A GDPR-compliant Ecosystem for Speech Recognition with Transfer, Federated, and Evolutionary Learning

DI JIANG, CONGHUI TAN, JINHUA PENG, and CHAOTAO CHEN, AI Group, WeBank Co., Ltd., China
XUEYANG WU, Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong
WEIWEI ZHAO and YUANFENG SONG, AI Group, WeBank Co., Ltd., China
YONGXIN TONG, BDBC, SKLSDE Lab and IRI, Beihang University, China
CHANG LIU and QIAN XU, AI Group, WeBank Co., Ltd., China
QIANG YANG, AI Group, WeBank Co., Ltd., China and Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong
LI DENG, Citadel LLC, USA

Automatic Speech Recognition (ASR) is playing a vital role in a wide range of real-world applications. However, Commercial ASR solutions are typically "one-size-fits-all" products and clients are inevitably faced with the risk of severe performance degradation in field test. Meanwhile, with new data regulations such as the European Union's General Data Protection Regulation (GDPR) coming into force, ASR vendors, which traditionally utilize the speech training data in a centralized approach, are becoming increasingly helpless to solve this problem, since accessing clients' speech data is prohibited. Here, we show that by seamlessly integrating three machine learning paradigms (i.e., **T**ransfer learning, **F**ederated learning, and **E**volutionary learning (TFE)), we can successfully build a win-win ecosystem for ASR clients and vendors and solve all the aforementioned problems plaguing them. Through large-scale quantitative experiments, we show that with TFE, the clients can enjoy far better ASR solutions than the "one-size-fits-all" counterpart, and the vendors can exploit the abundance of clients' data to effectively refine their own ASR products.

CCS Concepts: • **Computing methodologies → Speech recognition;**

Additional Key Words and Phrases: Speech recognition, federated learning, transfer learning, evolutionary learning

Xueyang Wu work done when he worked as an intern at AI Group, Webank Co., Ltd.
Authors' addresses: D. Jiang, C. Tan (corresponding author), J. Peng, C. Chen, AI Group, WeBank Co., Ltd., Shenzhen, China; emails: {dijiang, martintan, kinvapeng, chaotaochen}@webank.com; X. Wu, Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Kowloon, Hong Kong; email: xwuba@connect.ust.hk; W. Zhao and Y. Song, AI Group, WeBank Co., Ltd., Shenzhen, China; emails: {davezhao, yfsong}@webank.com; Y. Tong, BDBC, SKLSDE Lab and IRI, Beihang University, Beijing, China; email: yxtong@buaa.edu.cn; C. Liu and Q. Xu, AI Group, WeBank Co., Ltd., Shenzhen, China; emails: {changliu, qianxu}@webank.com; Q. Yang, AI Group, WeBank Co., Ltd., Shenzhen, China, Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong; email: qyang@cse.ust.hk; L. Deng, Citadel LLC, 131 South Dearborn Street, Chicago, IL, 60603, USA; email: deng629@gmail.com.

2157-6904/2021/04-ART30 $15.00
https://doi.org/10.1145/3447687

ACM Reference format:
Di Jiang, Conghui Tan, Jinhua Peng, Chaotao Chen, Xueyang Wu, Weiwei Zhao, Yuanfeng Song, Yongxin Tong, Chang Liu, Qian Xu, Qiang Yang, and Li Deng. 2021. A GDPR-compliant Ecosystem for Speech Recognition with Transfer, Federated, and Evolutionary Learning. *ACM Trans. Intell. Syst. Technol.* 12, 3, Article 30 (April 2021), 19 pages.
https://doi.org/10.1145/3447687

1 INTRODUCTION

With the popularity of smart devices such as voice-controlled speakers and televisions, the need for **Automatic Speech Recognition (ASR)** has become ubiquitous in contemporary life. Modern ASR systems can be roughly divided into two major components: the **acoustic model (AM)** component and the **language model (LM)** component [81, 83]. The AM is responsible for mapping acoustic signals to phones while LM is responsible for guiding the search of grammatically legitimate word sequences.

The vendors of commercial ASR systems typically provide "one-size-fits-all" products (i.e., fixed AM and LM) for all clients. For specific client scenarios, the "one-size-fits-all" ASR systems inevitably suffer from dramatic performance degradation due to the discrepancy between training and testing data [25]. As an imperfect patch up, the service of tuning LM for the client is increasingly supported by ASR vendors. However, this service heavily relies upon client uploading sensitive transcripts of their private speech data to the vendor. Considering that strict data regulations such as the European Union's **General Data Protection Regulation (GDPR)** [72] has come into effect, such privacy-violating practice becomes illegal for real-life applications. Meanwhile, since the amount and diversity of speech data utilized for training the ASR system is critical for the performance of AM, the speech data stored on clients' machines is invaluable resources for ASR vendors to further refine their ASR systems. However, with the arrival of the era of strict protection of data privacy, the ASR vendors are facing unprecedented challenges of obtaining the speech data that is generated from real-life scenarios. Hence, neither the clients nor the vendors would be prosperous if the status quo of the ASR ecosystem remained unchanged in the long run and a GDPR-compliant ecosystem needs to be established for the ASR industry.

In this article, we propose a novel framework that seamlessly integrates transfer learning, federated learning and evolutionary learning to meet the above requirements. In the proposed framework, transfer learning is responsible for tuning a highly customized ASR system for the client and overcoming the performance degrade caused by the "one-size-fit-all" LM and AM. Federated learning bridges the gap of information flow between the clients and the vendor in a privacy-preserving way. With differential privacy, the perturbed version of tuned AM works as a compact and secure proxy of clients' data and are communicated between the clients and the vendor. When the vendor receives the transmitted AMs from the clients, evolutionary learning is employed to integrate them and generate a next-generation general-purpose ASR system. To verify the effectiveness of the proposed framework, we conduct large-scale experiments on real-life datasets with regards to different quantitative metrics. The experimental results univocally verify its validity and technical superiority.

The major contributions of this article are summarized as follows:

- To the best of our knowledge, this is the first framework that equips ASR system with deep customization capability of both LM and AM for any client scenario.
- The proposed framework pioneers in communicating ASR components between the clients and the vendor while make the whole procedure privacy-preserving and GDPR-compliant.

- An evolutionary algorithm with new operators is proposed to integrate a wide range of tuned AMs to generate the next-generation ASR system for the vendor.

The rest of this article is organized as follows. We review related works in Section 2. Then, we discuss the architecture and the detailed techniques used for building the proposed TFE framework in Section 3. The experimental results are shown in Section 4. Finally, we conclude this article in Section 5.

2 RELATED WORK

This article is closely related to automatic speech recognition and machine learning paradigms such as transfer learning, federated learning, and evolutionary learning. We briefly review the most related work in the following subsections.

2.1 Automatic Speech Recognition

Speech recognition has been intensively studied for decades. The modern ASR system can be roughly divided into two major components: the LM and the AM. In an ASR system, the LM plays the role of guiding the candidate search and evaluating the quality of the decoding output.

For decades, traditional statistical LMs such as the backoff *n*-gram LM has dominated this area due to its simplicity and reliability. Neural LM is proposed in Reference [6], which uses a three layer neural network to predict the conditional probability distribution of the next words given previous words. Then, a hierarchical probabilistic neural LM is proposed in Reference [51] mainly focusing on speeding up the training time. The **recurrent neural network–based language model (RNNLM)** and its variant [47, 48] uses recurrent connections to preserve short term memory. **Bidirectional Encoder Representation from Transformers (BERT)** is proposed in Reference [19], which applies the bidirectional training of transformer, a popular attention model, to language model. It is proved that a language model that is bidirectionally trained has a deeper sense of language context and flow than single-direction language models.

As for AM, the deep learning–based AMs such as the DNN-HMM made a great breakthrough in ASR industry [18]. **Connectionist temporal classification (CTC)**, proposed in Reference [26], is a fully end-to-end acoustic model training that eliminates the need to pre-align the data, requiring only one input sequence and one output sequence to train. The speech recognition problem is essentially the problem of direct conversion of two variable length sequences. The Seq2Seq [70] model's elegant model structure and powerful performance make the speech recognition problem hopefully get rid of the language model completely and pronunciation dictionary. At present, carefully tuned DNN-HMM AM still maintains the state-of-the-art performance.

However, even with the advancement of AM and LM, existing commercial ASR systems still work suboptimally for the clients due to the discrepancy between training data and those from the client scenarios. To the best of our knowledge, there hardly exists any ASR products with deep customization capability for both LM and AM. Although some commercial ASR products provide functionality of tuning LM through client uploading transcripts or keywords, it is becoming illegitimate due to the serious breach of data privacy.

2.2 Transfer Learning

With the increasing scale of machine learning models, the in-domain annotated data play an significantly important role in real world applications, which are expensive at time and cost. How to alleviate the need of in-domain labeled data has drawn many research attentions, and researchers have

proposed plentiful algorithms that is summarized as transfer learning [55]. Transfer learning aims at leveraging the knowledge from one task (or domain) and help the learning of another task (or domain). According to what, when, and how to transfer knowledge, transfer learning algorithms are further categorized into four types: instance-based transfer learning, feature-based transfer learning, model-based transfer learning, and relation-based transfer learning. As deep learning techniques become dominant, more feature-based and model-based transfer learning techniques are proposed and widely applied, which significantly improve the performance on small data learning as well as reduce the training cost.

The transfer learning techniques for AM used are conventionally termed as acoustic model adaptation, aiming at adapting the acoustic model to the target domain to reduce the domain discrepancy. Most of works follows the idea to align two or more distributions, as has been summarized in Reference [68]. Previous works on acoustic model adaptation can be divided into three categories: (1) linear transformation, (2) conservative training, and (3) subspace method. From the aspect of transfer learning, methods (1) and (3) are related to feature-based transfer learning, and method (2) is related to model-based transfer learning. Linear transformation holds the assumption that the speech features can be normalized through linear mapping. Linear transformation simply add a transformation network (or transformation layer) into the existed network to perform linear mapping. It is a popular adaptation method for neural networks. The linear input networks [2, 52, 71] apply a transformation network before the input layer. Gemello et al. [23] examined the effectiveness of inserting transformation layer between different layers of the original network. Li et al. [39] proposed that the distortion of speech variation is an important factor of performance reduction. The target of subspace method is to find a subspace for each to construct adapted model parameters or transformations as a point in the subspace. Li et al. [41] used subspace to estimate transformation matrix by considering it as a random variable. They applied **principal components analysis (PCA)** to decompose the adaptation matrices into principal directions in speaker space. Conservative training has become the mainstreaming accented adaptation methods for it can utilize the trained model and needs small amount of accented data to performance adaptation. Literature [82] introduces **KL-Divergence (KLD)** regularization term to DNN. The mathematical form of KLD regularization is neat and differentiable, which is critical for deep learning training. Conservative training is effective and efficient, and it does not need too many data to achieve an acceptable result. However, conservative training involves too many parameters, which may still break the structure of model [78]. Recently, a more powerful transfer learning technique for distribution adaption has been proposed [73] whose main idea is to dynamically learn the relative importance of marginal and conditional distributions in transfer learning, while related method has not been adapted to speech recognition due to its relatively large computing costs.

In the area of text mining, it is quite common to transfer the knowledge in pretrained models to new tasks [20, 49, 58]. In particular, Chronopoulou et al. [17] propose to incorporate the LM objective to the task-specific optimization function to transfer the knowledge preserved by pretrained language models. BERT [20] pretrains LMs and shows improvements for various downstream tasks. When it comes to speech recognition area, the transfer learning techniques for LM used are termed as language model adaptation, aiming to bridge the gap between the source domain and the target domain. Cache-based methods were proposed in literatures [36, 37] and by saving the recent decoding results, the model has the ability of increasing the probability of recent words. Singh et al. [63] propose a discriminative language model in which a perceptron is trained to re-rank the N-Best using features extracted by the first-pass decoding. Topic model-based approaches such as LDA [8] and WVM [13] is also been used to capture the correlation between words and generate document-specific language models. Recently, **recurrent neural network**

language model (RNNLM) was proposed in Reference [75], and Reference [40] further proposed a DNN-based model to adapt the language model for ASR.

2.3 Federated Learning

In recent years, the concerns of data privacy and security have arisen, since the current data-driven machine learning algorithms usually consume a massive amount of data, which may involve very sensitive information such as individual privacy, businesses secrets, or even about public security. Unfortunately, many research [7, 14] have proved that adversaries can attack machine learning models for their interests. To avoid the potential damages, many countries enforce or plan to publish laws and regulations to protect the personal information privacy and security. In 2018, the European Union enforced a profoundly influential regulation, called the EU GDPR [72], which aims to protect the security and privacy of user, giving the right to user for ripping their personal information from companies. As it happens, China and the United States also establish relevant laws to regulate the collection and utilization of individual information.

However, the collaborations of data and model help boost the performance on many applications. For the case that in-domain high-quality labeled data are scarce, but we can find another relevant domain with rich labeled data, transfer learning from a source model is a popular choice [55]. In the case where data are fragment and distributed on many parties, collaborative machine learning can help jointly utilize these data and learn a more powerful model. However, neither of these methods is compatible with the aforementioned regulations or bills, as they suffers from the high risk of leaking private information.

Recently, researchers propose a novel privacy-preserving collaborative learning paradigm, federated learning [46, 76] to deal with the dilemma about the utilization and protection of data. Federated learning is combined with various machine learning algorithms such as neural networks [15, 46], SVMs [10], and Logistic regression [27]. As one of key motivations of federated learning is to protect the data and model from privacy leakage while collaborative learning, the proposed federated learning algorithms have to claim what level of the protection they can provide and how they can achieve. To ensure a higher level of data security, some federated learning algorithms introduce mechanisms in secure multi-party computing and cryptography. For example, work in Reference [9] improves the the horizontal federated learning algorithm by introducing the secure aggregation protocol. Works proposed by literatures [16, 43] introduce **homomorphic encryption (HE)** [62] to protect the intermediate result between federated parties. However, HE-based methods bring extra encryption computation costs and larger communication cost due to the ciphertext. Secure multi-party computation [77] methods are also used for collaborative learning, but the number of participants in this protocol are limited. Besides, due to the probabilistic property of differential privacy [21], differentially privacy is widely used in federated learning for protecting the transaction of models or data [5, 24, 28, 30, 57, 74]. The benefits of differential privacy techniques involves two phrase: (1) first it does not need extra communication and computation costs of transmitting encryption keys as well as encryption and decryption; (2) we can control the balance between privacy protection and data utility. In our work, we apply differential privacy to protect local model in to avoid potential privacy leakage.

2.4 Evolutionary Learning

Evolutionary learning [31] is a class of stochastic search and optimization techniques inspired by the natural evolution of biological systems. With the spirit of global search ability, parallelism, and flexibility, evolutionary learning has been widely applied to image classification [3, 4], object detection [11, 42], regression [34, 64], scheduling, and combinatorial optimization [33, 44, 53, 54, 80]. Recently, with the trend of deep learning, evolutionary learning has also been combined with the

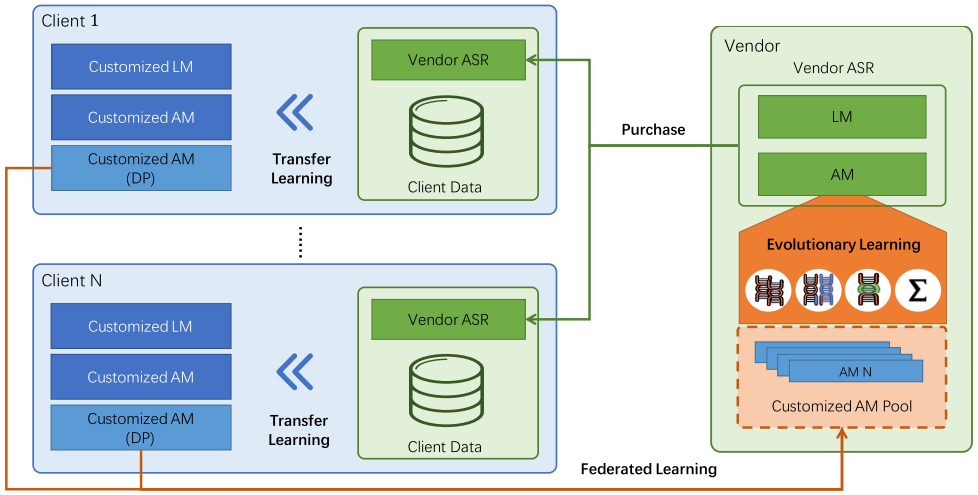

Fig. 1. Schematic Diagram of the TFE Ecosystem. Clients purchase LM and AM from the vendor, and tune both models on their local data using transfer learning. After that, customized AMs are collected onto vendor via federated learning, which will be utilized by evolutionary learning to improve the vendor's AM.

deep neural networks and shown encouraging performance, such as neural network optimization [38, 50, 69] and neural architecture search [12, 60, 61].

In evolutionary learning, a population of candidate solutions (individuals) is initialized and evolved towards good/optimal solutions through an evolutionary process. In each generation of the evolutionary process, each individual is evaluated by a fitness function, which measures the performance of the individual on the target task. Then only the fittest individuals are selected as parents to breed the new individuals through genetic operators (e.g., crossover and mutation), passing their characteristics to the next generation. With the iterative selection and reproduction, the initial population is improved until the final population achieves the maximized fitness. According to the genetic representation of individual, evolutionary learning algorithms can be roughly categorized into genetic algorithm and genetic programming. While genetic algorithm employs a a fixed-length string of genes (bits, real numbers, or symbols), genetic programming works with more flexible structures such as trees and graphs with variable sizes.

Noticeably, evolutionary learning is naturally suitable to cope with federated learning as it plays as an powerful tool for optimizing model in a distributed environment. Zhu and Jin [84] adapt evolutionary learning to assist model optimization by learning to reducing connectivity of networks and improving communication efficiency. Zou et al. [85] set up an evolutionary game environment for every mobile device to find their optimal training strategy that maximizing their utilities.

In this work, we leverage the genetic algorithm to find a better vendor ASR system by evolving the population of customized models and original vendor model.

3 THE TFE ECOSYSTEM

Here, we propose TFE, which is a win-win ecosystem for ASR clients and vendors with revolutionary performance. As illustrated in Figure 1, TFE seamlessly integrates three machine learning paradigms: transfer learning, federated learning and evolutionary learning. Transfer learning is applied to build a highly customized ASR system for each client. Federated learning is employed to transfer the tuned AMs from the clients to the vendor with **Local Differential Privacy (LDP)**.

Evolutionary learning is responsible for composing the next generation of ASR system by integrating the tuned ones collected from the clients.

3.1 Transfer Learning for Client

TFE provides a principled way for each client to obtain a highly customized ASR system, which performs significantly better than its "one-size-fits-all" counterparts. Since the client does not necessarily have a large amount of labeled speech data nor powerful computational capability, we resort to transfer learning to conduct customization for both LM and AM. Transfer learning provides a methodology to leverage the knowledge learned from a data-sufficient domain to help the learning in data-limited target domain [56]. Such philosophy works for ASR if we consider the vendor ASR system as the knowledge from data-sufficient domain and the client scenario as the data-limited target domain.

3.2 Transfer Learning for LM

As the most common choice in ASR system, the backoff n-gram model is adopted in TFE as the language model. Technically, the backoff n-gram LM can be considered as a list of tuples, each of which contains an n-gram as well as its corresponding logarithm probability. The transfer learning of LM is implemented by interpolating the LM trained on private data with the vendor LM as follows:

$$P(\mathbf{w}) = \lambda P^{V}_{LM}(\mathbf{w}) + (1-\lambda)P^{C}_{LM}(\mathbf{w}), \tag{1}$$

where $P(\mathbf{w})$ indicates the probability of the n-gram $\mathbf{w}$, $P^{V}_{LM}(\mathbf{w})$ is probability given by the vendor LM and $P^{C}_{LM}(\mathbf{w})$ is the probability given by the LM trained on the client data. This kind of transfer learning is effective to boost the probability of client-domain n-grams while preserving the wide coverage of general-purpose n-grams. λ is a hyper-parameter that controls the weights of the vendor and client language model.

3.3 Transfer Learning for AM

As we mentioned in Section 2, modern deep learning–based AMs can be classified as the DNN-HMM model and the end-to-end model. The DNN-HMM AM is a stacked model, with DNN responsible for extracting high-level features from acoustic signal such as MFCCs, and HMM responsible for modeling lexical sequence such as phonemes according to the features extracted by the DNN, where the lexical sequence is necessary for decoding into transcripts. The DNN component is essentially a neural network with acoustic features as input and senone as the output. Senones are the context-dependent lexical units (triphoneme), which are to the states of the downstream HMM component. However, the end-to-end model is purely a DNN, which also takes acoustic features as the input, but directly outputs the recognition results.

For either kind of AM, we only apply transfer learning to the DNN part, which means the HMM component of the HMM-DNN model is fixed during transfer learning. As mentioned in the related work, there are many transfer learning techniques for acoustic model, including linear transformation, conservative training, and subspace method. In this work, we focus on acoustic model adaptation for neural network, namely, conservative training (model-based transfer learning). Yosinki et al. [79] provide a thorough investigation on the effects of transfer learning on different layers of a deep neural network, leading to a conclusion that initial izing with transferred features helps boost the generalization performance, as well as fine-tuning from a well-trained neural work. They also reveal the fact that different layers capture different types of information, which may have different effect on the target task.

Assume the loss functions for training DNNs in the source domain and the target domain are $\mathcal{L}_S$ and $\mathcal{L}_T$, respectively. In the view of model-based transfer learning with KLD-regularization [32], our approach can be mathematically summarized as

$$\mathcal{L}_T = (1-\rho)\mathcal{L}_S + \frac{\rho}{N}\sum_{(x,y)\in\text{Target}} p_S(y|x)\log p_T(y|x), \tag{2}$$

where p_S and p_D are the posteriors associated the models of the source domain and the target domain, respectively, (x, y) is the data sample collected from the target domain, N is the number of such data samples in the target domain, and ρ is a hyper-parameter that controls the transfer ratio from the source domain. The KL-divergence in the last term prevents overtraining and keeps the adapted model from straying too far from the source domain model.

3.4 Federated Learning between Client and Vendor

To protect data privacy, the client's private data cannot be straightly transmitted to the vendor. Instead, we utilize the components of the customized ASR systems as the medium to convey necessary information for the vendor to further refine its ASR system. As obtaining a large volume of text corpus is relatively easy and the privacy-preserving approach of training language models is well documented in literature [29], here we focus on collecting the customized AMs from the clients.

We employ federated learning [46, 76] to transmit customized AMs from the clients to the vendor in a privacy-preserving approach. The aim of federated learning is to resolve the dilemma of data privacy and utility. With federated learning, each client train a secure version of the customized AMs and transmit it to the vendor.

Since the customized AMs are obtained through mini-batch **stochastic gradient descent (SGD)**, its gradients may expose information about the client's private data. Hence, it is necessary to generate a secure version of the AM through encrypting the gradients by **local differential privacy (LDP)** [1], and we denote this secure version as DP-AM thereafter. LDP employ the Laplace scheme to inject Laplacian noise into SGD gradient [22]. The Laplace scheme usually requires the input to be bounded, which is not satisfied by the gradients of DNN. Therefore, we first need to scale the mini-batch gradient as follows:

$$\Delta w_t' = \frac{\Delta w_t}{\max\{1,\ \|\Delta w_t\|_2/S\}}, \tag{3}$$

where Δw_t is the mini-batch gradient, and S is a user-specified parameter and is usually called sensitivity in differential privacy literature. By doing this, it is ensured that the scaled gradient is upper bounded by S. Now, we are ready to update the DNN parameters by a stochastic gradient descent step with injected noise:

$$w_{t+1} = w_t - \eta_t(\Delta w_t' + \delta_t). \tag{4}$$

Here δ_t is a random noise subjecting to Laplace distribution, and η_t is the learning rate. More specifically, if we assume the batch size of SGD is chosen as b, then noise δ_t should be drawn from the following probability density function:

$$p(\delta_t) \propto \exp\left\{-\frac{S}{b\epsilon}\|\delta_t\|_2\right\}, \tag{5}$$

where the parameter ϵ and δ measure the level of such risk. If ϵ is set to be small, then the probability of leaking data is small but the quality of DP-AM may degrade due to large gradient noise. It can be shown that process of training DP-AM is ϵ-differential private [65]. In TFE, these DP-AMs work as secure proxies of the clients' private data and are transmitted to the vendor. The property

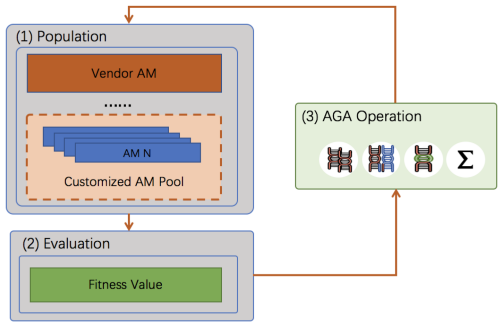

Fig. 2. Pipleline of the AGA Algorithm.

of LDP ensures that it is hard to recover the private data from checking the difference between original vendor AM and the DP-AMs.

3.5 Evolutionary Learning for Vendor

As it is the DP-AMs rather than speech data is transmitted to the vendor, a new training method is required by the vendor to exploit the value of these DP-AMs.

We resort to evolutionary learning for generating the new vendor AM by integrating the DP-AMs collected from the clients. As shown in Figure 2, we propose the **Acoustic Genetic Algorithm (AGA)**, and it contains four steps: initialization, selection, genetic operators and termination, each of which roughly corresponds to a particular facet of natural evolution. The goal of AGA is to minimize the **Word Error Rate (WER)** (see Section 4.1.3 for its detail definition) of the vendor's AM. The workflow of the AGA is presented as follows:

(1) Initialize the population with the original vendor AM and all the customized AMs encrypted by differential privacy. Each AM is divided into multiple components and each component is further encoded into a bit string with their corresponding model parameters. The genes of each AM is represented by the concatenation of the bit strings of its components.
(2) For each AM, compute its WER on a validation dataset to measure its fitness and individuals with large fitness are selected as the parents of the next generation.
(3) Populate next generation from the parents through a combination of genetic operators: Reproduction, Crossover, Mutation and WeightedAverage:
   - **Reproduction** copies the selected parents from the current population into the new population without alteration, to retain the best-so-far AMs.
   - **Crossover** recombines the genes from two selected parents to create their children. Specifically, we employ the **one-point crossover (P1XO)**, where a random crossover point is selected and the tails of parents' genes are swapped to get the new children:

$$\begin{aligned} C'_{1,<p} &= C_{1,<p}, \\ C'_{2,<p} &= C_{2,<p}, \\ C'_{1,>=p} &= C_{2,>=p}, \\ C'_{2,>=p} &= C_{1,>=p}, \end{aligned} \tag{6}$$

   where p is the random crossover point, $C_{\cdot,<p}$ and $C_{\cdot,>=p}$ are the head and tail genes of parents, $C'_{\cdot,<p}$ and $C'_{\cdot,>=p}$ are the head and tail genes of children.

- **Mutation** stochastically alters the genes of an individual to introduce more diversity into the population, allowing more solution space to be searched. In this work, We employ the **single-point mutation (SPM)**, which randomly flips the bits in genes.
- **WeightedAverage** is an operator inspired by the success of the FedAvg algorithm [45]. The FedAvg algorithm shows that the models independently trained on different data sets with the same random initialization can be aggregated to a better model by simple parameter averaging. Therefore, we further propose to create a child by weighted averaging two parents as follows:

$$w' = \lambda\, w_1 + (1 - \lambda)\, w_2, \tag{7}$$

where λ is randomly drawn from $(0, 1)$, w_1 and w_2 are the model parameters of parents, and w' is the model parameters of child.

(4) Repeat Step (2) to Step (3) until the evolution has reached the maximum number of generations. Then the AM with the best performance is kept as the new vendor AM.

In terms of Step (3), we first copy all the selected parents by reproduction, then each parent is also mutated to generate one extra offspring. While for Crossover and WeightedAverage, considering that they require a pair of parents as the input, and the number of parent pairs are too large, we only sample a subset of parent pairs for each operator, and offsprings are merely generated from this subset.

With the aid of this scheme, the vendor can extract valuable information from the collected clients' models, and improve its AM in an iterative way.

Finally, to have a clearer understanding of how transfer learning, federated learning and evolutionary learning are composed together, we summarize our framework in Algorithm 1 and Algorithm 2, where the behaviors of the vendor and clients are presented, respectively.

4 EXPERIMENTS

In this section, we present the experimental results. We first describe the experimental setup in Section 4.1. Then, we present the results of transfer learning in Section 4.2. We show the result of federated learning and evolutionary learning in Section 4.3. Finally, we compare the models generated by our TFE framework with human workers in Section 4.4.

ALGORITHM 1: TFE framework: vendor's side

```
Send LM_vendor and AM_vendor to all clients;
Receive all the customized AMs, say DP-AM_1, DP-AM_2, ..., DP-AM_n, from clients;
// evolutionary learning
Initialize AM population: P = {DP-AM_1, DP-AM_2, ..., DP-AM_n};
for t = 1, 2, ... do
    For each model or pair of models in P, apply the genetic operators onto it, respectively, to
    generate offspring;
    Evaluate the WER of each generated model;
    Update P to be the set of top-K models with lowest WERs;
end
Let AM_vendor to be best model in P;
return: AM_vendor as the new vendor's AM
```

Table 1. Details of the Datasets of the 10 Clients

Client	Domain	Training		Testing	
		No. of Waves (K)	Duration (h)	No. of Waves (K)	Duration (h)
1	radio programs	1,009.2	1,000.4	101.0	99.9
2	telephone customer service	7.5	3.9	1.9	0.9
3	daily conversations	361.7	211.7	67.2	35.0
4	dialogs with chatbots	235.5	231.5	46.2	41.3
5	daily conversations	74.8	88.2	7.4	8.6
6	news	152.6	225.4	7.2	10.0
7	news	151.8	204.7	7.2	10.0
8	voice commands	90.4	87.1	17.3	19.5
9	voice messages	113.7	142.8	17.4	22.4
10	daily conversations	122.8	74.6	19.4	11.6
Sum		2,319.9	2,270.3	292.1	259.3

ALGORITHM 2: TFE framework: client k's side

Receive $\text{LM}_{\text{vendor}}$ and $\text{AM}_{\text{vendor}}$ from vendor;
// transfer learning
Transfer LM_k from $\text{LM}_{\text{vendor}}$ according to (1);
Transfer AM_k from $\text{AM}_{\text{vendor}}$ by minimizing loss $\mathcal{L}_T$ in (2);
// federated learning
Initialize: DP-AM_k = $\text{AM}_{\text{vendor}}$;
while *not converged* **do**
  Iterative update DP-AM_k by (4), where $\Delta w = \nabla \mathcal{L}_T(w)$;
end
Send DP-AM_k to the vendor;
return: LM_k and AM_k as the local LM and AM

4.1 Experimental Setup

4.1.1 Dataset. We conduct large-scale experiments with a setting of 1 vendor and a varying number (maximum 10) of clients. The vendor's original ASR system is pre-trained with more than 10,000 h of speech data. The clients have speech data collected from very diverse scenarios, such as the radio programs, voice mail messages and daily conversations. The sizes of the clients' private data also vary a lot, ranging from around 5 h of speech data to more than 1,000 h, which makes the locally transfer-learnt models to have very different properties. All the speech data are stored in wav format with sampling rate 16 kHz. Each client's private data is randomly split into training and testing data, where the training data is used for local transfer learning on clients, and the testing data is retained for evaluating the model performance. We list the detailed statistics of this data set in Table 1.

4.1.2 System Implementation. The TFE framework is built through a full-fledged ASR system trained by the open-source Kaldi toolkit [59]. The AM is the Kaldi "Chain" model that adopts the DNN-HMM architecture and the LM is a backoff trigram model that is trained by the **SRI Language Modeling Toolkit (SRILM)** [66]. For the system architecture and deployment, such as the basic communication efficiency and protocols, we mainly build the framework based on

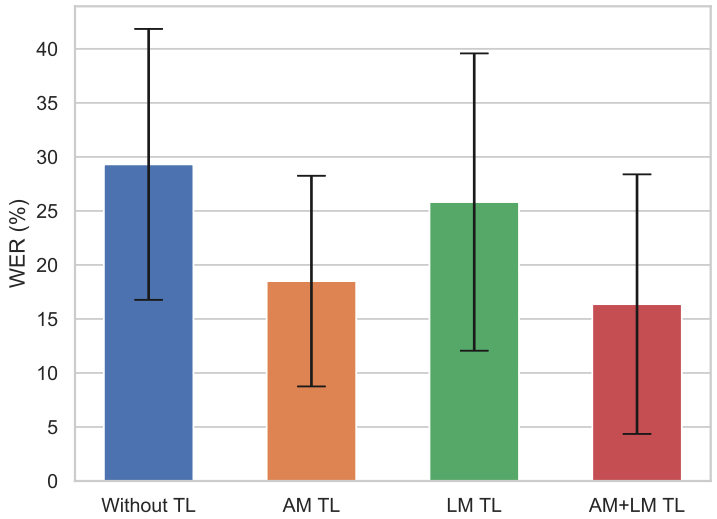

Fig. 3. Performance of different Transfer Learning (TL) settings.

similar parts of the **Federated AI Technology Enabler (FATE)**[1] platform. From the implementation perspective, it is worth noting that the transfer learning component, the federated learning component and the evolutionary learning component are loosely coupled. Each of the three component can work independently as long as their inputs follow the requirements in Figure 1. The whole system is deployed on a cluster of 11 nodes, one of which is the vendor while all the others are the clients. The hardware configuration of each machine is 314GB memory, Intel Xeon Processor with 72 cores, Tesla K80 GPU and CentOS.

4.1.3 Evaluation Metric. We quantitatively gauge the the performance of TFE through the standard metric WER [35]. WER is a widely used metric to gauge performance of an ASR system. It compares a reference to a hypothesis and is defined as follows:

$$\text{WER} = \frac{S + D + I}{N}, \tag{8}$$

where S, D, and I are the minimum number of substitutions, deletions, and insertions, respectively, to turn hypothesis into reference, and N is the number of words in the reference. The lower the WER, the better the performance of the corresponding ASR system (i.e., better AM or LM).

In what follows, we elaborate on how TFE effectively solves the problems plaguing ASR clients and vendors.

4.2 Evaluation of Transfer Learning

We quantitatively evaluate the effectiveness of transfer learning mechanism for each client. We first distribute the vendor's AM and LM to all the clients, then each client tunes the AMs and LMs with its private data. For each locally tuned customized model, we evaluate its WER on its corresponding testing set, and report the mean values and the variances of the WERs in Figure 3. We observe that either for AM or LM, transfer learning does reduce WER for each client. Especially, transfer learning for AM brings more improvement comparing to LM. Applying transfer learning

[1] https://gitee.com/WeBank/FATE.

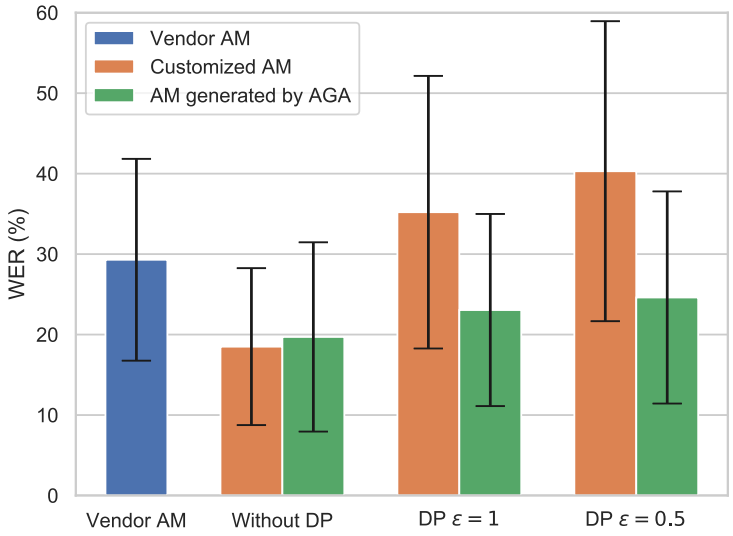

Fig. 4. Comparison of the original vendor AM, the customized AMs and the new AM generated by AGA.

simultaneously for AM and LM achieves the best performance. Such observation shows that transfer learning effectively smooths out the discrepancy between vendor data and client data. Hence, for the clients, TFE has incomparable advantage over the "one-size-fits-all" counterparts and those with simply tuned LMs through privacy-violating approach.

4.3 Evaluation of Federated Learning and Evolutionary Learning

To evaluate the efficiency of AGA, we collect the customized acoustic models from the previous subsection with federated learning, where different level of LDP noise is injected to protect clients' private data, and then merge these models by utilizing AGA. To observe the long-term behavior of AGA, the maximum number of generations is set to a fairly large number 150 so that the WERs of the generated models does not decrease anymore. For the finally produced vendor AM, we still evaluate it on all the 10 testing sets. Besides that, the customized models are also evaluated on their corresponding testing sets as baselines. The results are reported in Figure 4. It can be observed that the new vendor AMs generated by AGA, either based upon clean customized AMs or DP-AMs, are significantly better than the original vendor AM, which implies that customized AMs are informative for the vendor. Though the new vendor AM is worse than the customized models when LDP is not incorporated, it is understandable, since each of the customized models is well calibrated for its special setting, while the vendor AM is a single model needed to fit all the scenarios. Moreover, the performance difference between them is actually quite limited. However, when LDP noise is added, the customized models soon deteriorate, but the models generated AMs still maintain good performance. This phenomenon suggests that AGA is able to extract valuable information from AMs and effectively refine the vendor AM, even when moderate noise is injected.

Though it is already justified that AGA can greatly boost the performance of the generated models on the clients, it is important to evaluate whether the new AM generated by AGA generalizes well on the domains beyond those of the participating clients, since new ASR clients are not necessarily from the domains of current TFE participants. To achieve such goal, we look into the AM generated by AGA in a five-client setting and gauge its performance on the testing data from the domains of the other five clients. The results are presented in Figure 5. As for the scenario that

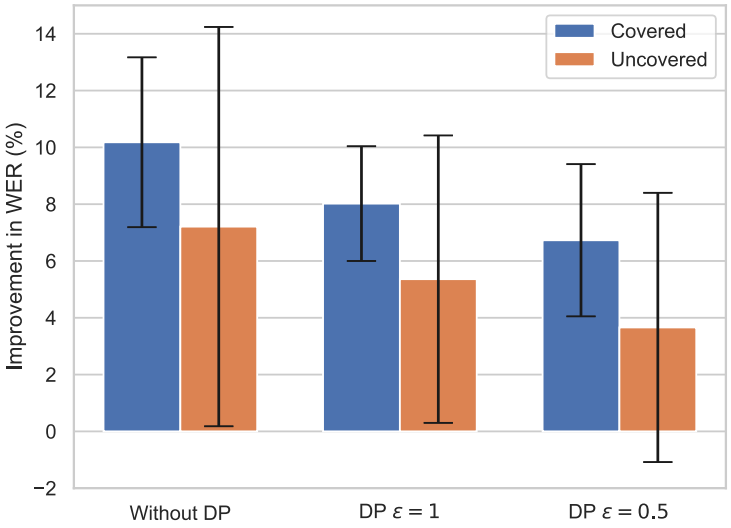

Fig. 5. WER Improvement of AGA trained with five-client setting. Covered refers to the case that the domains of the testing data and those of the five clients are the same, while uncovered means they have no overlap.

testing domains are covered by the training ones, the AM generated by AGA demonstrates larger performance improvement for vendor AM. Importantly, as for the scenario that testing domains are not covered by the training ones, the AM generated by AGA is still robust enough to achieve fairly good improvement compared with the original vendor AM. This observation is crucial for ASR vendors, since they can rely on TFE to consistently improve the experience of prospective clients. But of course, the improvement on the uncovered domains is not as significant as the covered ones. Besides, the variances on the uncovered domains are much larger. Especially for the case $\epsilon = 0.5$, the performance on one of the testing domains even worsen.

Another interesting study is to explore the performance of AGA by varying the number of clients participating federated learning. Though the number of clients is varied for training process, the testing data are always fixed as all the 10 testing sets to make the comparison fair. The results are reported in Figure 6. For the AMs without DP noise, more clients bring better performance, indicating that the diversity of AMs is a crucial factor for the performance AGA. However, when DP noise is involved, the story becomes different. As for either $\epsilon = 1$ or $\epsilon = 0.5$, the DP-AMs produced from five-client setting are better than their two-client counterpart. Surprisingly, it also outperforms the AM from the ten-client setting. It is likely that one of the other five models that does not participate in the five-client setting encodes wrong information after noise injection. Then, AGA with 10 clients is misled by it. This phenomenon suggests that increasing the number of clients does not always do help when random noise exists. A moderate amount of clients provides sufficient diversity and avoids involving much noise from the LDP. As a future work, we might explore how to filter out the contaminated models, which are harmful to AGA.

4.4 Comparison with Human Transcribers

Though we have demonstrated that the proposed TFE framework can greatly decrease the WERs of the ASR system, it is not clear what these values of WERs imply. To have a explicit understanding of the effectiveness of the proposed TFE, we compare the performance of our models with human transcribers to see if TFE has to potential to generate optimized ASR system whose performance is close to humans. We prepare 100 audios from a domain beyond those studied in previous study as

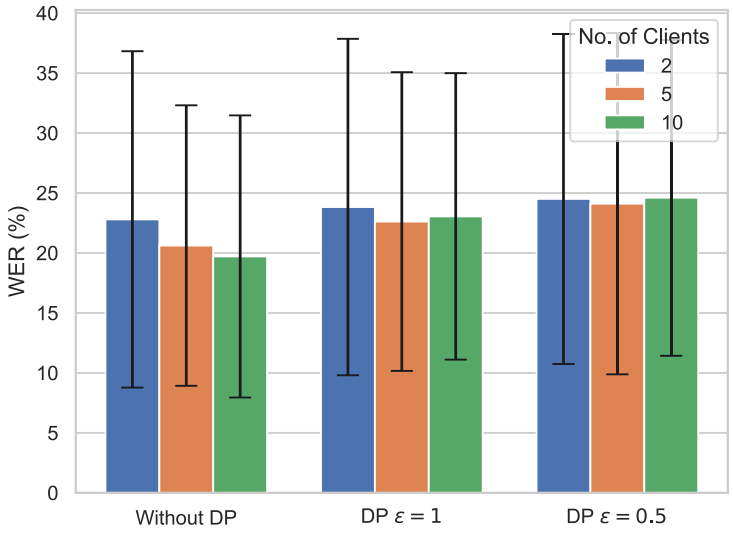

Fig. 6. The performance of AGA with different number of clients. The WER is calculated on the same testing dataset covering all the ten domains.

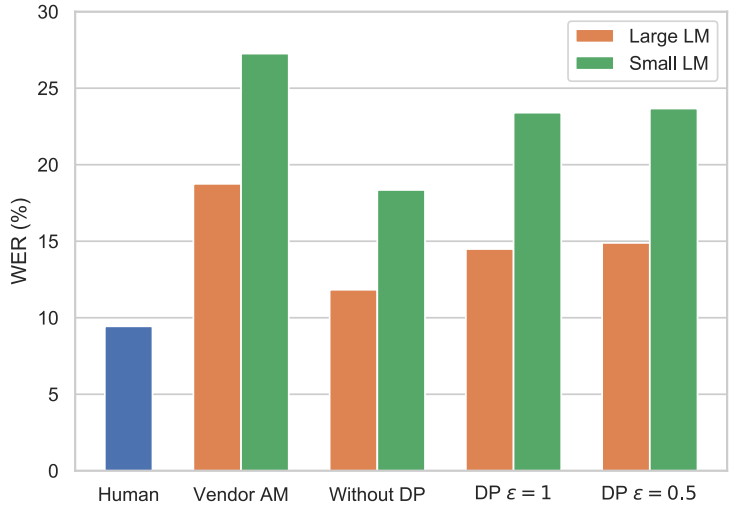

Fig. 7. Comparison with human workers.

the test data. All audios are split into short segments to make it easy for human worker to memorize the sentences. Following the human labeling process in Reference [67], a two-pass pipeline is utilized for human transcribers: The first transcriber works from scratch to label the data and the second conducts error correction. Each transcriber is restricted to listen to the audio once. All the transcribers are native speakers.

The LM used in the previous experiments is referred to as small LM (500 MB) while a large LM (8 GB) trained with more corpus is also introduced in this experiment. Based on the results presented in Figure 7, we observe that the WER of human transcribers is 9.4%, which is just half of

the WER of the original vendor AM, even when the large LM is applied. However, after the AM is improved by the TFE, the best model recorded a WER of 11.8%, not far from the performance of human. Though the performance worsen after the combination of DP, the final AMs still significantly outperforms the original vendor AM. These results are consistent with Figure 4 and provide further evidence of the effectiveness of TFE.

5 CONCLUSION

In this article, we propose a new ASR ecosystem named TFE, which solves the most challenging problems faced by ASR clients and vendors in the era of data regulations like GDPR coming into force. By transfer learning, TFE provides superior ASR products for each client. Through federated learning and evolutionary learning, TFE provides ASR vendors with necessary information to consistently refine their product in a GDPR-compliant way. We wish that TFE paved the way for building a prosperous ecosystem for the whole ASR industry.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. ACM, 308–318.

[2] Victor Abrash, Horacio Franco, Ananth Sankar, and Michael Cohen. 1995. Connectionist speaker normalization and adaptation. In *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech'95)*. Citeseer.

[3] Harith Al-Sahaf, Ausama Al-Sahaf, Bing Xue, Mark Johnston, and Mengjie Zhang. 2017. Automatically evolving rotation-invariant texture image descriptors by genetic programming. *IEEE Trans. Evolution. Comput.* 21, 1 (2017), 83–101.

[4] Wissam A. Albukhanajer, Johann A. Briffa, and Yaochu Jin. 2014. Evolutionary multiobjective image feature extraction in the presence of noise. *IEEE Trans. Cybernet.* 45, 9 (2014), 1757–1768.

[5] Johes Bater, Xi He, William Ehrich, Ashwin Machanavajjhala, and Jennie Rogers. 2018. Shrinkwrap: Differentially-private query processing in private data federations. Retrieved from https://arXiv:1810.01816.

[6] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.* 3 (Feb. 2003), 1137–1155.

[7] Peva Blanchard, Rachid Guerraoui, Julien Stainer, et al. 2017. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*. MIT Press, 119–129.

[8] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.* 3 (2003), 993–1022.

[9] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2016. Practical secure aggregation for federated learning on user-held data. Retrieved from https://arXiv:1611.04482.

[10] Theodora S. Brisimi, Ruidi Chen, Theofanie Mela, Alex Olshevsky, Ioannis Ch Paschalidis, and Wei Shi. 2018. Federated learning of predictive models from federated Electronic Health Records. *Int. J. Med. Info.* 112 (2018), 59–67.

[11] Armand R. Burks and William F. Punch. 2018. Genetic programming for tuberculosis screening from raw X-ray images. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'18)*. 1214–1221.

[12] Boyuan Chen, Harvey Wu, Warren Mo, Ishanu Chattopadhyay, and Hod Lipson. 2018. Autostacker: A compositional evolutionary learning system. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'18)*. 402–409.

[13] Kuan-Yu Chen, Hsuan-Sheng Chiu, and Berlin Chen. 2010. Latent topic modeling of word vicinity information for speech recognition. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP'10)*. IEEE, 5394–5397.

[14] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. Retrieved from https://arXiv:1712.05526.

[15] Yiqiang Chen, Xin Qin, Jindong Wang, Chaohui Yu, and Wen Gao. 2020. Fedhealth: A federated transfer learning framework for wearable healthcare. *IEEE Intell. Syst.* 35, 4 (2020), 83–93.

[16] Kewei Cheng, Tao Fan, Yilun Jin, Yang Liu, Tianjian Chen, and Qiang Yang. 2019. SecureBoost: A lossless federated learning framework. Retrieved from http://arxiv.org/abs/1901.08755.

[17] Alexandra Chronopoulou, Christos Baziotis, and Alexandros Potamianos. 2019. An embarrassingly simple approach for transfer learning from pretrained language models. Retrieved from https://arXiv:1902.10547.
[18] George E Dahl, Dong Yu, Li Deng, and Alex Acero. 2011. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Trans. Audio, Speech, Lang. Process.* 20, 1 (2011), 30–42.
[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. Retrieved from https://arXiv:1810.04805.
[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. Retrieved from https://arXiv:1810.04805.
[21] Cynthia Dwork. 2008. Differential privacy: A survey of results. In *Proceedings of the Theory and Applications of Models of Computation 5th International Conference (TAMC'08)*. 1–19.
[22] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.* 9, 3–4 (2014), 211–407.
[23] Roberto Gemello, Franco Mana, Stefano Scanzio, Pietro Laface, and Renato De Mori. 2007. Linear hidden transformations for adaptation of hybrid ANN/HMM models. *Speech Commun.* 49, 10 (2007), 827–835.
[24] Robin C. Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially private federated learning: A client level perspective. Retrieved from https://arXiv:1712.07557.
[25] Shweta Ghai and Rohit Sinha. 2016. Adaptive feature truncation to address acoustic mismatch in automatic recognition of children's speech. *APSIPA Trans. Signal Info. Process.* 5 (2016).
[26] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*. ACM, 369–376.
[27] Xiawei Guo, Quanming Yao, WeiWei Tu, Yuqiang Chen, Wenyuan Dai, and Qiang Yang. 2018. Privacy-preserving Transfer Learning for Knowledge Sharing. Retrieved from https://arXiv:1811.09491.
[28] Jihun Hamm, Yingjun Cao, and Mikhail Belkin. 2016. Learning privately from multiparty data. In *Proceedings of the International Conference on Machine Learning*. 555–563.
[29] Andrew Hard, Kanishka Rao, Rajiv Mathews, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated learning for mobile keyboard prediction. Retrieved from https://arXiv:1811.03604.
[30] Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Richard Nock, Giorgio Patrini, Guillaume Smith, and Brian Thorne. 2017. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. Retrieved from https://arXiv:1711.10677.
[31] John H. Holland. 1992. *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge, MA.
[32] Yan Huang, Dong Yu, Chaojun Liu, and Yifan Gong. 2014. Multi-accent deep neural network acoustic model with accent-specific top layer using the KLD-regularized model adaptation. In *Proceedings of the 15th Annual Conference of the International Speech Communication Association*.
[33] Josiah Jacobsen-Grocott, Yi Mei, Gang Chen, and Mengjie Zhang. 2017. Evolving heuristics for dynamic vehicle routing with time windows using genetic programming. In *Proceedings of the IEEE Congress on Evolutionary Computation, (CEC'17)*. 1948–1955.
[34] Yanfei Kang, Rob Hyndman, and Smith-Miles Kate. 2017. Visualising forecasting algorithm performance using time series instance spaces. *Int. J. Forecast.* 33, 2 (2017), 345–358.
[35] Dietrich Klakow and Jochen Peters. 2002. Testing the correlation of word error rate and perplexity. *Speech Commun.* 38, 1–2 (2002), 19–28.
[36] Roland Kuhn and Renato De Mori. 1990. A cache-based natural language model for speech recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 12, 6 (1990), 570–583.
[37] Raymond Lau, Ronald Rosenfeld, and Salim Roukos. 1993. Trigger-based language models: A maximum entropy approach. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 2. IEEE, 45–48.
[38] Joel Lehman, Jay Chen, Jeff Clune, and Kenneth O. Stanley. 2018. ES is more than just a traditional finite-difference approximator. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'18)*. 450–457.
[39] Bo Li and Khe Chai Sim. 2010. Comparison of discriminative input and output transformations for speaker adaptation in the hybrid NN/HMM systems. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association*.
[40] Ke Li, Hainan Xu, Yiming Wang, Daniel Povey, and Sanjeev Khudanpur. 2018. Recurrent neural network language model adaptation for conversational speech recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'18)*. 1–5.
[41] Xiao Li and Jeff Bilmes. 2006. Regularized adaptation of discriminative classifiers. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'06)*, Vol. 1. IEEE, I–I.

[42] Yuyu Liang, Mengjie Zhang, and Will N. Browne. 2015. A supervised figure-ground segmentation method using genetic programming. In *Proceedings of the European Conference on the Applications of Evolutionary Computation*. 491–503.

[43] Yang Liu, Tianjian Chen, and Qiang Yang. 2018. Secure federated transfer learning. Retrieved from http://arxiv.org/abs/1812.03337.

[44] Yuxin Liu, Yi Mei, Mengjie Zhang, and Zili Zhang. 2017. Automated heuristic design using genetic programming hyper-heuristic for uncertain capacitated arc routing problem. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'17)*. 290–297.

[45] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS'17)*. 1273–1282.

[46] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. 2016. Communication-efficient learning of deep networks from decentralized data. Retrieved from https://arXiv:1602.05629.

[47] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association*.

[48] Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'11)*. IEEE, 5528–5531.

[49] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*. MIT Press, 3111–3119.

[50] David J. Montana and Lawrence Davis. 1989. Training feedforward neural networks using genetic algorithms. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'89)*. 762–767.

[51] Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'05)*, Vol. 5. Citeseer, 246–252.

[52] Joao Neto, Luís Almeida, Mike Hochberg, Ciro Martins, Luis Nunes, Steve Renals, and Tony Robinson. 1995. Speaker-adaptation for hybrid HMM-ANN continuous speech recognition system. In *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech'95)*. 2171–2174.

[53] Su Nguyen, Yi Mei, and Mengjie Zhang. 2017. Genetic programming for production scheduling: A survey with a unified framework. *Complex Intell. Syst.* 3, 1 (2017), 41–66.

[54] Su Nguyen, Mengjie Zhang, Mark Johnston, and Kay Chen Tan. 2014. Automatic design of scheduling policies for dynamic multi-objective job shop scheduling via cooperative coevolution genetic programming. *IEEE Trans. Evolution. Comput.* 18, 2 (2014), 193–208.

[55] Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* 22, 10 (2010), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

[56] Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* 22, 10 (2010), 1345–1359.

[57] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. 2016. Semi-supervised knowledge transfer for deep learning from private training data. Retrieved from https://arXiv:1610.05755.

[58] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543.

[59] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The Kaldi speech recognition toolkit. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society.

[60] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V. Le. 2018. Regularized evolution for image classifier architecture search. Retrieved from https://arXiv:1802.01548.

[61] Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Jie Tan, Quoc V. Le, and Alexey Kurakin. 2017. Large-scale evolution of image classifiers. In *Proceedings of the International Conference on Machine Learning (ICML'17)*. 2902–2911.

[62] Ronald L. Rivest, Len Adleman, Michael L. Dertouzos, et al. 1978. On data banks and privacy homomorphisms. *Found. Secure Comput.* 4, 11 (1978), 169–180.

[63] Natasha Singh-Miller and Michael Collins. 2007. Trigger-based language modeling using a loss-sensitive perceptron algorithm. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'07)*, Vol. 4. IEEE, IV–25.

[64] Ankur Sinha, Pekka Malo, and Timo Kuosmanen. 2015. A multiobjective exploratory procedure for regression model selection. *J. Comput. Graphic. Stat.* 24, 1 (2015), 154–182.

[65] Shuang Song, Kamalika Chaudhuri, and Anand D. Sarwate. 2013. Stochastic gradient descent with differentially private updates. In *Proceedings of the IEEE Global Conference on Signal and Information Processing*. IEEE, 245–248.
[66] Andreas Stolcke. 2002. SRILM-an extensible language modeling toolkit. In *Proceedings of the 7th International Conference on Spoken Language Processing*.
[67] Andreas Stolcke and Jasha Droppo. 2017. Comparing human and machine errors in conversational speech transcription. In *Proceedings of the Interspeech Conference*. 137–141. https://academic.microsoft.com/paper/2963980299
[68] Baochen Sun and Kate Saenko. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *Proceedings of the European Conference on Computer Vision*. Springer, 443–450.
[69] Yanan Sun, Gary G. Yen, and Zhang Yi. 2019. Evolving unsupervised deep neural networks for learning meaningful representations. *IEEE Trans. Evolution. Comput.* 23, 1 (2019), 89–103.
[70] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*. MIT Press, 3104–3112.
[71] Jan Trmal, Jan Zelinka, and Luděk Müller. 2010. Adaptation of a feedforward artificial neural network using a linear transform. In *Proceedings of the International Conference on Text, Speech and Dialogue*. Springer, 423–430.
[72] Paul Voigt and Axel Von dem Bussche. 2017. The EU general data protection regulation (GDPR). *A Practical Guide,* 1st ed. Springer International Publishing, Cham.
[73] Jindong Wang, Yiqiang Chen, Wenjie Feng, Han Yu, Meiyu Huang, and Qiang Yang. 2020. Transfer learning with dynamic distribution adaptation. *ACM Trans. Intell. Syst. Technol.* 11, 1 (2020), 1–25.
[74] Yang Wang, Quanquan Gu, and Donald Brown. 2018. Differentially private hypothesis transfer learning. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 811–826.
[75] Hainan Xu, Ke Li, Yiming Wang, Jian Wang, Shiyin Kang, Xie Chen, Daniel Povey, and Sanjeev Khudanpur. 2018. Neural network language modeling with letter-based features and importance sampling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'18)*. IEEE, 6109–6113.
[76] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.* 10, 2 (2019), 12.
[77] Andrew Chi-Chih Yao. 1982. Protocols for secure computations. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS'82)*, Vol. 82. 160–164.
[78] Jiangyan Yi, Hao Ni, Zhengqi Wen, Bin Liu, and Jianhua Tao. 2016. CTC regularized model adaptation for improving LSTM RNN based multi-accent Mandarin speech recognition. In *Proceedings of the 10th International Symposium on Chinese Spoken Language Processing (ISCSLP'16)*. IEEE, 1–5.
[79] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*. MIT Press, 3320–3328.
[80] Daniel Yska, Yi Mei, and Mengjie Zhang. 2018. Genetic programming hyper-heuristic with cooperative coevolution for dynamic flexible job shop scheduling. In *Proceedings of the European Conference of Genetic Programming (EuroGP'18)*. 306–321.
[81] Dong Yu and Li Deng. 2016. *Automatic Speech Recognition*. Springer.
[82] Dong Yu, Kaisheng Yao, Hang Su, Gang Li, and Frank Seide. 2013. KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'13)*. IEEE, 7893–7897.
[83] Chao Zhang, Zichao Yang, Xiaodong He, and Li Deng. 2019. Multimodal intelligence: Representation learning, information fusion, and applications. Retrieved from https://arXiv:1911.03977.
[84] Hangyu Zhu and Yaochu Jin. 2019. Multi-objective evolutionary federated learning. *IEEE Trans. Neural Netw. Learn. Syst.* 31, 4 (2019), 1310–1322.
[85] Yuze Zou, Shaohan Feng, Dusit Niyato, Yutao Jiao, Shimin Gong, and Wenqing Cheng. 2019. Mobile device training strategies in federated learning: An evolutionary game approach. In *Proceedings of the International Conference on Internet of Things (iThings'19) and IEEE Green Computing and Communications (GreenCom'19) and IEEE Cyber, Physical and Social Computing (CPSCom'19) and IEEE Smart Data (SmartData'19)*. IEEE, 874–879.

Received March 2020; revised November 2020; accepted January 2021