Federated Learning for Privacy-Preserving Speaker Recognition

ABRAHAM WOUBIE , AND TOM BÄCKSTRÖM
Department of Signal Processing and Acoustics, Aalto University, Espoo, Finland
Corresponding author: Abraham Woubie (abraham.zewoudie@aalto.fi)

This work has been supported by the Jane and Aatos Erkko foundation funding under contract 700795 AUTHSPKR.

ABSTRACT The state-of-the-art speaker recognition systems are usually trained on a single computer using speech data collected from multiple users. However, these speech samples may contain private information which users may not be willing to share. To overcome potential breaches of privacy, we investigate the use of federated learning with and without secure aggregators both for supervised and unsupervised speaker recognition systems. Federated learning enables training of a shared model without sharing private data by training the models on edge devices where the data resides. In the proposed system, each edge device trains an individual model which is subsequently sent to a secure aggregator or directly to the main server. To provide contrasting data without the need for transmitting data, we use a generative adversarial network to generate imposter data at the edge. Afterwards, the secure aggregator or the main server merges the individual models, builds a global model and transmits the global model to the edge devices. Experimental results on Voxceleb-1 dataset show that the use of federated learning both for supervised and unsupervised speaker recognition systems provides two advantages. Firstly, it retains privacy since the raw data does not leave the edge devices. Secondly, experimental results show that the aggregated model provides a better average equal error rate than the individual models when the federated model does not use a secure aggregator. Thus, our results quantify the challenges in practical application of privacy-preserving training of speaker training, in particular in terms of the trade-off between privacy and accuracy.

INDEX TERMS edge computation, federated learning, privacy, secure aggregator, speaker recognition

I. INTRODUCTION Speaker recognition is the process of automatically recognizing who is speaking on the basis of individual identifying information included in speech signals. It can further be subdivided into speaker identification and speaker verification; Speaker identification determines which registered speaker provides a given utterance from amongst a set of known speakers [1]. Speaker verification accepts or rejects the identity claim of a speaker [2]. Thus, speaker recognition is a vital feature when providing access to private services, as actions should be triggered only when a user with sufficient access rights has been identified.

Machine learning methods have recently been successfully used in a wide range of applications including speaker recognition [3]. Machine learning techniques usually train a deep network model by using a large number of labeled training data samples. The data samples are often collected on end devices such as smartphones and the model is trained using a computationally powerful centralized server [4]. The users send their data to the server which in turn uses the huge amount of data it collected from different uses to train a generic deep neural network model. However, the users’ data may contain private information which the users prefer not to share [5].

The transmission of large amounts of training data from each user to the server may also bring a substantial load on the communication link. This creates the need to train the model on the individual devices, i.e., train a centralized model in a distributed fashion [6]. The federated learning technique proposed in [7] can be used to update such decentralized models. Federated learning is a distributed machine learning method where a model can be trained on a large corpus of decentralized data. Thus, instead of requiring the users to share their private data, each user trains the network locally, and sends only updates to its locally trained model to the server. Afterwards, the server aggregates these updates into a global model [8, 9], commonly using a weighted average, known as federated averaging [7].

Mobile phones and smart devices are examples of the modern distributed networks that generate huge amounts of data
each day [10]. Since the computation power of these devices is growing rapidly together with the concerns of transmitting private information, federated learning has attracted attention for storing the data locally and pushing the network to the edge [10]. Federated learning methods have recently been used by different companies [8], [11], and play a crucial role by supporting different types of privacy-sensitive applications where the training data is distributed at different edge devices [12]–[15]. The growing demand of federated learning for different applications has resulted in a wide range of tools such as TensorFlow Federated [16], Federated AI Technology Enabler [17], Leaf [18], PaddleFL [19]. Though privacy-preserving data studies attracted attention beginning from the 1970s, it is only recently that it is being widely used at large scale [20]. For example, Google has recently started using federated learning in the Gboard mobile keyboard [13] and Android messages [21]. Apple is also using federated learning in iOS 13 [22] for applications such as “Hey Siri” [23].

Privacy concerns are considered as one of the major challenges in speaker recognition applications [24] as it involves the complete sharing of speech data, which can bring threatening consequences to people’s privacy. Federated learning can avoid privacy infringement by involving multiple participants to collaboratively learn a shared model without revealing their local data. For example, many commercial and public organizations want to take utmost care to uphold user privacy. This makes federated learning an important approach to consider when dealing with data that is private since it protects users’ privacy.

The main contribution of this work is the use of federated learning techniques in the training of both supervised and unsupervised deep neural network based speaker recognition classifiers with the objective of preserving user privacy. In the proposed system, each device trains its own model locally, and sends its local model to a secure aggregator or directly to a server. Then, the secure aggregator accumulates the local models from the different devices, builds a global model and sends it to the main server, or the main server builds directly the global model without the interaction of the secure aggregator. Finally, the main server sends the global model back to all devices.

The proposed system enables training of a speaker recognition model based on a deep neural network, using data stored on the devices which will never leave the corresponding device. The models are combined in the cloud with federated averaging, constructing a global model which is pushed back to devices for inference. The implementation of secure aggregation ensures that on a global level individual updates from devices are uninspectable. Since edge devices transmit only model-updates, raw data never leaves the edge. Thus, the aggregator has access only to a model trained to identify a local speaker and all other information about speech signals remains at the edge.

A second novelty is the use of a generative adversarial network (GAN) to generate imposter data at edge devices. By using a GAN, we avoid the need to transmit imposter data to the edge or collect such data at the edge. Transmission of imposter data could be a significant burden on bandwidth, but, importantly, could also provide access to differential information about the local speaker. Collection of imposter data at the edge could, on the other hand, be impractical.

- **Learning over smartphones:** By jointly learning the speech characteristics of a large number of mobile or similar devices, a joint statistical model can be trained to identify speakers. However, users may not be willing to physically transfer their data to a central server in order to protect their personal privacy. Thus, federated learning can be used to train a central speaker independent model without leaking private data.
- **Learning across organizations:** Organizations such as Universities can also be viewed as remote devices that contain a multitude of student data. However, Universities are usually required to keep their data and operate under strict privacy practices, and may face legal, administrative, or ethical constraints if the data is leaked. Federated learning can be used to enable private learning between various devices/organizations.

Our experiments with the Voxceleb-1 dataset show that both the supervised and unsupervised speaker recognition systems can benefit from federated learning, by not exporting sensitive user data to central servers, while achieving promising results compared to individual local models. In conclusion, the experimental results quantify the challenges in practical application of privacy-preserving training of speaker training, in particular in terms of the trade-off between privacy and accuracy.

### II. PROPOSED SYSTEM

The federated learning problem involves learning a single global statistical model from data stored from a few to potentially millions of remote devices. In particular, the goal is typically to minimize the following objective function:

\[
\min_w F(w), \quad \text{where } F(w) := \sum_{k=1}^{m} p_k F_k(w), \tag{1}
\]

where \(m\) is the total number of devices, \(F_k\) is the local objective function for the \(k\)th device, and \(p_k\) specifies the relative impact of each device with \(p_k \geq 0\) and \(\sum_{k=1}^{m} p_k = 1\).

Federated learning enables distributed training of speaker recognition models with heterogeneous clients. As shown in Fig. 1, the proposed federated learning system for speaker recognition operates at three main locations: edge devices, secure aggregator and main server. The edge devices could be mobile phones, laptops, or similar devices. The aggregator and main server are typically cloud services.

Fig. 1 further shows how a central model is trained over a distributed dataset, where a large number of nodes (e.g. user devices) hold variable-sized subsets of the data. A model update is computed at the device level and communicated to a central server. A large number of these updates or gradients are combined at the central server during each iteration of
analyze the impact of the secure aggregator. In the system
where the secure aggregator is used, firstly, an individual
model is trained locally on each individual device. Secondly,
each individual device sends its locally trained models to
the secure aggregator. Then, the secure aggregator builds
a global model and sends this aggregated model to the main
server. Finally, the main server sends the aggregated model
to each individual device. In the system where the secure
aggregator is not used, each device trains an individual model
locally and sends the model directly to the main server.
Afterwards, the main server builds a global model and sends
back the global model to each device.

Privacy is a major motivation for federated learning
applications. Such systems protect user data by sharing
model updates (e.g., gradient information) instead of the raw
data [27]–[29]. However, sending the model updates through-
out the training process could potentially reveal sensitive
private information, either to a third-party, or to the central
server [30]. Although the recent methods try to enhance
the privacy of federated learning by using tools such as
secure multiparty computation or differential privacy, these
approaches often provide privacy at the cost of reduced
model performance. Thus, the secure aggregator is a class
of secure multi-party computation algorithms wherein a group
of mutually distrustful devices \( d \in U \) each hold a private
value \( x_d \) and collaborate to compute an aggregate value,
such as the sum \( \sum_{u \in U} x_u \), without revealing to one another
any information about their private value except what is learnable
from the aggregate value itself.

The proposed system tries to preserve the privacy of feder-
ated learning by using secure multiparty computation [31].
[32]. The use of secure multiparty computation protects
individual model updates [31]. The central server can not see
any local updates. It could only view the exact aggregated
results at each round.

We use the classical federated learning average (Fed-
Avg) [7] that consists of two levels of optimization as shown
in Algorithm 1. These are local optimization performed on
participating clients, and a server step to update the global
model. Algorithm 4 further shows that the devices only
communicate updated weights instead of speech data, which
remain secure locally. Thus, this technique keeps the privacy
of the user’s speech data.

One of the main challenges of federated learning is the
transfer of a large number of updated model parameters from
the users to the server, whose throughput is typically con-
strained [9]. [10]. [33]. [34]. This challenge can be tackled
by reducing the number of participating users, for example,
by scheduling policies [35]. [36].

**III. EXPERIMENTS**

**A. EXPERIMENTAL SETUP**

The input features of the system are mel-spectrograms com-
puted within a 30 ms frame window at 10 ms shift using
Librosa [37]. Mean and variance normalization is performed
on every frequency bin of the spectrum. The mel-spectrogram
Algorithm 1: FedAvg. The C devices are indexed by c, epochs are indexed by e, and n is the number of speech samples.

0: Initialize \( w^0 \)
0: for each epoch \( e = 1, 2, \ldots \) do
0: \( D \leftarrow (\text{random subset of } M \text{ clients}) \)
0: for each client \( c \in C \) do
0: \( \hat{w}_c^e \leftarrow \text{ClientUpdate}(c, w^e) \)
0: \( \Delta w_c^e = w^e - \hat{w}_c^e \)
end for
0: \( \bar{w}^e = \sum_{c=1}^{C} \frac{n_c}{n} \Delta w_c^e \) \( \triangleright \) weighted average
end for
0: \( w^{e+1} = w^e - \eta \bar{w}^e \) \( \triangleright \) Serverupdate = 0

TABLE 1: The architecture used for the supervised speaker verification system.

| Layer | Kernel | Filters | Output size |
|-------|--------|---------|-------------|
| Conv-1 | 3x3 | 64 | 350 X 80 X 64 |
| Conv-2 | 3x3 | 128 | 175 X 40 X 128 |
| Conv-3 | 3x3 | 256 | 87 X 20 X 256 |
| fc-1 | - | 1000 | - |
| fc-2 | - | 400 | - |
| fc-3 | - | 1 | - |

features are extracted from the first 3.5 seconds of Voxceleb-1 audio files. Thus, the size of the mel-spectrogram is 350 × 80.

The system architectures are:

- **Supervised system**: Since the size of mel-spectrograms is 350 by 80, we use “Conv 2D” to train the model. The CNN architecture used in this work is similar to VGG-M [38], widely used for image classification and speech-related applications [39]. The details are shown in Table I. We also apply a max pooling layer of size 2 by 2, batch normalization and dropout layers. The supervised system has been implemented using the Keras deep learning library [40] to train the model. The network is trained on Titan X GPUs for 100 epochs or until the validation error stops decreasing, whichever is sooner, using a batch-size of 64. We use SGD with momentum (0.9), weight decay (5 × 10^-4) and a logarithmically decaying learning rate (initialised to 10^-2 and decaying to 10^-6).

- **Unsupervised system**: We use an autoencoder to train the unsupervised system. The autoencoder model for each speaker has been trained by using VoxCeleb-1 (i.e., the speaker 1 model is trained using the speaker 1 data from Voxceleb). The main reason for using an autoencoder architecture is to teach the network to learn a representation of speaker specific speech data. The CNN component of the network, which we recognised as the encoder, was optimised by learning a complex representation of the given mel-spectrograms, while the decoder component was optimised to decode the encoder’s output back to the corresponding mel-spectrogram. After the training process, we discard the decoder component and reuse the already learnt encoding representation for the speaker verification task. We do not use the impostor data in the unsupervised system since the main task is to learn the mel-spectrogram representation of different speakers.

The proposed speaker verification system has been carried out on the VoxCeleb-1 database [25]. It contains 148 642 development and 4874 test utterances, which belong to 1211 and 40 speakers, respectively. We selected 1000 speakers’ speech data from the development set to evaluate the performance of the proposed model. We used 90% of each speaker data to train an individual speaker dependent model and used 10% of the speaker data to do the evaluation. For example, if speaker 1 has 100 speech files in the development set of Voxceleb-1, we use 90 files to train the speaker model and 10 files to do the evaluation. In addition to this, we also add impostor data in the test set. We can not use the test set of Voxceleb-1 for this work as there is no overlap between the speakers in the development and test set.

Firstly, we wanted to train the individual speaker model using only the true client speech of a speaker on each device. However, since the number of files for each speaker are few in Voxceleb-1 dataset (most of the speakers have less than 100 speech files), training individual speaker models using only true client speech led to an over-fitting problem. Thus, we trained the individual speaker models using the true speech of the speaker and by using the speech of other speakers as impostor speech.

We use two different methods to generate the impostor speech for each individual device. In the first method, we take the speech of other speakers from the Voxceleb dataset as impostor speech. We select 100 samples of other persons as impostor speech for each speaker on a given device. In the second method, the impostor data is generated using the GAN model. We use the work of [26] to train the GAN model on the Voxceleb dataset. Similar to the first method, we generate 100 impostor speech data for each individual device.

The main issue of training the GAN model to generate the impostor speech is its training time. The computational cost of training the GAN model using 50 hours of Voxceleb dataset on Quadro P2000 GPU is 3.5 hours. But, once the GAN model is trained, the extraction of impostor speech samples on the edge devices is quite fast. The training of the GAN model is only a one-time task.

The performance of the proposed system is evaluated using equal error rate (EER) which is the rate at which both acceptance and rejection errors are equal.

B. EXPERIMENTAL RESULTS

Our objective is to quantify both the overall performance of federated learning in speaker recognition, but in particular, to determine also how the supervised and unsupervised systems perform in comparison to each other, both with and without the secure aggregator.
1) Supervised Systems without Secure Aggregator

Fig. 3 shows the performance of the individual and aggregated speaker models for the supervised system with and without using GAN, respectively. The main difference between the models in Fig. 3(a) and 3(b) is in the generation of impostor speech. In the case of Fig. 3(a), the impostor speech samples are generated by selecting other persons’ speech (i.e., we take the speeches of other persons from Voxceleb as impostor speech for a given speaker). But, in the case of Fig. 3(b), we use the GAN model to generate the impostor speeches.

The main difference between the individual and aggregated speaker models is that, in the case of the individual model, a speaker model is first trained for each speaker using his/her own speech data. Then, the speech samples of the speaker are evaluated using this trained speaker model. In this work, we use the individual model as a baseline system (i.e., 1000 individual speaker models are trained using the speech samples of speakers in 1000 devices). But, in the case of the aggregated model, each of the 1000 devices send their parameters to the secure aggregator and the aggregator takes the average of these parameters and sets them as its new weight parameters and passes them back to the 1000 devices. Thus, we call this aggregated (federated) speaker model.

As shown in Fig. Fig. 3, the EER of all 1000 devices when each device uses its own individual model is greater than 3.6. But, when we use the federated/aggregated models, irrespective of impostor generation method, most of the devices provide us EER values of less than 3.6. In the first method where the speech of other persons are used as impostor speech, about 350 devices provide us EER value of less than 3.6. In the second aggregation method where we use GAN to generate impostor speech, almost similar number of speakers/devices provide us EER value of less than 3.6.

Fig. 3(a) and Fig. 3(b) show that whether we use GAN or use other persons speech as impostor data, the aggregated speaker model provides better average EER than the individual model. The figures also reveal that the two aggregated methods provide more or less similar average EER. This shows that we can use GAN to generate impostor speech on edge devices rather than transferring impostor data from other sources to the edge devices.

Table 2 shows that the average EER of individual models of the 1000 devices/speakers of the supervised speaker verification system is 4.09. Note that the EER of individual speakers has been computed using the data trained for each speaker/device. For example, the EER of speaker1 is computed by using the speaker model trained using speaker1’s data. We consider the average EER of the individual models as the baseline system.

Table 2 also shows that the average EER of 1000 devices of the federated model that does not use both the secure aggregator and GAN is 3.5. This represents a 14.42% relative EER improvement compared to the baseline systems. Similarly, using GAN to generate impostor data in training the speaker models also provides better EER results compared to the baseline system. The table shows that the federated model that uses GAN data, rather than impostor data from Voxceleb, provides an average EER of 3.65. This is a 10.75% relative improvement compared to the baseline systems.

2) Supervised Systems using Secure Aggregator

Table 2 shows that the average EER of 1000 devices of the federated model of the supervised system that uses both a secure aggregator and impostor speech from Voxceleb dataset is 4.46. Similarly, the average EER of 1000 devices of the federated model of the same system but that uses GAN technique to generate impostor speech is 4.59. Thus, the results show that irrespective of the impostor generation method, the use of a secure aggregator provides worse results compared both to the individual systems and the federated systems that do not use a secure aggregator.

The main reason for the deterioration of the EER when the federated system uses a secure aggregator is that while recent
3.50 3.75 4.00 4.25 4.50 4.75 5.00 5.25 5.50
EER (%)
0
20
40
60
80
100
120
140
160
Number of speakers
Individual Model
Federated Model
(a) Impostors selected from Voxceleb dataset.

(b) Impostors generated using GAN.

FIGURE 3: Histograms of the equal error rate (EER) over 1000 devices using the individual vs. federated model of the supervised system that uses a secure aggregator (SA).

TABLE 2: Equal Error Rate (EER) comparison of the supervised speaker verification systems of the individual and federated systems with and without using the Secure Aggregator (SA). The use of GAN generated data as impostor speech is included in the table.

| System                        | With SA | Without SA |
|-------------------------------|---------|------------|
| Individual Model              | 4.09    |            |
| Federated Model without GAN   | 4.46    | 3.5        |
| Federated Model with GAN      | 4.59    | 3.65       |

methods aim to enhance the privacy of federated learning by using a secure aggregator, this approach often provides privacy at the cost of reduced model performance or system efficiency. Thus, we need to take into consideration the privacy aspect in addition to the EER values. Still, the results of the systems that use secure aggregators are acceptable.

Figure 4 depicts the EER distribution of the 1000 devices of the individual and aggregated models for the supervised system with and without a secure aggregator. The figure also displays the impact of using GAN as an impostor speech generation method. The figure shows the minimum, lower quartile, median, upper quartile, and maximum EER values, respectively. As shown in the figure, the use of aggregated models that does not use the secure aggregator, irrespective of impostor generation methods, provides better average EER than the individual models.

3) Unsupervised Systems

Figure 5 shows the Equal Error Rates (EERs) of the individual and aggregated speaker models of the unsupervised system with and without using a secure aggregator. As shown in Fig. 5, the EER of all the 1000 devices when each device uses its own individual model is greater than 4.3. But, when the federated model does not use a secure aggregator as shown in Fig. 5 (a), about 680 devices provide us an EER value of less than 4.3. This shows that 68% of the speaker models provide us better results with the aggregated speaker model.

Contrary to the results reported in Fig. 5 (a), the use of a secure aggregator worsens the results. As Fig. 5 (b) shows, the use of a secure aggregator deteriorates the EER results and provides worse results compared to the individual models.

Table 3 exhibits that the average EER of individual models of the 1000 devices/speakers of the unsupervised speaker verification system is 4.76. The table shows the average EER of 1000 devices of the federated model for the same systems that do not use a secure aggregator is 4.2. This represents a 11.76% relative EER improvement compared to the baseline unsupervised system. But, the table also shows that the use of the secure aggregator in the federated model provides worse result than the baseline system.

The results of Table 2 and 3 show that irrespective of the speaker verification system (i.e., whether it is supervised or unsupervised), the use of aggregated models provides better results than the individual models.
FIGURE 5: Equal error rate (EER) of 1000 devices using the individual vs. federated model for the unsupervised system.

TABLE 3: Equal Error Rate (EER) comparison of the unsupervised speaker verification systems of the individual and federated systems with and without using the Secure Aggregator (SA).

| System               | With SA | Without SA |
|----------------------|---------|------------|
| Individual Model     | 4.76    | 4.2        |
| Federated Model      | 5.19    | 4.2        |

FIGURE 6: Box plot of the equal error rates (EER) of the unsupervised individual and federated models of the 1000 devices with/without using Secure Aggregator (SA).

The results reported demonstrate that federated learning is a useful tool for protecting privacy in training of speaker recognition models. The experimental results show that even if the aggregated models do not always provide better EER results compared to the individual model for all the devices, the aggregated speaker models do provide a better average EER than the individual models.

The proposed work uses 1000 devices to compare the performance of the individual against the federated model. We use Student’s t-test statistical technique to check if the means of individual and federated models are significantly different from each other. Student’s t-test uses a null hypothesis and an alternate hypothesis. The null hypothesis is valid when all the sample means are equal, or they don’t have any significant difference. Thus, we compared the EERs of the three experiments to make sure that the EER differences are because of the federated model, not merely by chance.

We compute two Student’s t-test values. While the first one compares the individual model against the federated model that selects impostor speech from Voxceleb (i.e., federated model 1), the second one compares the individual model again the federated model that uses GAN to generate impostor speech (i.e., federated model 2). The Student’s t-test computation on the first and second comparison provides P values of 0.0001 and 0.00001, respectively. Thus, since the P values in both cases are less than the standard significance

unsupervised), the use of a secure aggregator reduces the performance of the federated mode. But, when the federated model does not use a secure aggregator, the EER results improve both for the supervised and unsupervised systems compared to the individual models. Although the inclusion of the secure aggregator deteriorates the EER results a bit, we still need to take into account that using the secure aggregator keeps the privacy of the data. But, still, the EER results are good and we could accept the deterioration of the EER results.

Note that in addition to the individual and federated models, we carried out another experiment by pooling all the speech samples from the 1000 speakers and trained one
level of 0.05, we reject the null hypothesis. Thus, the experimental results of Fig. 4 and 6 show that the difference of the mean EER values of the individual and federated models are statistically significant.

Finally, in the proposed work, each device sends its model updates, (i.e., the learned parameters) to the secure aggregator only once (i.e., each device sends the local model to the secure aggregator after it finishes the training). Thus, the federated model results reported in use only one model update. We also assess the impact of updating the local model updates more than once. Thus, we run another experiment where each device sends its model updates two times to the secure aggregator. Our experimental analysis shows that updating the local model update more than once does not improve the EER. One possible reason could be since the training data used in each device is almost similar for each training phase, the EER results of updating the local models either once or twice are similar. The real performance could be revealed if the data was partitioned among devices, but we can not use this method since the privacy of the data would then be compromised.

IV. CONCLUSIONS
In this work, we propose the use of federate learning to keep the privacy of speech on edge devices both for supervised and unsupervised speaker recognition systems. The proposed technique is a decentralized training method that does not require devices to send their raw data to central servers. Instead, the users’ data is stored and processed only on the corresponding edge device. Thus, the training is carried out only locally, and each device contributes updates to a central model. Then, the secure aggregator creates one federated model from the local models and distributes it through the main server back to the devices. In addition to this, the work analyzed the impact of using the secure aggregator on performance of speaker recognition systems.

The proposed system provides two main advantages. Firstly, since raw data does not leave the individual devices, the privacy of the speaker is retained. Secondly, the experimental results also show that the federated model that does not use the secure aggregator provides a better average Equal Error Rate (EER) than the individual models. But, when the federated model uses the secure aggregator, the aggregated model provides worse EER results than individual models. Still, the EER results are good and we need to take into account the trade-offs between privacy and performance.

In future work, we should study better aggregation methods rather than using a simple averaging technique. In addition, we should also explore the effect of increasing the number of devices beyond the 1000 devices used in this work.

REFERENCES
[1] D. A. Reynolds, “Automatic speaker recognition using gaussian mixture speaker models,” in The Lincoln Laboratory Journal. Citeseer, 1995.
[2] S. Furui, “An overview of speaker recognition technology;” in Automatic speech and speaker recognition. Springer, 1996, pp. 31–56.
[3] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” nature, vol. 521, no. 7553, pp. 436–444, 2015.
[4] J. Chen and X. Ran, “Deep learning with edge computing: A review,” Proceedings of the IEEE, vol. 107, no. 8, pp. 1655–1674, 2019.
[5] A. Nautsch, A. Jiménez, A. Treiber, J. Kolberg, C. Jasserand, E. Kindt, H. Delgado, M. Todisco, M. A. Hmani, A. Mtibaa et al., “Preserving privacy in speaker and speech characterisation,” Computer Speech & Language, vol. 58, pp. 441–480, 2019.
[6] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker et al., “Large scale distributed deep networks,” 2012.
[7] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in Artificial Intelligence and Statistics. PMLR, vol. 123, pp. 1273–1282.
[8] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, H. B. McMahan et al., “Towards federated learning at scale: System design,” arXiv preprint arXiv:1902.01046, 2019.
[9] P. Kairouz, H. B. McMahan, B. Avent, B. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings et al., “Advances and open problems in federated learning,” arXiv preprint arXiv:1912.04977, 2019.
[10] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” IEEE Signal Processing Magazine, vol. 37, no. 3, pp. 50–60, 2020.
[11] W. Group et al., “Federated learning white paper v1. 0,” 2020.
[12] M. Huang, H. Li, B. Bai, C. Wang, K. Bai, and F. Wang, “A federated multi-view deep learning framework for privacy-preserving recommendations,” arXiv preprint arXiv:2008.10808, 2020.
[13] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, “Federated learning for mobile keyboard prediction,” arXiv preprint arXiv:1811.03604, 2018.
[14] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” ACM Transactions on Intelligent Systems and Technology (TIST), vol. 10, no. 2, pp. 1–19, 2019.
[15] Y. Zhao, J. Zhao, L. Jiang, R. Tan, and D. Niyato, “Mobile edge computing, blockchain and reputation-based crowdsourcing in federated learning: A secure, decentralized and privacy-preserving system,” 2020.
[16] The TFF Authors, “TensorFlow Federated,” https://www.tensorflow.org/federated/2019, accessed: 2021-09-20.
[17] The FATE Authors, “Federated AI Technology Enabler,” [URLhttps://www.fate.ai], 2019, accessed: 2021-09-20.
[18] The Leaf Authors, “Leaf,”[https://leaf.cmum.edu/], 2019, accessed: 2021-09-20.
[19] The PaddleFL Authors, “PaddleFL,” [https://github.com/PaddlePaddle/PaddleFL], 2019, accessed: 2021-09-20.
[20] U. Erlingsson, V. Pihur, and A. Korolova, “Rappor: Randomized aggregatable privacy-preserving ordinal response,” in Proceedings of the 2014 ACM SIGSAC conference on computer and communications security, 2014, pp. 1054–1067.
[21] support.google, “Gmail chats stay private while Messages improves suggestions,” [https://support.google.com/messages/answer/93529092], 2019, accessed: 2021-09-20.
[22] Apple, “Private Federated Learning (NearIPS 2019 Expo Talk Abstract),” [https://nips.cc/ExpoConferences/2019/schedule?talk_id=40], 2019, accessed: 2021-09-20.
[23] Apple, “Designing for privacy,” [https://developer.apple.com/videos/play/wwdc2019/0706/], 2019, accessed: 2021-09-20.
[24] Y. Rahulamathavan, K. R. Sutharsini, I. G. Ray, R. Lu, and M. Rajaratnam, “Privacy-preserving vector-based speaker verification,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 27, no. 3, pp. 496–506, 2018.
[25] A. Nagrani, J. S. Chung, and A. Zisserman, “Voxceleb: a large-scale speaker identification dataset,” arXiv preprint arXiv:1706.08612, 2017.
[26] C. Donahue, J. McAuley, and M. Puckette, “Adversarial audio synthesis,” arXiv preprint arXiv:1811.03604, 2018.
Abraham Woubie et al.: Federated Learning for Privacy-Preserving Speaker Recognition

[29] C. Dwork, A. Roth et al., “The algorithmic foundations of differential privacy,” *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, 2014.

[30] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, “Learning differentially private recurrent language models,” *arXiv preprint arXiv:1710.06963*, 2017.

[31] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, “Practical secure aggregation for privacy-preserving machine learning,” in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[32] B. Ghazi, R. Pagh, and A. Velingker, “Scalable and differentially private distributed aggregation in the shuffled model,” *arXiv preprint arXiv:1906.08320*, 2019.

[33] H. Brendan McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, “Communication-efficient learning of deep networks from decentralized data,” *ArXiv e-prints*, pp. arXiv–1602, 2016.

[34] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, “A joint learning and communications framework for federated learning over wireless networks,” *IEEE Transactions on Wireless Communications*, 2020.

[35] H. H. Yang, Z. Liu, T. Q. Quek, and H. V. Poor, “Scheduling policies for federated learning in wireless networks,” *IEEE transactions on communications*, vol. 68, no. 1, pp. 317–333, 2019.

[36] M. M. Amiri, D. Gündüz, S. R. Kulkarni, and H. V. Poor, “Update aware device scheduling for federated learning at the wireless edge,” in *2020 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2020, pp. 2598–2603.

[37] B. McFee, M. McVicar, C. Raffel, D. Liang, O. Nieto, J. Moore, D. Ellis, D. Repetto, P. Viktorin, J. F. Santos et al., “Librosa: v0. 4.0,” *Zenodo 2015*, 2015.

[38] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” *arXiv preprint arXiv:1405.3531*, 2014.

[39] J. S. Chung and A. Zisserman, “Out of time: automated lip sync in the wild,” in *Asian conference on computer vision*. Springer, 2016, pp. 251–263.

[40] F. Chollet et al., “Keras (2015),” 2017.

ABRAHAM WOUBIE completed his Bachelor and Master of Science Degrees in Computer Science with very great distinctions from Arba Minch and Addis Ababa Universities in Ethiopia, respectively. He completed his PhD in September 2017 in Speech Signal Processing from Technical University of Catalonia in Barcelona, Spain. He is now a Post-Doctoral Researcher at Aalto University. His research interests include deep reinforcement learning, speech signal processing and machine learning. He is the author of papers on these topics.

TOM BÄCKSTRÖM (Senior Member, IEEE) received the master’s and Ph.D. degrees from Aalto University, in 2001 and 2004, respectively, which was then known as the Helsinki University of Technology. He has been an Associate Professor with the Department of Signal Processing and Acoustics, Aalto University, Finland, since 2016. He was a Professor at the International Audio Laboratory Erlangen, Friedrich-Alexander University, from 2013 to 2016, and a Researcher at Fraunhofer IIS from 2008 to 2013. He has contributed to several international speech and audio coding standards and is the chair and co-founder of the ISCA Special Interest Group “Security and Privacy in Speech Communication.” His research interests include technologies for spoken interaction, emphasizing efficiency and privacy, and in particular in multi-device and multi-user environments.