Federated Learning for Audio Semantic Communication

Haonan Tong, Zhaohui Yang, Sihua Wang, Ye Hu, Omid Semiari, Walid Saad, and Changchuan Yin

***Abstract*—In this paper, the problem of audio semantic communication over wireless networks is investigated. In the considered model, wireless edge devices transmit large-sized audio data to a server using semantic communication techniques. The techniques allow devices to only transmit audio semantic information that captures the contextual features of audio signals. To extract the semantic information from audio signals, a wave to vector (wav2vec) architecture based autoencoder is proposed, which consists of convolutional neural networks (CNNs). The proposed autoencoder enables high-accuracy audio transmission with small amounts of data. To further improve the accuracy of semantic information extraction, federated learning (FL) is implemented over multiple devices and a server. Simulation results show that the proposed algorithm can converge effectively and can reduce the mean squared error (MSE) of audio transmission by nearly 100 times, compared to a traditional coding scheme.**

***Index Terms*—Audio semantic communication, federated learning.**

I. Introduction

Future wireless networks require high data rate and massive connection for emerging applications such as the Internet of Things (IoT) [1]–[5]. In particular, in human-computer interaction scenarios, humans may simultaneously control multiple IoT devices using speech, thus making audio communication pervasive in wireless local area network such as smart home. However, due to bandwidth constrains, the wireless network in smart home may not be able to support a broad and prolonged wireless audio communication. This, in turn, motivates the development of semantic communication techniques that allow devices to only transmit semantic information. Semantic communication aims at minimizing the difference between the meanings of the transmitted messages and that of the recovered messages, rather than the recovered symbols. The advantage of such an approach is that semantic communication transmits less amounts of data than traditional communication techniques. However, despite recent interest in semantic communications [6]–[12], there is still a lack of reliable encoder and decoder models for audio semantic communication (ASC).

Existing works in [6]–[12] studied the important problems related to semantic communications. In [6], the authors pointed out that semantic communication should consider higher-level information such as content or semantic-related information rather than relying only on data-oriented metrics such as data rate or bit error probability. To efficiently transmit information, the work in [7] investigated a model-based approach for semantic data compression and showed that classical source and channel coding theorems have semantic counterparts. Furthermore, the authors in [8] proposed Bayesian game theory to design the transmission policies for transceivers and minimize the end-to-end average semantic metric while capturing the expected error between the meanings of intended and recovered messages. Besides, the authors in [9] proposed a semantic-aware network architecture to reduce the required communication bandwidth and significantly improve the communication efficiency. In [10], the authors defined a semantic based network system to reduce the data traffic and the energy consumption, hence increasing the wireless devices that can be supported. The work in [11] proposed a deep learning (DL) based text semantic communication system to reduce wireless traffic load. Meanwhile, in [12], the authors developed a new distributed text semantic communication system for IoT devices and they showed that nearly 20 times compression ratio can be achieved without any performance degradation. However, most of these existing works [6]–[12] that focused on the use of semantic communication for text data processing did not consider how to extract the meaning out of the audio data. Here, we note that audio data is completely different from text data since audio signals have a very high temporal resolution, at least 16,000 samples per second [13].

The prior art in [13]–[16] studied the problem of audio feature extraction. In [13], the authors adopted the so-called Mel-frequency cepstral coefficients (MFCC) features to represent the characteristics of audio signals. However, MFCC features are extracted only with a frequency domain, which lacks the contextual relation mining of audio sequence data. Recently, the works in [14]–[16] used DL based natural language processing (NLP) models to extract audio semantic features. In particular, the authors in [14] proposed a wave to vector (wav2vec) architecture to effectively extract semantic information. The authors in [15] proposed an end-to-end model that recognizes various language speeches. In [16], the authors proposed a speech generator which can generate speech audio signals with different styles using wave data. However, the works in [14]–[16] did not account for the impact of the channel noise on the transmitted data. Meanwhile, the work in [16] did not proposed any method to generate the audio

This work was supported by U.S. National Science Foundation (NSF) under Grants CNS-2007635 and CNS-2008646.

H. Tong, S. Wang, and C. Yin are with the Beijing Key Laboratory of Network System Architecture and Convergence, and also with the Beijing Advanced Information Network Laboratory, Beijing University of Posts and Telecommunications, Beijing, 100876 China, Emails: hntong@bupt.edu.cn, sihuawang@bupt.edu.cn, ccyin@bupt.edu.cn.

Z. Yang is with the Department of Electronic and Electrical Engineering, University College London, WC1E 6BT London, UK, Email: zhaohui.yang@ucl.ac.uk.

Y. Hu and W. Saad are with the Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, 24060, USA, Email: yeh17@vt.edu, walids@vt.edu.

O. Semiari is with the Department of Electrical and Computer Engineering, University of Colorado Colorado Springs, Colorado Springs, CO, 80918 USA. E-mail: osemiari@uccs.edu.

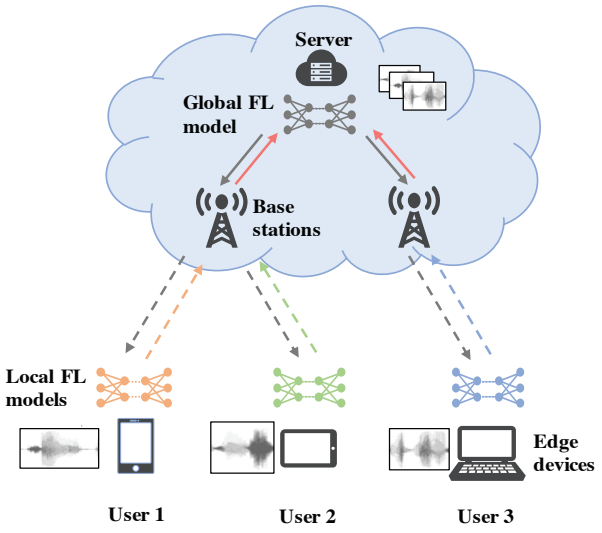


Fig. 1. The architecture of an FL based ASC system over wireless networks.

signals from the transmitted semantic information.

The use of federated learning (FL) in edge networks was studied in [17]–[27]. In [17], [18], the authors introduced FL method to generate a global model through collaboratively learning from multiple edge devices, thus learning a distributed algorithm without sharing datasets. The work in [19] proposed an FL framework in wireless networks and jointly considered wireless resource allocation and user selection while optimizing FL learning performance. To accelerate the convergence of FL, the authors in [20] proposed a probabilistic user selection scheme to enhance the efficiency of model aggregation, thus improving convergence speed and the FL training loss. Besides, the authors in [21] introduced over-the-air computation for fast global model aggregation which is realized using superposition property of a wireless multiple-access channel. To explore the applications of FL, the works in [22]–[25] provided comprehensive summaries on FL deployed on IoT devices. Besides, the work in [26] proposed an energy-efficient scheme to minimize the FL energy consumption and complete time, where closed-form solutions of wireless resource allocation are derived. In [27], the authors proposed efficient incentive mechanisms for FL to improve the learning security and accuracy, which used blockchain based reputation with contract theory. However, most of the above works [17]–[27] studied the prediction models which ignored the impact of FL on the performance of semantic communication.

The main contribution of this paper is a novel semantic communication model for audio communication, which is trained via federated learning (FL). Our key contributions include:

- We develop a realistic implementation of an ASC system in which wireless devices transmit large audio command data to a server. For the considered system, the bandwidth for audio data transmission is limited and, thus, semantic information is extracted and transmitted to overcome this limitation. To further improve the accuracy of semantic information extraction, the semantic extraction model must learn from multiple devices. Hence, FL is introduced to train the model with reducing the communication overhead of sharing training data. We formulate this audio communication problem as a signal recovery problem whose goal is to minimize the mean squared error (MSE) between the recovered audio signals and the source audio signals.
- To solve this problem, we propose a wav2vec based autoencoder that uses flexible CNNs to extract semantic information from source audio signals. The autoencoder consists of an encoder and a decoder. The encoder perceives and encodes temporal features of audio signals into semantic information, which is transmitted over an imperfect wireless channel with noise. Then, the decoder decodes the received semantic information and recovers the audio signals while alleviating channel noise. In this way, the proposed autoencoder transmits less data while jointly designing the source coding and channel coding in the autoencoder.
- To improve the accuracy of semantic information extraction, FL is implemented to collaboratively train the autoencoder over multiple devices and the server. In each FL training period, each local model is first trained with the audio data from the local device. Then, the parameters of the local models are transmitted to the server. Finally, the server aggregates the collected local models into a global model and broadcasts the global model to all the devices participated in the FL. Thus, the proposed autoencoder can integrate more audio features from multiple users and, hence, improve the accuracy of semantic information extraction.
- We perform fundamental analysis on the noise immunity and convergence of the proposed autoencoder. We theoretically show that the number of semantic features, time domain downsampling rate, and FL training method can significantly influence performance of the autoencoder.

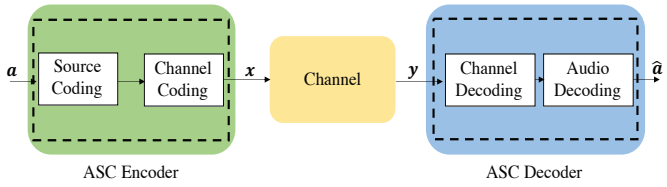


Fig. 2. The architecture of audio semantic communication (ASC).

Simulation results show that the proposed algorithm can effectively converge and reduce the MSE between the recovered and the source audio signals by nearly 100 times, compared to a traditional coding scheme. To our best knowledge, *this is the first work that studies the ASC model and uses FL to improve model performance, while avoiding the need for sharing training data.*

The rest of this paper is organized as follows. The system model and problem formulation are discussed in Section II. In Section III, we provide a detailed description of the proposed audio semantic encoder and decoder. The simulation results are presented and analyzed in Section IV. Finally, conclusions are drawn in Section V.

II. System Model and Problem Formulation

We consider a spectrum resource-limited uplink wireless network to deploy an ASC system, which consists of U edge devices, B base stations (BSs), and one server. Each edge device will transmit large audio packets to the server via the closest BS, as shown in Fig. 1. Due to the limited spectrum, audio semantic information must be extracted for data transmission, thus reducing communication overhead and improving the spectrum efficiency. In particular, edge devices must send audio semantic information via wireless channels to the BSs, and, then, the semantic information is delivered via optical links to the server for decoding. To *extract the audio semantic information* with high efficiency and accuracy, we assume that the edge devices and the server cooperatively train an ASC model *using FL*. The ASC model consists of an ASC encoder and an ASC decoder, as shown in Fig 2. In particular, the ASC encoder is deployed on each edge device to extract audio semantic information while the ASC decoder is deployed on the server to recover audio signals. The objective of the ASC model is to recover the audio signals as accurate as possible. We assume that the connections between BSs and the server use the optical links and have sufficient spectrum resource to support accurate transmission. We mainly consider the transmission impairments from the wireless channel between the edge devices and BSs. To enhance noise immunity, the ASC model must be trained using the received semantic information while taking into account the wireless channel impairments. Hence, the BSs are set to reliably send back the received semantic information to each device, which only occurs during the short-term training stage. Since the extraction of semantic information determines the accuracy of ASC, we consider the architecture design of the ASC model for audio communications.

A. ASC Encoder

The ASC encoder is used to encode the input audio data and to extract the semantic information from the raw audio data. We define $\boldsymbol{a} = [a_1, a_2, \ldots, a_T]$ as the raw audio data vector where each element a_t is the audio data in sample t with T being the number of samples. Let $\boldsymbol{x} = [x_1, x_2, \ldots, x_N]$ be the semantic information vector to be transmitted where x_n is element n in the vector. The ASC encoder extracts $\boldsymbol{x}$ from $\boldsymbol{a}$ by using a neural network (NN) model parameterized by $\boldsymbol{\theta}$, thus, the relationship between $\boldsymbol{a}$ and $\boldsymbol{x}$ can be given by:

$$\boldsymbol{x} = \boldsymbol{T}_{\boldsymbol{\theta}}(\boldsymbol{a}), \tag{1}$$

where $\boldsymbol{T}_{\boldsymbol{\theta}}(\cdot)$ indicates the function of the ASC encoder.

B. Wireless Channel

When transmitted over a wireless channel, semantic information will experience channel fading and noise. We assume that the audio transmission uses a single wireless link and, hence, the transmitted signal will be given by:

$$\boldsymbol{y} = h \cdot \boldsymbol{x} + \boldsymbol{\sigma}, \tag{2}$$

where $\boldsymbol{y}$ is the received semantic information at the decoder with transmission impairments, h is the channel coefficient, and $\boldsymbol{\sigma} \sim \mathcal{N}\left(0, \sigma^2 \boldsymbol{I}\right)$ is a Gaussian channel noise at the receiver with variance σ^2. $\boldsymbol{I}$ is the identity matrix.

C. ASC Decoder

The ASC decoder is used to recover the audio data $\boldsymbol{a}$ from the received semantic information $\boldsymbol{y}$ and to alleviate transmission impairments. The functions of the decoder and the encoder are generally reciprocal. Let $\widehat{\boldsymbol{a}}$ be the decoded audio data and $\boldsymbol{\varphi}$ be the parameters of the NN model in the ASC decoder. Then the relationship between $\widehat{\boldsymbol{a}}$ and $\boldsymbol{y}$ can be given by:

$$\widehat{\boldsymbol{a}} = \boldsymbol{R}_{\boldsymbol{\varphi}}(\boldsymbol{y}), \tag{3}$$

where $\boldsymbol{R}_{\boldsymbol{\varphi}}(\cdot)$ indicates the function of the ASC decoder.

D. ASC Objective

The objective of the ASC system is to recover the audio signals as accurate as possible. Since ASC system transmits semantic information, the use of bit error rate (BER) as a metric is not suitable to assess ASC. Hence, we use the mean squared error (MSE) to evaluate the quality of ASC at the semantic level. The ASC system objective function can be formulated to minimize the MSE between $\boldsymbol{a}$ and $\widehat{\boldsymbol{a}}$, as follows:

$$\min_{\boldsymbol{\theta},\boldsymbol{\varphi}} \ \mathcal{L}_{\text{MSE}}\left(\boldsymbol{\theta}, \boldsymbol{\varphi}, \boldsymbol{a}, \widehat{\boldsymbol{a}}\right) = \min_{\boldsymbol{\theta},\boldsymbol{\varphi}} \ \frac{1}{T}\sum_{t=1}^{T}\left(a_t - \widehat{a}_t\right)^2, \tag{4}$$

where $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ are the parameters of the ASC encoder and ASC decoder, respectively. Here, we assume that the architectures of $\boldsymbol{T}_{\boldsymbol{\theta}}$ and $\boldsymbol{R}_{\boldsymbol{\varphi}}$ are stay fixed and we only update the weights of NNs when solving problem (4). Hence, it is necessary to properly design the architecture of the ASC encoder and the ASC decoder. To this end, we introduce an autoencoder to extract audio semantic information.

III. Audio Semantic Encoder and Decoder

To solve problem (4), we first propose a wav2vec architecture based autoencoder to efficiently extract audio information. Then, to further improve the accuracy of semantic information extraction, the autoencoder is trained with FL over multiple devices and the server. Thus, the proposed autoencoder can learn semantic information extraction from the audio information of diverse users.

A. Wav2vec Architecture Based Autoencoder

In the proposed architecture, as shown in Fig. 2, the ASC system can be interpreted as an autoencoder [28]–[31], This autoencoder is trained to recover the input signals at the output end using compressed data features. Since the data must pass through each layer of the autoencoder, the autoencoder must find a robust representation of the input data at each layer [30]. In particular, NN models are used to build each layer in the autoencoder. Since convolutional neural networks (CNNs) are

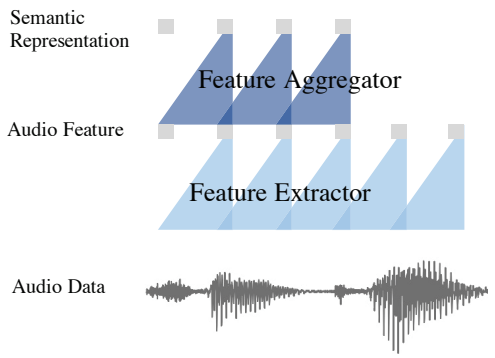


Fig. 3. The wav2vec architecture.

particularly good at extracting features and can be parallel deployed over time on multiple devices, we prefer to use CNNs instead of other NNs such as recurrent neural networks [32]–[34]. Next, we introduce our CNN-based wav2vec architecture for semantic information extraction.

To extract the semantic information, we use a wav2vec model as the audio semantic encoder. A simplification of our wav2vec architecture is shown in Fig. 3. From Fig. 3, we see that, the wav2vec architecture uses two cascaded CNNs, called feature extractor and feature aggregator [14], to extract audio semantic information. Given the raw audio vector, the extractor refines rough audio features and the aggregator combines the rough audio features into a higher-level latent variable that contains semantic relations among contextual audio features [14].

According to the wav2vec architecture, we design an audio semantic decoder, whose network architecture is symmetrical to the original wav2vec model [14]. Combining together an audio semantic encoder and the corresponding semantic decoder, we propose a wav2vec based autoencoder as shown in Fig. 4. In the autoencoder, the audio semantic encoder and the decoder extracts the semantic information and recovers audio signals from the semantic information, respectively. Each single encoder or decoder implements the function of joint source coding and channel coding. Considering the transmission impairments, the semantic information is designed to accurately capture the time domain contextual relations of the audio signals, so as to resist channel fading and noise interference.

Fig. 5 shows the NN layers of the proposed autoencoder. According to Fig. 5, we observe that, given the raw audio signals $\boldsymbol{a}$, the audio semantic encoder is used to extract the semantic vector $\boldsymbol{x}$. In the proposed audio semantic encoder, the data first passes through a feature extractor then a feature aggregator. The feature extractor and the aggregator consist of L_{ext} and L_{agg} convolution blocks, respectively. In particular, each convolution block consists of a) a convolution layer, b) a dropout layer, and c) a batch normalization layer, defined as follows:

- *Convolutional layer*: In CNNs, a convolutional layer is used to extract the spatial correlation of the input data with 1-D convolution between the input data $\boldsymbol{Z}^{l-1}$ and the kernel matrix. Mathematically, given the input $\boldsymbol{Z}^{l-1} \in \mathbb{R}^{\lambda_{l-1}\times M^{l-1}}$, the output of the convolutional layer l is $\boldsymbol{Z}^l = [\boldsymbol{z}^{l,1}, \cdots, \boldsymbol{z}^{l,m}, \cdots, \boldsymbol{z}^{l,M^l}], m = 1, \cdots, M^l$, where $\boldsymbol{z}^{l,m} \in \mathbb{R}^{\lambda_l \times 1}$ is the feature map m of convolutional layer l with M^l being the number of output features. Hence, the input $\boldsymbol{Z}^0$ of convolutional layer 1 is the raw audio data or the output of the last NN module. The output of feature map $\boldsymbol{z}^{l,m}$ in each convolutional layer l is given by:

$$\boldsymbol{z}^{l,m} = f\left(\sum_{k=1}^{M^{l-1}} \boldsymbol{z}^{l-1,k} \otimes \boldsymbol{W}_c^{l,m} + \boldsymbol{b}_c^{l,m}\right), \quad (5)$$

  where $f(x) = x$ is the linear activation function, M^{l-1} is the number of feature maps in the last convolutional layer $l-1$, $\otimes$ denotes 1-D convolution operation, and $\boldsymbol{W}_c^{l,m} \in \mathbb{R}^{s_k \times 1}$ and $\boldsymbol{b}_c^{l,m}$ are convolution kernels and bias vector of feature map m in convolutional layer l, respectively, with s_k being the kernel size. Let the convolution stride be s_c, the padding size be p, and the size of feature map λ_l satisfies $\lambda_l = \lfloor \frac{\lambda_{l-1}+2p-s_k}{s_c} \rfloor + 1$.
- *Dropout layer*: The input $\boldsymbol{Z}^l$ of a dropout layer l is the output of convolutional layer l. In the training stage, the dropout layer randomly abandons the effect of each neuron with a probability called dropout rate, and, in the inference stage, the dropout layer counts on the effects of all neurons. The dropout layer is used as a regularization approach to avoid overfitting problem.
- *Batch normalization layer*: A batch normalization layer normalizes the values of activated neurons to avoid gradient vanishing. We define α_i as the value of the activated neuron i in convolution block l The normalized value $\widehat{\alpha}$ of the neuron is given by $\widehat{\alpha}_i = \frac{\alpha_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$, where $\mu_{\mathcal{B}} = \frac{1}{\lambda_l}\sum_{i=1}^{\lambda_l} \alpha_i$, $\sigma_{\mathcal{B}}^2 = \frac{1}{\lambda_l}\sum_{i=1}^{\lambda_l} (\alpha_i - \mu_{\mathcal{B}})^2$, and ϵ is a positive constant.

Since the amplitude of an audio signal is limited, $\tanh(\cdot)$ is introduced as the activation function of the output layer in the feature extractor [16], where $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$. To shape the transmitted semantic information with an adequate amplitude, the last layer of the feature aggregator is set as batch normalization layer without activation function [31].

In the proposed audio semantic decoder, as shown in Fig 5, the received semantic information first passes through a feature decomposer then an audio generator. Different from the encoder, a deconvolution operation is introduced to build the feature decomposer and audio generator which consist of L_{de} and L_{gen} deconvolution blocks, respectively. Correspondingly, each deconvolution block consists of a) one deconvolution layer, b) one dropout layer, and c) one batch normalization layer. Mathematically, the processes of the dropout layer and the batch normalization layer are similar to those in the convolution blocks, except for the deconvolution layer.

In the deconvolution layer, the feature matrix is first uniformly filled with zeros in each column. Given the filled input matrix $\tilde{\boldsymbol{Z}}^{l-1} = [\tilde{\boldsymbol{z}}^{l-1,1}, \cdots, \tilde{\boldsymbol{z}}^{l-1,m}, \cdots, \tilde{\boldsymbol{z}}^{l-1,M^{l-1}}] \in \mathbb{R}^{\tilde{\lambda}_{l-1}\times M^{l-1}}$, the output of a deconvolution layer l is $\boldsymbol{Z}^l = [\boldsymbol{z}^{l,1}, \cdots, \boldsymbol{z}^{l,m}, \cdots, \boldsymbol{z}^{l,M^l}] \in \mathbb{R}^{\lambda_l \times M^l}, m = 1, \cdots, M^l$, where $\tilde{\boldsymbol{z}}^{l,m}$ is the filled feature map m and $\tilde{\lambda}_{l-1}$ is the filled

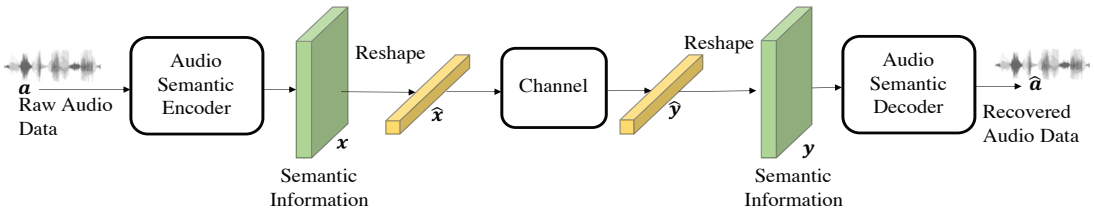


Fig. 4. Data shape in the proposed autoencoder over ASC system.

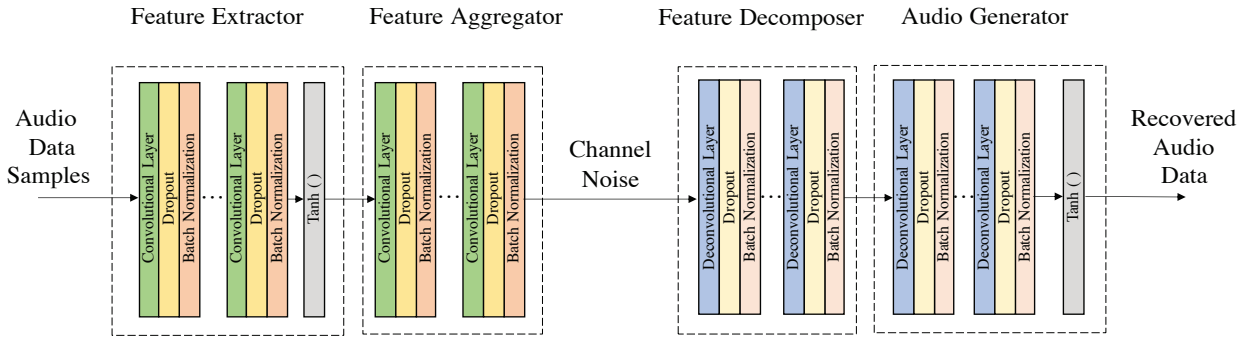


Fig. 5. The architecture of the proposed autoencoder.

feature map size. The output of feature map $\boldsymbol{z}^{l,m}$ in each convolutional layer l is given by:

$$\boldsymbol{z}^{l,m} = f\left(\sum_{k=1}^{M^{l-1}} \tilde{\boldsymbol{z}}^{l-1,k} \otimes \boldsymbol{W}_d^{l,m} + \boldsymbol{b}_d^{l,m}\right), \tag{6}$$

where M^{l-1} is the number of features of layer $l-1$, and $\boldsymbol{W}_d^{l,m}$ and $\boldsymbol{b}_d^{l,m}$ are deconvolution kernels and bias vector in deconvolutional layer l, respectively. In deconvolution layer, the filled feature map size $\tilde{\lambda}_{l-1}$ satisfies $\tilde{\lambda}_l = s_c(\lambda_{l-1}-1) - 2p + 2s_k - 1$ and the size of feature map λ_l satisfies $\lambda_l = \tilde{\lambda}_{l-1} - s_k + 1 = s_c(\lambda_{l-1}-1) - 2p + s_k$, where p is the padding size of layer l. Note that, to appropriately recover the audio signals, the output layer of the audio generator is set as $\tanh(\cdot)$ function.

To amplify the inference error and avoid gradient vanishing, we introduces the normalized root mean squared error (NRMSE) for the autoencoder. Then the objective of the autoencoder is given by:

$$\min_{\boldsymbol{\theta},\boldsymbol{\varphi}} \ \mathcal{L}_{\text{NRMSE}}\left(\boldsymbol{\theta},\boldsymbol{\varphi},\boldsymbol{a},\widehat{\boldsymbol{a}}\right) = \min_{\boldsymbol{\theta},\boldsymbol{\varphi}} \ \frac{\sum_{t=1}^{T}\left(a_t-\widehat{a}_t\right)^2}{\sum_{t=1}^{T} a_t^2}. \tag{7}$$

B. FL Training Method

Next, our goal is to minimize the errors between the recovered audio signals and the source audio signals using FL training method. In FL, the server and the devices collaboratively learn the proposed autoencoder by sharing the model parameters [19], [35]–[37]. We define $\boldsymbol{w} = (\boldsymbol{\theta},\boldsymbol{\varphi})$ as the total parameter of the proposed autoencoder, which includes both the encoder and decoder. The server generates a global model $\boldsymbol{w}^{\mathbf{g}}$ and each device i locally trains a local autoencoder model $\boldsymbol{w}_i$ which shares the same architecture as $\boldsymbol{w}^{\mathbf{g}}$, as shown in Fig. 1. The global model periodically aggregates local models from U devices that participate in FL and broadcasts the aggregated global model back to the devices. Then the aggregated global model can be given by $\boldsymbol{w}^{\mathbf{g}} = \frac{1}{U}\sum_{i=1}^{U}\boldsymbol{w}_i$. We use $\boldsymbol{A}_i$ to capture the audio dataset of local model i. According to problem (7), the objective of FL training method is given by:

$$\min_{\boldsymbol{w}^{\mathbf{g}}} \ \sum_{i=1}^{U} \mathcal{L}_{\text{NRMSE}}\left(\boldsymbol{w}_i, \boldsymbol{A}_i, \widehat{\boldsymbol{A}_i}\right). \tag{9}$$

During the local model training stage, the server first defines the architecture of the autoencoder and broadcasts it to all edge devices to randomly initialize the local models. To keep the coordination between the encoder and the decoder of the proposed autoencoder, we jointly set that the encoder and the decoder update the parameters simultaneously to minimize the loss function (9). Hence, both the encoder and decoder update the parameters with stochastic gradient descent (SGD) once after a batch of data passes through the autoencoder. The training process of each local model can be shown in Algorithm 1, where η in (8) is the learning rate.

Algorithm 1 Local model training algorithm of the autoencoder

1: **Initialize:** Randomly initialize parameters $\boldsymbol{\theta}^{(0)}$ and $\boldsymbol{\varphi}^{(0)}$, initialize training epoch $i = 0$.
2: **Input:** Batches of audio data $\boldsymbol{a}$
3: **while** model does not converge **or** $i <$ max epoch **do**:
4: $\quad \boldsymbol{T}_{\boldsymbol{\theta}}(\boldsymbol{a}) \rightarrow \boldsymbol{x}$.
5: $\quad$ Transmit $\boldsymbol{x}$ over wireless channel and receive $\boldsymbol{y}$.
6: $\quad \boldsymbol{R}_{\varphi}(\boldsymbol{y}) \rightarrow \widehat{\boldsymbol{a}}$.
7: $\quad$ Calculate $\mathcal{L}_{NRMSE}(\boldsymbol{\theta},\boldsymbol{\varphi},\boldsymbol{a},\widehat{\boldsymbol{a}})$ according to (9).
8: $\quad$ Update encoder's and decoder's parameters simultaneously with SGD:

$$\begin{aligned}\boldsymbol{\theta}^{(i+1)} &\leftarrow \boldsymbol{\theta}^{(i)} - \eta\nabla_{\boldsymbol{\theta}^{(i)}}\mathcal{L}_{\text{NRMSE}}\left(\boldsymbol{\theta}^{(i)},\boldsymbol{\varphi}^{(i)}\right)\\ \boldsymbol{\varphi}^{(i+1)} &\leftarrow \boldsymbol{\varphi}^{(i)} - \eta\nabla_{\boldsymbol{\varphi}^{(0)}}\mathcal{L}_{\text{NRMSE}}\left(\boldsymbol{\theta}^{(i)},\boldsymbol{\varphi}^{(i)}\right)\end{aligned} \tag{8}$$

9: $\quad i \leftarrow i+1$
10: **end while**
11: **Output:** $\boldsymbol{T}_{\boldsymbol{\theta}}(\cdot)$ and $\boldsymbol{R}_{\varphi}(\cdot)$ combined as $\boldsymbol{w}$.

Algorithm 2 FL training algorithm of the global model [22].

1: **Initialization:** Initialize the model architecture, the local models and the global model are initialized with random parameters. Initialize model aggregation step $\tau = 0$.
2: **while** model does not converge **or** $\tau <$ max aggregation step **do**:
3: Each local model updates $\boldsymbol{w}_1^{(\tau)}, \boldsymbol{w}_2^{(\tau)} ... \boldsymbol{w}_U^{(\tau)}$ through training from local dataset according to Algorithm 1.
4: Transmit local trained models $\boldsymbol{w}_1^{(\tau)}, \boldsymbol{w}_2^{(\tau)} ... \boldsymbol{w}_U^{(\tau)}$ to the server.
5: Update the global model: $\boldsymbol{w}^{\mathbf{g}(\tau)} = \frac{1}{U} \sum_{i=1}^{U} \boldsymbol{w}_i^{(\tau)}$
6: Dispatch local models: $\boldsymbol{w}_1^{(\tau+1)}, \boldsymbol{w}_2^{(\tau+1)} ... \boldsymbol{w}_U^{(\tau+1)} \leftarrow \boldsymbol{w}^{\mathbf{g}(\tau)}$
7: $\tau \leftarrow \tau + 1$
8: **end while**
9: **Output:** Global model $\boldsymbol{w}^{\mathbf{g}}$.

During the training process of the global model, each edge device is set to transmit the parameters of the local models $\boldsymbol{w}_i$ to the server every a fixed number of epochs. Thus, the server periodically collects the transmitted models, aggregates the parameters of the local models, and then broadcasts the updated global model to each device. In the next period, the local models update their parameters through training from local datasets $\boldsymbol{A}_i$, before transmitting $\boldsymbol{w}_i$ to the server, as shown in Algorithm 1. The FL algorithm for the global model is summarized in Algorithm 2.

C. Complexity Analysis

The proposed FL algorithm used to solve problem (9) is summarized in Algorithm 2. The complexity of the proposed algorithm lies in training the proposed autoencoder. The complexity for training the autoencoder is $\mathcal{O}\left(\sum_{l=1}^{L} \lambda_l^2 {s_k}^2 M^l M^{l-1}\right)$ [38], where $L = L_{\text{ext}} + L_{\text{agg}} + L_{\text{de}} + L_{\text{gen}}$, with $L_{\text{ext}}, L_{\text{agg}}, L_{\text{de}}, L_{\text{gen}}$, and L being the number of convolution or deconvolution layers in the feature extractor, the feature aggregator, the feature decomposer the audio generator, and the proposed autoencoder, respectively. Let L_o be the number of model aggregations until the FL global model converges. The complexity of the FL training method is $\mathcal{O}\left(L_o U \sum_{l=1}^{L} \lambda_l^2 {s_k}^2 M^l M^{l-1}\right)$ [20]. In consequence, the major complexity of training the autoencoder, which depends on the number of NN layers, the kernel sizes and the numbers of features in each layer, is linear. Meanwhile, since the layers in the autoencoder are finite, the local training is achievable and, hence the edge devices can support the FL training in the considered wireless network. Once the training process is completed, the trained autoencoder can be used for ASC in a long term period.

IV. Simulation and Performance Analysis

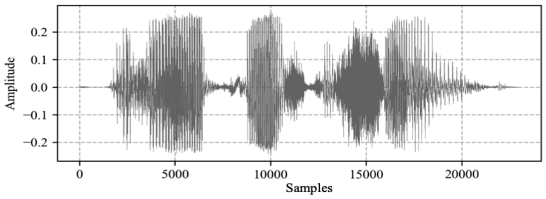
To evaluate the proposed autoencoder, we train the model using a training set from the speech dataset Librispeech [39], which contains 1000 hours of 16 kHz read English speech. The learning rate η is 10^{-5}. The proposed autoencoder is trained under additive white Gaussian noise (AWGN) channels with a fixed channel coefficient h and a 6dB signal-to-noise-ratio (SNR), and it is tested on 200000 samples of speech data. The simulation parameters are listed in Table I [40]. We train the model using FL method with 1 global model and 2 local models of user 1 and user 2, each local model is trained using read speech from a single person, and the FL models are tested with read speech of another user 3. The global model aggregates local models every 10 local training epochs.

TABLE I
Simulation Parameters.

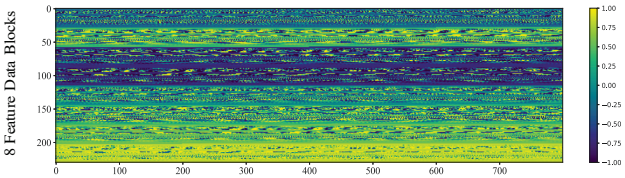
Module	Setting	Parameter	Value
feature extractor	$L_{\text{ext}} = 3$	feature M^l	8,8,8
		kernel size s_k	1,2,4
		stride s_c	1,1,1
		dropout rate	0.5
feature aggregator	$L_{\text{agg}} = 4$	feature M^l	8,8,8,8
		kernel size s_k	2,4,8,16
		stride s_c	1,1,1,1
		dropout rate	0.5
feature decomposer	$L_{\text{de}} = 4$	feature M^l	8,8,8,8
		kernel size s_k	2,4,8,16
		stride s_c	1,1,1,1
		dropout rate	0.5
audio generator	$L_{\text{gen}} = 4$	feature M^l	8,8,8,1
		kernel size s_k	1,2,4,1
		stride s_c	1,1,1,1
		dropout rate	0.5

For comparison purposes, we simulate a baseline scheme for high-quality audio transmission, which uses 128 kbps pulse code modulation (PCM) with 8 bits quantization levels [41] for source coding, low-density parity-check codes (LDPC) [42] for channel coding, and 64-QAM [43] for modulation. In this section, for notational convenience, we call the proposed autoencoder for ASC a "semantic method", and we call the baseline scheme a "traditional method". Note that, the autoencoder is trained via NRMSE, and tested via MSE. This is because that NRMSE induces larger gradient for training the autoencoder and MSE provides more obvious fluctuations for result comparison. To verify the performance of the proposed FL algorithm, we compare two baselines: transfer learning method and local gradient descent FL [22]. In the transfer learning method, the feature aggregator and the decomposer in the autoencoder are first initialized with a pre-trained model, then the autoencoder is trained using local audio data. In the local gradient descent FL, at the start of each iteration, all devices first share an aggregated model, then each device simultaneously computes a fixed number of local gradient descent updates (1000 steps) in parallel.

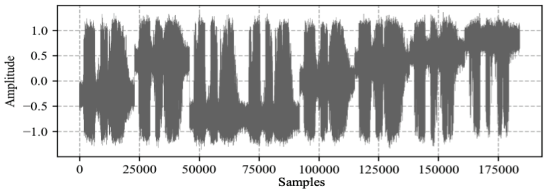
Fig. 6 shows examples of the raw audio data, the extracted semantic information reshaped in block form, the received semantic information, and the recovered audio data in one local model. From Fig. 6(a)-(c), we see that, the audio semantic information signals are amplified by the the proposed semantic encoder before being transmitted through the channel. From Fig. 6(b), it is also observed that, the extracted 8 different blocks of semantic features have correlations. From Fig. 6(c) and Fig. 6(d), we see that the proposed semantic decoder eliminates the channel noise from the received signals. The elimination of the noise is due to the fact that the semantic decoder relieves the noise using multiple semantic features. Fig. 6 shows that the proposed autoencoder can effectively guarantee the accuracy of ASC.



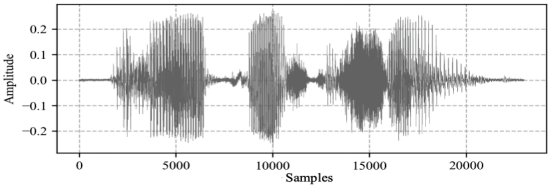
(a) Raw audio data $\boldsymbol{a}$ for transmission.



(b) Extracted semantic information $\boldsymbol{x}$, which is reshaped in block form. Each block represents one feature. A brighter pixel indicates a higher value of the semantic information.



(c) Received semantic information $\boldsymbol{y}$ by the decoder.



(d) Recovered audio data $\widehat{\boldsymbol{a}}$ by the decoder.

Fig. 6. Visualizations of a raw audio fragment, the corresponding semantic information that is reshaped in the block form, the received semantic information, and the recovered audio signals.

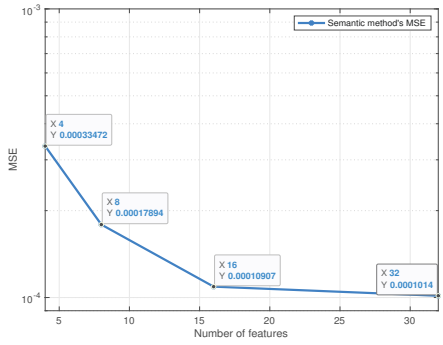


Fig. 7. Transmission MSE of a local autoencoder model as the number of features varies, in AWGN channels with a 6dB SNR.

Fig. 7 shows how transmission MSE of a local model using semantic method changes as the number of features varies. From Fig. 7, we see that, as the number of features increases, the MSE of the proposed semantic method decreases first and, then remains unchanged. This phenomenon is due to the fact that higher dimension features provide better semantic representations thus improving the transmission performance of the semantic method. From Fig. 7, we can also see that, when the number of features is larger than 16, the MSE of the semantic method tends to be leveling off. This result is because of the existence of redundant semantic features which provide limited noise immunity for ASC.

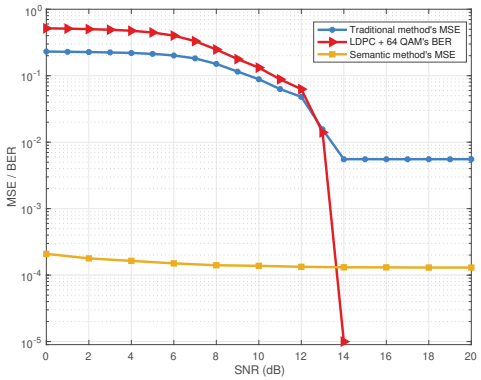


Fig. 8. Transmission MSE of a local model using semantic method, BER and transmission MSE of traditional method as SNR varies.

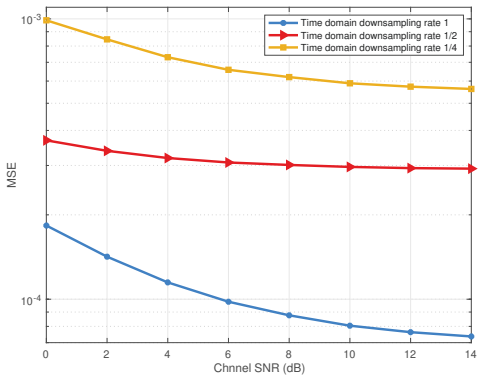


Fig. 9. Transmission MSE of the proposed semantic method with different time domain downsampling rates. The number of semantic features is 8.

In Fig. 8, we show how the transmission MSE of a local model using the proposed semantic method, BER and MSE of the traditional method change as the channel SNR varies. In this simulation, the semantic method reduces communication overhead by decreasing nearly 1/3 of the transmission data amount compared to the traditional method. From Fig. 8, we observe that, as the channel SNR increases, the error of communication decreases as expected. From Fig. 8, we can also see that our semantic method reduces the transmission MSE by nearly 100 times, compared to the traditional method, and the MSE of semantic method varies flatter than that of traditional method. The improvement is due to the fact that the semantic method has a better transmission accuracy and noise immunity performance. From Fig. 8, we can also see that, the MSE of the traditional method remains unchanged when the SNR is larger than 14 dB. The phenomenon is because, for a lower BER, the accuracy of the traditional coding scheme will reach the coding limit and, hence, the MSE will stay at a quantization error level caused by PCM quantization.

Fig. 9 shows how the transmission MSE changes versus various channel SNR, where the semantic method uses dif-

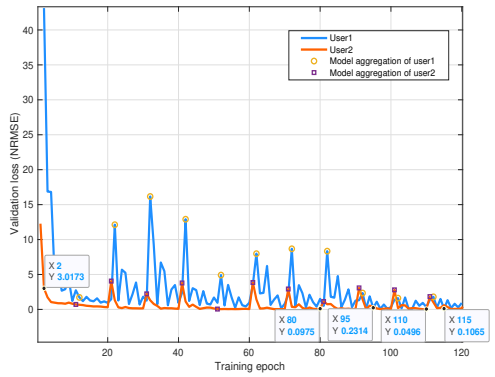


Fig. 10. Convergence results of the proposed FL models.

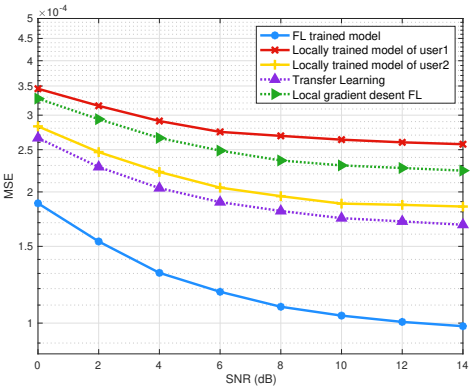


Fig. 11. Transmission MSE of FL trained model, locally trained models, transfer learning and local gradient descent FL.

ferent time domain downsampling rates. In this simulation, lower time domain downsampling rates can reduce transmission data amount exponentially and are realized by changing the convolution strides in the feature extractor and feature decomposer. From Fig. 9, we can see that, a lower time domain downsampling rate leads to more transmission error, which is because of the more loss of semantic information. From Fig. 9, we can also observe that as the SNR increases, the decreasing speed of the MSE differs among different downsampling rates. The disparity is due to the fact that, the semantic information extracted with different downsampling rates has diverse sensitivities to the SNR. Fig. 9 shows that reducing the time domain sampling rates decreases the communication accuracy. In consequence, Fig. 7 and Fig. 9 demonstrate that, in terms of improving the performance of semantic communication, the complexity of semantic features trades off the data compression rate.

In Fig. 10, we show how the validation loss changes as the training epoch increases. From Fig. 10, we observe that, the validation loss initially decreases with fluctuation first and then remains unchanged. The fact that the validation loss remains unchanged demonstrates that the FL algorithm converges. From Fig. 10, we can see that, when the FL global model is aggregated, the loss of local models increases for several epochs first and, then decreases in a long-term view. The result is due to the difference of the multiple local audio datasets from different users. At the beginning of training, the aggregation of multiple local models will critically change the parameter distribution of the global model. Then, as the training process continues, the global model parameters fit multiple local datasets. Hence the fluctuation caused by FL model aggregation weakens, and the local models of multiple users converge. From Fig. 10, we can also see that FL model aggregation further decreases the lower bound of loss in each local model. This phenomenon is because that FL training method aggregates audio semantic features from multiple users, thus enhancing model performance compared with local training method.

Fig. 11 shows how the transmission MSE of all algorithms changes as the channel SNR varies. From Fig. 11, we observe that the performance of the proposed model differs among the diverse users due to the various audio characteristics. We can also see that transfer learning can improve the model performance compared to locally training. Besides, local gradient descent FL outperforms part, but not all of the locally trained models. The difference of the baselines is because that transfer learning can further learn audio semantic extraction based on pre-trained model parameters. Whilst local gradient descent FL aggregates the global model with low frequency, where the difference among local models leads to the inefficiency on improving semantic extraction. From Fig. 11, we can also see that, the proposed FL algorithm outperforms the locally trained models. The superiority is because that the FL trained model aggregates audio characteristics of all users and hence obtaining more robust performance. We can also observe from the dotted lines that the proposed FL training method is superior over transfer learning and local gradient descent FL. The superiority is due to the fact that the proposed FL algorithm aggregates the model in a frequent and synchronous way, which guarantees a more accurate semantic extraction than that of the baselines.

V. Conclusion

In this paper, we have developed an FL trained model over an ASC architecture in the wireless network. We have considered avoidance of training data sharing and heavy communication overhead of the large-sized audio transmission between edge devices and the server. To solve this problem, we have proposed a wav2vec based autoencoder to effectively encode, transmit, and decode audio semantic information, rather than traditional bits or symbols, to reduce communication overhead. Then, the autoencoder is trained with FL to improve the accuracy of semantic information extraction. Simulation results have shown that the proposed algorithm can converge effectively and yields significant reduction on transmission error compared to existing coding scheme which uses PCM, LDPC and 64-QAM algorithm.

References

[1] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Network*, vol. 34, no. 3, pp. 134–142, May 2020.

[2] S. K. Lee, M. Bae, and H. Kim, "Future of IoT networks: A survey," *Applied Sciences*, vol. 7, no. 10, Oct. 2017.

[3] Y. Hu, M. Chen, W. Saad, H. V. Poor, and S. Cui, "Distributed multi-agent meta learning for trajectory design in wireless drone networks," *IEEE Journal on Selected Areas in Communications*, to appear, 2021.

[4] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for internet of things (IoT) security," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 3, pp. 1646–1685, 3rd Quart 2020.

[5] C. Huang, Z. Yang, G. C. Alexandropoulos, K. Xiong, L. Wei, C. Yuen, Z. Zhang, and M. Debbah, "Multi-hop RIS-empowered terahertz communications: A DRL-based hybrid beamforming design," *IEEE Journal on Selected Areas in Communications*, to appear, 2021.

[6] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, Oct. 1948.

[7] J. Bao, P. Basu, M. Dean, C. Partridge, A. Swami, W. Leland, and J. A. Hendler, "Towards a theory of semantic communication," in *Proc. IEEE Network Science Workshop*, West Point, NY, USA, Jun. 2011, pp. 110–117.

[8] B. Guler, A. Yener, and A. Swami, "The semantic communication game," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 787–802, Dec. 2018.

[9] G. Shi, Y. Xiao, Y. Li, and X. Xie, "From semantic communication to semantic-aware networking: Model, architecture, and open problems," *arXiv:2012.15405*, 2020. [Online]. Available: http://arxiv.org/abs/2012.15405

[10] E. Uysal, O. Kaya, A. Ephremides, J. Gross, M. Codreanu, P. Popovski, M. Assaad, G. Liva, A. Munari, T. Soleymani, B. Soret, and K. H. Johansson, "Semantic communications in networked systems," *arXiv:2103.05391*, 2021. [Online]. Available: http://arxiv.org/abs/2103.05391

[11] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," *arXiv:2006.10685*, 2020. [Online]. Available: http://arxiv.org/abs/2006.10685

[12] H. Xie and Z. Qin, "A lite distributed semantic communication system for internet of things," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 142–153, Jan. 2021.

[13] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. NJ, USA, Prentice-Hall, Inc., 2009.

[14] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "Wav2vec: Unsupervised pre-training for speech recognition," *arXiv:1904.05862*, 2019. [Online]. Available: http://arxiv.org/abs/1904.05862

[15] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *Proc. International Conference on Machine Learning (ICML)*, NY, USA, Jun. 2016, pp. 173–182.

[16] A. V. D. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv:1609.03499*, 2016. [Online]. Available: http://arxiv.org/abs/1609.03499

[17] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecný, S. Mazzocchi, H. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards federated learning at scale: System design," *ArXiv*, vol. abs/1902.01046, 2019.

[18] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE Conference on Computer Communications*, Paris, France, Apr. 2019.

[19] 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*, vol. 20, no. 1, pp. 269–283, Jan. 2021.

[20] M. Chen, H. Vincent Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2457–2471, Dec. 2020.

[21] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.

[22] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, "A survey on federated learning for resource-constrained IoT devices," *IEEE Internet of Things Journal*, to appear, 2021.

[23] 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, May 2020.

[24] A. Imteaj and M. H. Amini, "Distributed sensing using smart end-user devices: Pathway to federated learning for autonomous IoT," in *Proc. International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, USA, Apr. 2019.

[25] M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, and H. V. Poor, "Distributed learning in wireless networks: Recent progress and future challenges," *arXiv:2104.02151*, 2021. [Online]. Available: http://arxiv.org/abs/2104.02151

[26] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Transactions on Wireless Communications*, to appear, 2020.

[27] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 700–10 714, Dec. 2019.

[28] T. O'Shea and J. Hoydis, "An introduction to machine learning communications systems," *ArXiv:1702.00832*, 2017. [Online]. Available: http://arxiv.org/abs/1702.00832

[29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[30] Y. Lu, P. Cheng, Z. Chen, Y. Li, W. H. Mow, and B. Vucetic, "Deep autoencoder learning for relay-assisted cooperative communication systems," *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5471–5488, Sept. 2020.

[31] S. Drner, S. Cammerer, J. Hoydis, and S. T. Brink, "Deep learning based communication over the air," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 132–143, Feb. 2018.

[32] A. Shewalkar, "Performance evaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 9, no. 4, pp. 235–245, Mar. 2019.

[33] T. Hori, J. Cho, and S. Watanabe, "End-to-end speech recognition with word-based RNN language models," in *Proc. IEEE Spoken Language Technology Workshop (SLT)*, Athens, Greece, Dec. 2018.

[34] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, BC, Canada, May 2013.

[35] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar, and S. Cui, "Communication-efficient federated learning," *Proceedings of the National Academy of Sciences*, vol. 118, no. 17, Apr. 2021.

[36] Z. Yang, M. Chen, K.-K. Wong, H. V. Poor, and S. Cui, "Federated learning for 6G: Applications, challenges, and opportunities," *arXiv:2101.01338*, 2021. [Online]. Available: http://arxiv.org/abs/2101.01338

[37] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang, and D. Niyato, "Federated learning for 6G communications: Challenges, methods, and future directions," *China Communications*, vol. 17, no. 9, pp. 105–118, Sept. 2020.

[38] Y. Wang, M. Chen, Z. Yang, T. Luo, and W. Saad, "Deep learning for optimal deployment of uavs with visible light communications," *IEEE Transactions on Wireless Communications*, vol. 19, no. 11, pp. 7049–7063, Nov. 2020.

[39] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Queensland, Australia, Apr. 2015, pp. 5206–5210.

[40] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable federated learning for mobile networks," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 72–80, Feb. 2020.

[41] K. Nakano, H. Moriwaki, T. Takahashi, K. Akagiri, and M. Morio, "A new 8-bit pcm audio recording technique using an extension of the video track," *IEEE Transactions on Consumer Electronics*, vol. CE-28, no. 3, pp. 241–249, Aug. 1982.

[42] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[43] T. Pfau, S. Hoffmann, and R. Noe, "Hardware-efficient coherent digital receiver concept with feedforward carrier recovery for M-QAM constellations," *Journal of Lightwave Technology*, vol. 27, no. 8, pp. 989–999, Apr. 2009.