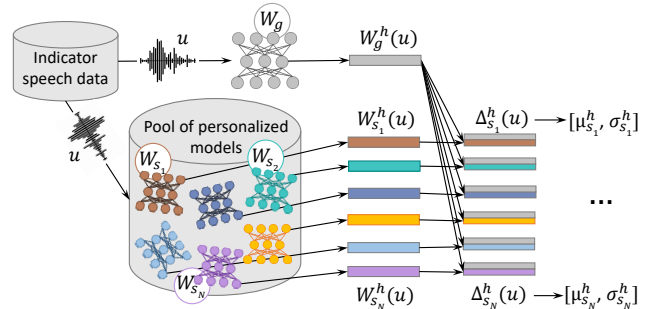


**Fig. 2**. Statistic computation for the attack model **A1**.

## 3.2.2. Attack model A2

For the second attack model, we train a NN model as shown in Figure 3. This NN model uses personalized and global models and the speech *Indicator* dataset for training. It is trained to predict a speaker identity provided the corresponding personalized model. When the model is trained, we use it in evaluation time to extract speaker embeddings similarly to x-vectors and apply probabilistic linear discriminant analysis (PLDA) [27, 28].

As shown in Figure 3, the model consists of two parts (frozen and trained). The outputs of the frozen part are $\mathbf{\Delta}_{s_i}^h$ sequences of vectors computed per utterance of the *Indicator* data as defined in Formula (1). For every personalized model $W_{s_i}$, we compute $\mathbf{\Delta}_{s_i}^h$ for all the utterances $\boldsymbol{u}$ of the *Indicator* corpus; then $\mathbf{\Delta}_{s_i}^h(\boldsymbol{u})$ is used as input to the second (trained) part of the NN which comprises several time delay neural network (TDNN) layers [29] and one statistical pooling layer.
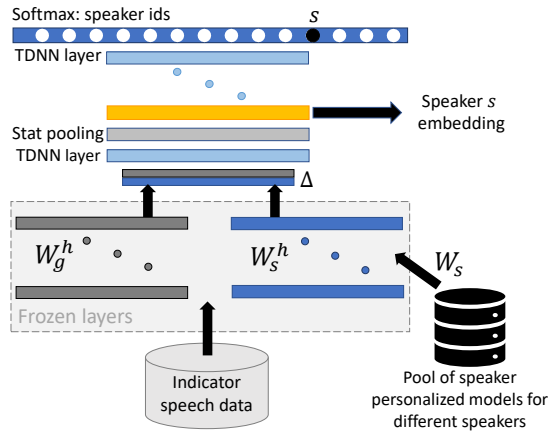


**Fig. 3**. Training a speaker embedding extractor for the attack model **A2**.

# 4. EXPERIMENTS

## 4.1. Data

The experiments were conducted on the speaker adaptation partition of the TED-LIUM 3 corpus [30]. This publicly available data set contains TED talks that amount to 452 hours speech data in English from about 2K speakers, 16kHz. Similarly to [3], we selected from the TED-LIUM 3 training dataset three datasets: *Train-G, Part-1, Part-2* with disjoint speaker subsets as shown in Table 1. The *Indicator* dataset was used to train an attack model. It is comprised of 320 utterances selected from all 32 speakers of test and development datasets of the TED-LIUM 3 corpus. The speakers in the *Indicator* dataset are disjoint from speakers in *Train-G, Part-1*, and *Part-2*. For each speaker in the *Indicator* dataset we select 10 utterances. The size of the *Indicator* dataset is 32 minutes. The *Train-G* dataset was used to train an initial global AM $W_g$. *Part-1* and *Part-2* were used to obtain two sets of personalized models.[1]

## 4.2. ASR acoustic models

The ASR AMs have a TDNN model architecture [29] and were trained using the Kaldi speech recognition toolkit [31]. 40-dimensional Mel-frequency cepstral coefficients (MFCCs) without cepstral

---

|  | Train-G | Part-1 | Part-2 | Indicator |
|---|---|---|---|---|
| Duration, hours | 200 | 86 | 73 | 0.5 |
| Number of speakers | 880 | 736 | 634 | 32 |
| Number of personalized models | — | 1300 | 1079 | — |

**Table 1**. Data sets statistics

truncation appended with 100-dimensional i-vectors were used as the input into the NNs. Each model has thirteen 512-dimensional hidden layers followed by a softmax layer where 3664 triphone states were used as targets[2]. The initial global model $W_g$ was trained using the lattice-free maximum mutual information (LF-MMI) criterion with a 3-fold reduced frame rate as in [32]. The two types of speech data augmentation strategies were applied for the training and adaptation data: speed perturbation (with factors 0.9, 1.0, 1.1) and volume perturbation, as in [29]. Each model has about 13.8M parameters. The initial global model $W_g$ was trained on the *Train-G*. Personalized models $W_{s_i}$ were obtained by fine-tuning all the parameters of $W_g$ on the speakers' data from *Part-1* and *Part-2* as described in [3]. For all personalized speaker models, we use approximately the same amount of speech data to perform fine-tuning (speaker adaptation) – about 4 minutes per model. For most of the speakers (564 in *Part-1*, 463 in *Part-2*) we obtained two different personalized models (per speaker) on disjoint adaptation subsets, for the rest speakers we have adaptation data only for one model.

## 4.3. Attack models

We investigate two approaches for attack models: **A1** – a simple approach based on the comparative statistical analysis of the NN outputs and the associated similarity score between personalized models, and **A2** – a NN based approach. For the test target trials, we use comparisons between different personalized models of the same speakers (564 in *Part-2* and 1027 in the *Part-1+Part-2*), and for the non-target trials we randomly selected 10K pairs of models from different speakers in a corresponding dataset.

### 4.3.1. Attack model A1

The first attack model was applied as described in Section 3.2.1. The parameters $\alpha_\mu$, $\alpha_\sigma$ in Formula (4) equal to 1 and 10 respectively. This model was evaluated on two datasets of personalized models corresponding to *Part-2* and combined *Part-1+Part-2* datasets. The *Indicator* dataset is the same in all experiments.

### 4.3.2. Attack model A2

For training the attack model **A2**, we use 1300 personalized speaker models corresponding to 736 unique speakers from *Part-1*. When we applied the frozen part of the architecture shown in Figure 3 to the 32-minute *Indicator* dataset for each speaker model in *Part-1*, we obtained the training data with the amount corresponding to about 693h (32×1300). The trained part of the NN model, illustrated in Figure 3, has a similar topology to a conventional x-vector extractor [27]. However, unlike the standard NN x-vector extractor, that is trained to predict speaker id-s by the input speech segment, our proposed model learns to predict a speaker identity from the $W_s^h$ part of a speaker personalized model. We trained 2 attack models corresponding to the two values of parameter $h \in \{1, 5\}$ – a hidden

---

layer in the ASR neural AMs at which we compute the activations. Values $h$ were chosen based on the results for the attack model **A1**. The output dimension of the frozen part is 512. The frozen part is followed by the trained part that consists of seven hidden TDNN layers and one statistical pooling layer introduced after the fifth TDNN layer. The output is a softmax layer with the targets corresponding to speakers in the pool of speaker personalized models (number of unique speakers in *Part-1*).

## 4.4. Results

The attack models were evaluated in terms of equal error rate (EER). Denoting by $P_{\text{fa}}(\theta)$ and $P_{\text{miss}}(\theta)$ the false alarm and miss rates at threshold $\theta$, the EER corresponds to the threshold $\theta_{\text{EER}}$ at which the two detection error rates are equal, i.e., $\text{EER} = P_{\text{fa}}(\theta_{\text{EER}}) = P_{\text{miss}}(\theta_{\text{EER}})$.

Results for the attack model **A1** are shown in Figure 4 for *Part-2* and combined *Part-1* and *Part-2* datasets. Speaker information can be captured for all values $h$ with varying success: EER ranges from 0.86% (for the first hidden layer) up to 20.51% (for the top hidden layer) on *Part-2*. For the *Part-1+Part-2* we observe similar results.
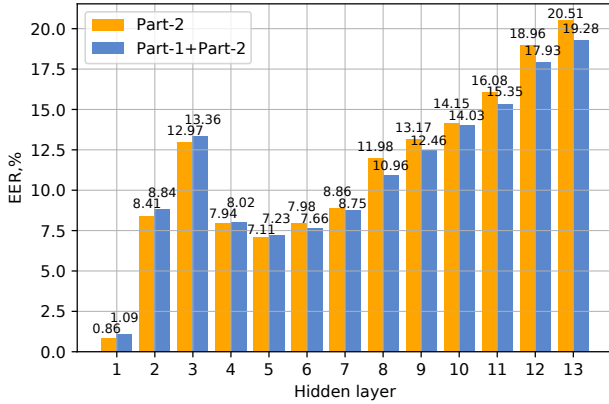


**Fig. 4**. EER, % for the attack model **A1** depending on the hidden layer $h$ (in $W_g$ and $W_{s_i}$) which was used to compute outputs, evaluated on *Part-2* and on the combined *Part-1+Part-2* dataset.
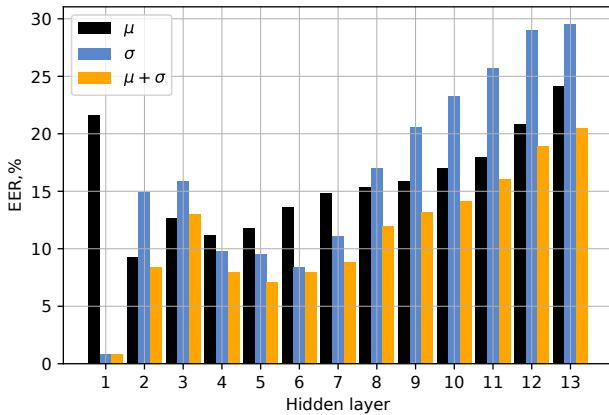


**Fig. 5**. EER, % for the attack model **A1** depending on the hidden layer $h$, evaluated on *Part-2* dataset. $\mu + \sigma$ – both means and standard deviations were used to compute similarity score $\rho$; $\mu$ – only means; and $\sigma$ – only standard deviations were used.

To analyze the impact of each component in Formula (4) on the ASV performance, we separately compute similarity score $\rho$ either using only means ($\alpha_\sigma = 0$) or only standard deviations ($\alpha_\mu = 0$). Results on the *Part-2* dataset are shown in Figure 5. Black bars correspond to $\alpha_\sigma = 0$ when only means were used to compute similarity scores $\rho$ between personalized models. Blue bars represent results for $\alpha_\mu = 0$ when only standard deviations were used to compute $\rho$. Orange bars correspond to the combined usage of means and standard deviations as in Figure 4 ($\alpha_\mu = 1$, $\alpha_\sigma = 10$). The impact of each component in the sum changes for different hidden layers. When we use only standard deviations, we observe the lowest EER on the first layer. In case of using only means, the first layer is, on the contrary, one of the least informative for speaker verification. For all other layers, combination of means and standard deviations provided superior results over the cases when only one of these components were used. These surprising results for the first hidden layer could possibly be explained by the fact that the personalized models incorporate i-vectors in their inputs, thus the speaker information can be easily learnt at this level of the NN. We plan to investigate this phenomena in detail in our future research.

We choose two values $h \in \{1, 5\}$ which demonstrate promising results for the model **A1**, and use the corresponding outputs to train two attack models with the configuration **A2**. The comparative results for the two attack models are presented in Table 2. For $h = 5$, the second attack model provides significant improvement in performance over the first one and reduces EER from 7% down to 2%. For $h = 1$, we could not obtain any improvement by training a NN based attack model: the results for **A2** in this case are worse than for the simple approach **A1**. One explanation for this phenomenon could be the following. The first layers of the AMs provide highly informative features for speaker classification, however, training the proposed NN model on these features results in overfitting because training criterion of the NN is speaker accuracy, but not the target EER metric, and the number of targets is relatively small, hence, the NN overfits to classify the seen speakers in the training dataset.

| Attack model | h=1 | h=5 |
|---|---|---|
| **A1** | **0.86** | 7.11 |
| **A2** | 12.31 | **1.94** |

**Table 2**. EER, % evaluated on *Part-2*, $h$ - indicator of a hidden layer

## 5. CONCLUSIONS

In this work, we focused on the privacy protection problem for ASR AMs trained in a FL framework. We explored to what extent ASR AMs are vulnerable to privacy attacks. We developed two attack models that aim to infer speaker identity from the locally updated personalized models without access to any speech data of the target speakers. One attack model is based on the proposed similarity score between personalized AMs computed on some external *Indicator* dataset, and another one is a NN model. We demonstrated on the TED-LIUM 3 corpus that both attack models are very effective and can provide EER of about 1% for the simple attack model **A1** and 2% for the NN attack model **A2**. Another important contribution of this work is the finding that the first layer of personalized AMs contains a large amount of speaker information that is mainly contained in the standard deviation values computed on *Indicator* data. This interesting property of NN adapted AMs opens new perspectives also for ASV, and in future work, we plan to use it for developing an efficient ASV system.

# 6. REFERENCES

[1] Xiaodong Cui, Songtao Lu, and Brian Kingsbury, "Federated acoustic modeling for automatic speech recognition," in *ICASSP*, 2021, pp. 6748–6752.

[2] Dimitrios Dimitriadis, Kenichi Kumatani, Robert Gmyr, et al., "A federated approach in training acoustic models.," in *Interspeech*, 2020, pp. 981–985.

[3] Salima Mdhaffar, Marc Tommasi, and Yannick Estève, "Study on acoustic model personalization in a context of collaborative learning constrained by privacy preservation," in *Speech and Computer*, 2021, pp. 426–436.

[4] Dhruv Guliani, Françoise Beaufays, and Giovanni Motta, "Training speech recognition models with federated learning: A quality/cost framework," in *ICASSP*, 2021, pp. 3080–3084.

[5] Dimitrios Dimitriadis, Kenichi Kumatani, Robert Gmyr, et al., "Federated transfer learning with dynamic gradient aggregation," *arXiv preprint arXiv:2008.02452*, 2020.

[6] Wentao Yu, Jan Freiwald, Sören Tewes, Fabien Huennemeyer, and Dorothea Kolossa, "Federated learning in ASR: Not as easy as you think," *arXiv preprint arXiv:2109.15108*, 2021.

[7] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[8] Viraaji Mothukuri, Reza M Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021.

[9] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller, "Inverting gradients–how easy is it to break privacy in federated learning?," *arXiv preprint arXiv:2003.14053*, 2020.

[10] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," in *28th Security Symposium*, 2019, pp. 267–284.

[11] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei, "Demystifying membership inference attacks in machine learning as a service," *IEEE Transactions on Services Computing*, 2019.

[12] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *IEEE INFOCOM*. IEEE, 2019, pp. 2512–2520.

[13] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, et al., "Practical secure aggregation for federated learning on user-held data," *arXiv preprint arXiv:1611.04482*, 2016.

[14] Cynthia Dwork, "Differential privacy," in *International Colloquium on Automata, Languages, and Programming*, 2006.

[15] Manas A Pathak, Bhiksha Raj, Shantanu D Rane, and Paris Smaragdis, "Privacy-preserving speech processing: cryptographic and string-matching frameworks show promise," *IEEE signal processing magazine*, vol. 30, no. 2, pp. 62–74, 2013.

[16] Paris Smaragdis and Madhusudana Shashanka, "A framework for secure speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1404–1413, 2007.

[17] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou, "Differentially private generative adversarial network," *arXiv preprint arXiv:1802.06739*, 2018.

[18] Alice Cohen-Hadria, Mark Cartwright, Brian McFee, and Juan Pablo Bello, "Voice anonymization in urban sound recordings," in *IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2019, pp. 1–6.

[19] Natalia Tomashenko, Brij Mohan Lal Srivastava, Xin Wang, Emmanuel Vincent, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, et al., "Introducing the VoicePrivacy initiative," in *Interspeech*, 2020, pp. 1693–1697.

[20] Natalia Tomashenko, Xin Wang, Emmanuel Vincent, Jose Patino, et al., "The VoicePrivacy 2020 Challenge: Results and findings," *Computer Speech and Language*, 2022.

[21] David Leroy, Alice Coucke, Thibaut Lavril, Thibault Gisselbrecht, and Joseph Dureau, "Federated learning for keyword spotting," in *ICASSP*. IEEE, 2019, pp. 6341–6345.

[22] Andrew Hard, Kurt Partridge, Cameron Nguyen, Niranjan Subrahmanya, Aishanee Shah, Pai Zhu, et al., "Training keyword spotting models on non-iid data with federated learning," *arXiv preprint arXiv:2005.10406*, 2020.

[23] Siddique Latif, Sara Khalifa, Rajib Rana, and Raja Jurdak, "Federated learning for speech emotion recognition applications," in *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2020, pp. 341–342.

[24] Filip Granqvist, Matt Seigel, Rogier van Dalen, Áine Cahill, Stephen Shum, and Matthias Paulik, "Improving on-device speaker verification using federated learning with privacy," *arXiv preprint arXiv:2008.02651*, 2020.

[25] Salima Mdhaffar, Jean-Francois Bonastre, Tommasi Marc, Tomashenko Natalia, et al., "Retrieving speaker information from personalized acoustic models for speech recognition," in *ICASSP*, 2022.

[26] Brendan McMahan, Eider Moore, Daniel Ramage, et al., "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[27] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *ICASSP*. IEEE, 2018, pp. 5329–5333.

[28] Sergey Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision*, 2006, pp. 531–542.

[29] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth annual conference of the international speech communication association*, 2015.

[30] François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Estève, "TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation," in *Speech and Computer*, 2018, pp. 198–208.

[31] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, et al., "The Kaldi speech recognition toolkit," in *ASRU*. IEEE Signal Processing Society, 2011.

[32] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, et al., "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *Interspeech*, 2016, pp. 2751–2755.