Deep Over-the-Air Computation

Hao Ye, Geoffrey Ye Li, Biing-Hwang Fred Juang

School of Electrical and Computer Engineering
Georgia Institute of Technology
Email: yehao@gatech.edu; liye@ece.gatech.edu; juang@ece.gatech.edu

GLOBECOM 2020 - 2020 IEEE Global Communications Conference | 978-1-7281-8298-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/GLOBECOM42002.2020.9321975

***Abstract*—As an efficient data fusion method, over-the-air computation integrates computation and communication by exploiting the superposition property of multiple access channels. In this paper, a framework on deep learning enabled over-the-air computation is proposed, where both the pre-processing and post-processing functions are represented by deep neural networks (DNNs). In this way, the over-the-air computation can approximate any function via learning through the data. The deep over-the-air framework is useful to a variety of machine learning applications on the Internet-of-Things (IoT). The experiments on distribution regression and anomaly detection have shown the effectiveness of the proposed method.**

I. Introduction

The future Internet-of-Things (IoT) network is expected to connect an enormous number of sensors and edge devices, generate huge amounts of data at the edge of the network, and support a wide range of machine learning and/or artificial intelligence based applications [1]. On the other hand, unprecedented challenges have also been brought in order to manage and analyze the huge volumes of highly distributive data. Due to the constraints on the latency, bandwidth, and privacy, the conventional 'aggregate-then-compute' approach becomes impractical. In order to tackle these issues, the over-the-air computation has been developed as an efficient data fusion method, where the superposition property of wireless channels is leveraged to allow multiple devices to transmit simultaneously [2]. However, the existing over-the-air computations only focus on limited predefined functions, such as the weighted summation, min and max function, and geometric mean, which are far from satisfactory for advanced machine learning applications.

In the last decade, deep learning has achieved remarkable success in a wide range of applications, including computer vision, speech recognition, and natural language processing. With the hierarchical structure of stacking nonlinear layers, deep neural networks (DNNs) can efficiently represent the data with hierarchical features and approximate complicate functions. Building on the huge volumes of data in the IoT, DNNs will be able to perform complicate sensing and recognition tasks, providing new ways of interactions between humans and physical environments. However, one critical challenge to apply deep learning for data analysis in IoT is that the dimension of the input is extremely high and varies with the number of edge devices. Recently, permute-invariant DNNs have been developed to approximate functions over collections of elements, where the order of the elements does not affect the value of the functions. To enforce permutation invariance, sum-decomposition operation on the latent space is designed and the learning structure can be expressed as

$$y = \rho\left(\sum_{x\in\mathcal{X}} \phi(x)\right), \tag{1}$$

where x represents each element in the collection $\mathcal{X}$ and $\rho(\cdot)$ and $\phi(\cdot)$ are processing functions represented by DNNs [3]. One of the important applications of (1) is parameter aggregation in federated learning [4].

Besides achieving the order invariance, the sum-decomposition operation makes the DNNs suitable for the IoT systems via the over-the-air computation. In this article, a deep learning based over-the-air computation framework is developed upon the sum-decomposition operation over the latent space. The superposition property of the multi-access wireless channels is leveraged to compute the summation of the latent features. ϕ and ρ are implemented in the edge devices and the access point (AP) as the pre-processing and post-processing functions, respectively. The effects of the wireless channels on the deep learning based over-the-air computation are investigated, including the channel noise and fadings. In addition, the deep over-the-air computation framework can be further extended to a decentralized setting, where there is no central AP to collect the information and each edge device needs to communicate with others based on the over-the-air computation to share information. Two types of application examples are considered, i.e., distribution regression and anomaly detection, which show that the deep over-the-air computation can achieve outstanding performance and save communication resources.

The main contributions of this article are to develop a deep learning based over-the-air computation framework for the intelligent machine learning applications on the IoT systems with the centralized and decentralized structures.

II. Related Works

The proposed approach is related to several topics in wireless communications and machine learning, including over-

This work was supported by the National Science Foundation under Grants 1815637 and 1731017.

the-air computation, permutation-invariant DNN, and learning based end-to-end communication systems.

Over-the-air Computation: The idea of over-the-air computation has been first proposed in the seminal work [2], which shows that the interference of the channel can be exploited for functional computation via structure codes. And it has been further demonstrated that this simple analog transmission without coding can achieve the minimum functional distortion in special cases [5]. In addition, practical issues in over-the-air computation have been addressed, including power control [7], synchronization errors [8], and beamforming in the MIMO systems [9]. Recently, the over-the-air computation has been utilized in the federated learning in the IoT system, with which the gradients computed at edge devices can be aggregated efficiently for model training at the cloud server [11], [12].

There have been several prior theoretical works showing that the expression ability of the functions of over-the-air computation, i.e., the nomographic functions, is powerful enough to approximate any continuous function with proper pre-processing and post-processing functions [10]. Nevertheless, the previous work focused on limited predefined functions while the deep over-the-air framework can approximate any unknown functions via learning through the data.

Permutation Invariance Learning: Permutation invariance learning aims to develop machine learning algorithms with a set of samples as input, where the order of the samples does not affect the output. In order to enforce the permutation invariance, the sum decomposition over the latent space has been proposed [3], which shares the same structure with nomographic functions. In fact, the sum-decomposition has already been employed by many machine learning algorithms in order to deal with permutation invariant inputs with the variable size. The attention model performs a weighted summation of a group of features [13]. In a graph neural network, each node updates its hidden state by a weighted sum of the states of their neighborhood [14]. The PointNet, a 3D point cloud classifier, obtains the global feature by computing a weighted sum of all point features[15]. In this paper, the sum decomposition over the latent space is adopted so that it can be easily computed by leveraging the superposition of the multi-access channel.

End-to-End Communications: The proposed deep over-the-air computation can be also seen as an extension of the learning based end-to-end communication systems since the transmitter and the receiver are represented by DNNs in both systems. The difference is that the previous end-to-end communication systems focused on the point-to-point communication while there are multiple transmitters in the over-the-air computation framework. The end-to-end communication has been first proposed in [17]. Subsequently, it has been extended to the orthogonal frequency-division multiplexing (OFDM) system [18] and multiple-input multiple-output (MIMO) system [6]. Recently, end-to-end communication systems without the channel model has attracted much attention. Several model-free end-to-end learning methods have been proposed based on approaches, such as reinforcement learning [19] and generative adversarial net (GAN) [20], [21].

III. Methodology

With the DNNs as pre-processing and post-processing functions, the deep over-the-air computation can learn to approximate any target function in a data-driven manner. In this section, the deep over-the-air computation is presented in detail, including the concept of over-the-air computation, the architectures of the deep learning enabled over-the-air computation, and the training algorithms.

A. Over-the-Air Computation for Aggregation

We consider an IoT system consisting of K edge devices, each having an l-dimensional signal $\mathbf{s}_k \in \mathbb{R}^l$. Instead of transmitting $\mathbf{s}_k$ separately and aggregating in the central AP, over-the-air computing allows the edge devices to send their data simultaneously. If an ideal multi-access channel is considered, the received signal at the AP will be

$$\mathbf{y} = \sum_{k=1}^{K} \mathbf{s}_k. \tag{2}$$

Therefore, for computing the sum, the results can be obtained with the received signal directly without knowing any local data $\mathbf{s}_k$ at the edge devices. In this way, the communication resources can be saved up to a factor of $\mathcal{O}(K)$.

With proper pre-processing and post-processing functions, over-the-air computation can be extended to non-linear functions. In general, a certain class of functions, called nomographic functions, have the structure that can be calculated easily via over-the-air computation.

Definition (Nomographic Functions [10]). Let $\mathcal{A}^k$, $k > 2$, be a metric space. Then every function $f \in \mathcal{F}[\mathcal{A}^k]$ with a representation

$$f(x_1, ..., x_K) = \rho(\sum_{k=1}^{K} \phi(x_k)), \tag{3}$$

is called nomographic function, where $\phi(\cdot)$ and $\rho(\cdot)$ are the pre-processing and post-processing function, respectively. It has been analytically shown that the nomographic functions can be used to approximate any continuous function [10].

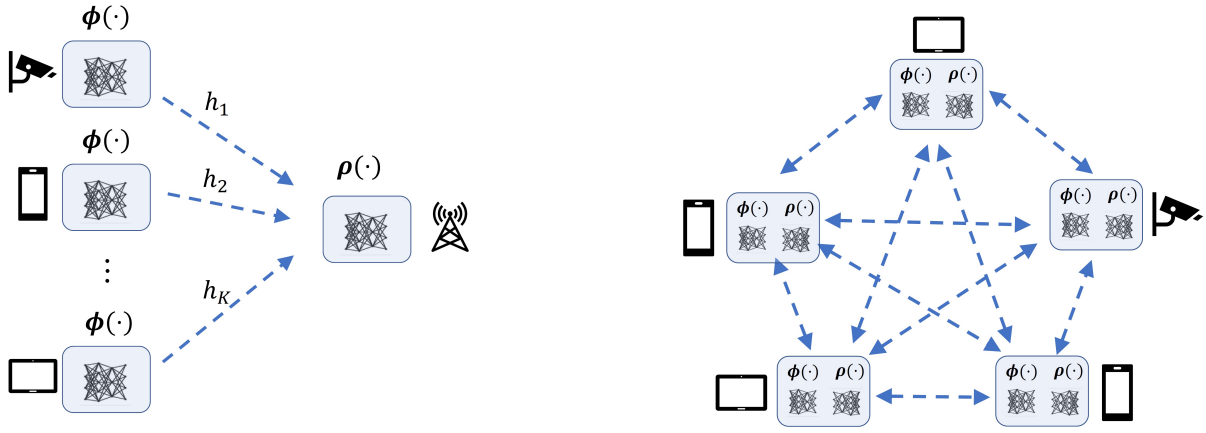
When considering a real wireless channel with additive noise and fading. The output of over-the-air computation at the AP can be expressed as

$$\mathbf{y} = \rho(\sum_{k=1}^{K} h_k \cdot \phi(\mathbf{s}_k) + \mathbf{n}), \tag{4}$$

where h_k is the channel coefficient and $\mathbf{n}$ is the received additive noise. It is obvious that $h_k = 1$ and $\mathbf{n} = 0$ for the ideal multi-access channel shown in (2)

B. Architectures

The structure of the deep over-the-air framework is shown in Fig. 1(a), where DNNs are used at each edge device and the AP as the pre-processing and post-processing functions, respectively. In each device, the pre-processing DNN takes the local data $\mathbf{s}_k$ as the input and outputs an embedding vectors



(a) Centralized deep over-the-air computation

(b) Decentralized deep over-the-air computation

Fig. 1: Structure of deep over-the-air computation.

which will be sent to the multi-access channel. The weights of the pre-processing DNN are shared across the devices. In addition, the over-the-air computation heavily relies on the channel state information (CSI). We assume that each edge device has the CSI of the link to the AP while the AP does not have access to the CSI, which is considered as a part of the input of the pre-processing DNN in order to be adaptive to the fading of the wireless channel. Therefore, the deep over-the-air computation can be expressed as

$$\mathbf{y} = \rho_{\mathbf{w}_\rho}(\sum_{k=1}^{K} h_k \cdot \phi_{\mathbf{w}_\phi}(\mathbf{s}_k, h_k) + \mathbf{n}), \tag{5}$$

while $\phi_{\mathbf{w}_\phi}(\cdot)$ and $\rho_{\mathbf{w}_\rho}(\cdot)$ represent the pre-processing and post-processing DNNs with weights $\mathbf{w}_\phi$ and $\mathbf{w}_\rho$, respectively.

Besides the centralized architecture, we may also consider a fully decentralized deep over-the-air computation framework shown in Fig. 1(b). In this case, there is no central AP to collect the data and each device communicates via the multi-access channel to share the information. In each device, both the DNNs for pre-processing and post-processing are equipped. The cost for channel estimation is heavy in this case because of the huge number of links.

Without the CSI, the decentralized over-the-air computation framework works as follows. With the pre-processing DNN, the embedding features for the local data are extracted at each device and sent to the multi-access channel. If full-duplex transmission scheme is assumed, each device can receive a signal from all other devices. Otherwise, each device can only get information from devices that do not collide with itself. With its local embedding features and received signals from other devices, the post-processing DNN can get the desired output.

C. Training

With DNNs representing the pre-processing and post-processing functions of over-the-air computation, the system can be optimized to approximate any function $f(\mathbf{s}_1, \cdots, \mathbf{s}_K)$ via minimizing an empirical loss on samples of the target function $\{\mathbf{s}_{1:K}, f(\mathbf{s}_{1:K})\}$, even without knowing the expression for $f(\cdot)$.

A loss function is chosen to measure the distance of the post-processing DNN output and the desired function output. The stochastic gradient descent (SGD) algorithm is used to minimize the empirical loss. The gradient for the over-the-air computation can be expressed as

$$\begin{aligned}&\partial_{\mathbf{w}_\phi} \rho_{\mathbf{w}_\rho}(\sum_{k=1}^{K} h_k \cdot \phi_{\mathbf{w}_\phi}(\mathbf{s}_k) + \mathbf{n}) \\ &= \rho'_{\mathbf{w}_\rho}(\sum_{k=1}^{K} h_k \cdot \phi_{\mathbf{w}_\phi}(\mathbf{s}_k) + \mathbf{n}) \cdot \sum_{k'=1}^{K} h_{k'} \cdot \partial_{w_\phi} \phi_{\mathbf{w}_\phi}(\mathbf{s}_{k'}).\end{aligned} \tag{6}$$

The training set consists of a local dataset where the local data $\{\mathbf{s}_{1:K}\}$ at the edge devices are collected, and a channel set, where the realization of the wireless channels $\{h_{1:K}\}$ are collected. With the two datasets, the training can be conducted to minimize the loss function.

IV. Applications

The deep over-the-air computation framework can accommodate a variety of machine learning based applications for the IoT system, where the data lies on the edge devices distributively and the edge devices communicate with the AP or other devices via the multi-access channel. Since only aggregated information can be obtained at the receiver, the privacy can be preserved while saving the communication resources. In this section, two types of applications are shown as examples, i.e., the distribution regression and the anomaly detection.

A. Distribution Regression

Unlike the typical machine learning problems, where the predictions are made for each instance, distribution regression is a problem of learning regression functions from a group of samples to a single set level label [16]. Specifically, the

training data is in the form of $\{(\{\mathbf{x}_{i,n}\}_{n=1}^{N_i}, y_i)\}_{i=1}^{D}$, where the elements in the ith sample, $\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \cdots \mathbf{x}_{i,N_i}$, are i.i.d. from a distribution p_i, and y_i is the label for p_i. The objective is to learn to predict a new label y_{D+1} for a new batch of samples $\{\mathbf{x}_{D+1,n}\}_{n=1}^{N_{D+1}}$ drawn from an unknown distribution. The concrete examples of distribution regression include approximating real-valued functions of distributions such as entropy or mutual information, where the training data can be acquired through a reliable but computationally intensive Monte Carlo approach. Also, we may wish to take a set of images as input and obtain useful information, such as the number of pedestrians crossing a street.

For the IoT system, we assume that $\mathbf{x}_{i,n}$ is distributively allocated at edge devices and the predictions are made at a central AP without acquiring the samples from the edge devices. With the deep over-the-air computation framework, each device encodes its local sample $\mathbf{x}_{i,n}$ by the pre-processing network and sends the output feature vector to the AP simultaneously. The prediction can be made at the AP by the post-processing DNN after receiving the combined features from the multi-access channel. During the training, l_1 or l_2 loss is used for optimizing the parameters of the pre-processing and post-processing DNNs so that the empirical loss on the training set is minimized.

B. Anomaly Detection

Besides making set-level predictions over a group of samples, the decentralized deep over-the-air computation framework can be used for anomaly detection over the IoT system. The objective is to find the rare samples or outliers from a sample set $\mathbf{s}_{1:K}$. We assume that the the samples are distributed across the edge devices and decisions are made for each device without sharing the local sample $\mathbf{s}_i$. In addition, most of samples are assumed to belong to one distribution while there may have several outliers belonging to a different distribution.

With the decentralized deep over-the-air framework, each device can communicate with other devices via the multi-access channel. The embedding features for the local data $\mathbf{s}_i$ on each device are first obtained via the pre-processing DNN and shared with other devices via the over-the-air computation. Then each device can obtain a combination of features from other devices although the combination coefficients are unknown. With the local embedding feature and the received combination of features, the post-processing DNN can determine whether the local sample is consistent with most of the samples. This problem can be cast as a binary classification problem, where the binary cross-entropy loss is used during the training.

V. Experiments

In this section, we evaluate the performance of deep over-the-air computation on distribution regression and anomaly detection. Three experiments have been conducted to illustrate the effectiveness of the deep over-the-air computation framework and the wireless channel effects are considered and quantized.

A. Estimating the Population Statistics

Experiment Settings: In this experiment, the deep over-the-air framework is used to estimate the entropy of Gaussian distributions without prior information about the Gaussianity. The Gaussian distributions are generated in the following way. We first randomly generate a 2×2 Gaussian distribution p_0 with covariance matrix $\mathbf{\Sigma} = \mathbf{C}\mathbf{C}^T$, where each element of $\mathbf{C} \in \mathbf{R}^{2\times 2}$ is random drawn from $\mathcal{N}(0,1)$. We then generate 1,000 Gaussian distributions with covariance matrix $\mathbf{M}_i$ defined by rotating $\mathbf{\Sigma}$ with a 2d rotation matrix $\mathbf{R}(\alpha_i) = \begin{bmatrix} \cos\alpha_i, & -\sin\alpha_i \\ \sin\alpha_i, & \cos\alpha_i \end{bmatrix}$, where $\alpha_i = \frac{i\pi}{1000}, i \in \{1, 2, \cdots, 1,000\}$. The covariance matrix $\mathbf{M}_i$ of the rotated distributions is $\mathbf{M}_i = \mathbf{R}(\alpha_i)\mathbf{\Sigma}\mathbf{R}^T(\alpha_i)$. Our goal is to estimate the entropy of the first marginal distribution from samples, which can be analytically expressed as $H_i = \frac{1}{2}\ln(2\pi e\mathbf{M}_i(1,1))$. The training and test sets are obtained via sampling 200 samples from each generated Gaussian distribution and the samples are assumed to be distributed across 200 edge devices in a IoT system, each contains only one sample. The Rayleigh fading channels are assumed between the edge devices and the AP.

The deep over-the-air computation model is trained using l_1 loss and three fully connected layers with Relu activation functions are utilized for both pre-processing and post-processing DNN. The input consists of the local Gaussian samples and the local CSI. A l_2 normalization layer is added at the end of the pre-processing DNN to control the transmission power. The numbers of hidden neurons are 256, 128, 30 in the pre-processing DNN while the numbers are 256, 128, 1 in the post-processing DNN. Adam is used as the optimizer and the batch size is 120.

Baseline: We compare the deep over-the-air computation framework with a baseline system, where each device also encodes local data $\mathbf{s}_k$ with a DNN. Instead of using over-the-air computation for transmission, the edge devices communicate with the AP system at different frequency bands and send the embedding features without interference. With the sequentially received embedding features, a long short-term memory (LSTM) network is employed to predict the group label. Therefore, the LSTM approach requires the bandwidth K times larger than the deep over-the-air computation framework. The pre-processing DNN remains the same as the deep over-the-air computation framework and the post-processing DNN at the AP consists of an LSTM layer with 256 neurons followed by two fully connected layer with 128 and 1 neurons.

Results: Fig. 2 shows the mean-squared error (MSE) of both the deep over-the-air computation and the baseline system at different signal-to-noise ratios (SNRs). The proposed approach achieves better performance than the LSTM approach while saving communication bandwidth.

B. Digit Sum

Experiment Settings: We next predict the sum of digit images distributed in edge devices. The experiment setting

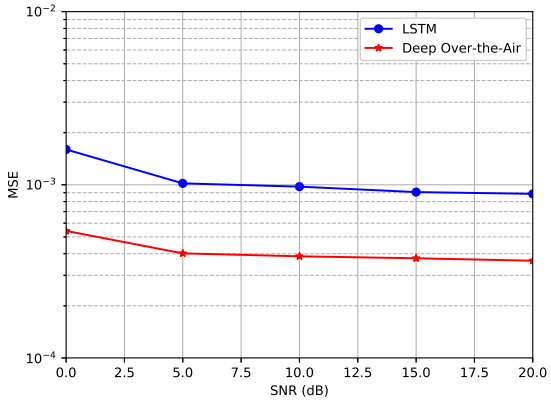


Fig. 2: Entropy estimation.

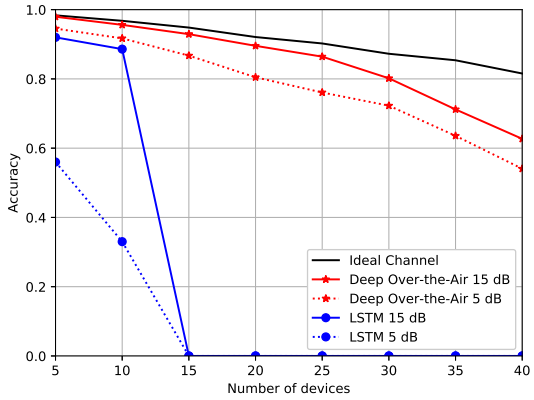


Fig. 3: Digit sum estimation.

follows [3] except that the IoT scenario and the wireless fading channels are introduced. In particular, mnist8m dataset [22] is used for evaluation, which contains 8 million samples of 28×28 grey-scale images of digits in $\{0, ..., 9\}$. We randomly select image sets from this dataset to build the training sets, where each set contains maximum 10 images and the set-label is the sum of digits in that set. Similar to the previous experiment, images in each set are assumed to be distributed across the edge devices and Rayleigh channels between the edge devices and the AP are assumed.

The deep over-the-air computation model is trained using l_1 loss and fully connected layers with Relu activation for both pre-processing and post-processing DNN. In the pre-processing DNN, the image is processed by three fully connected layers with 300, 100, 30 hidden neurons. The output is then concatenated with the CSI and followed by 3 1D convolutional layers with 256, 128 and 2 filters. In the post-processing DNN, the input is connected to a fully connected layer with one neuron. Adam is used as the optimizer and the batch size is 120.

Baseline: Similar to the previous experiment, the deep over-the-air computation framework is compared with a baseline system, where the independent communications and LSTM are adopted. The structure of pre-processing DNN used at each device is identical to the deep over-the-air computation approach and the structure of the DNN used at AP contains an LSTM layer with 100 neurons, followed by one fully connected layer with one neuron. Besides, we also compare with the performance with ideal multi-access channel (4) without channel fading and noise, which is consistent with approaches in [3].

Results: The accuracy for prediction of the deep over-the-air computation and the baseline approaches are shown in Fig. 3, where the number of devices ranges from 10 to 30. The deep over-the-air computation outperforms the LSTM baseline and provides more accurate prediction especially when there are more than 10 devices. In addition, compared with the ideal multi-access channel, the performance degrades due to the channel fadings and the channel noise. The accuracy drops about 10% when the SNR decrease from 15dB to 5dB.

C. Decentralized Anomaly Detection

Experiment Settings: In this experiment, the decentralized over-the-air computation is used for detecting the anomalous data. CelebA dataset [23] is used for evaluation, where there are 202,599 face images, each annotated with 40 boolean attributes. In particular, we choose four of the attributes, including 'Male', 'Eyeglasses', 'Wearing Hat', and 'Mustache'. With the selected attributes, we build the training and testing sets of face images, where there are 20 images in each set, including 19 normal images and a single target image. For each set, an attribute is first selected from the four attributes and the normal images are selected from images with the particular attribute while the single target image is selected from images without the particular attribute. There is no individual person's face appears in both training and test sets. The images in each training and test sets are assumed to be distributed at the edge devices. With communicating via over-the-air computation, the outliers can be found locally at the edge devices. As before, the channels among the devices are assumed to be Rayleigh channel but the CSI is no longer available for the devices. The system is trained and tested with SNR equals to 20dB.

In the pre-processing DNN, there are 9 convolution layers with 3×3 receptive fields. The model has three sets of convolution layers. The first set has 32, 32, and 64 feature-maps followed by a max-pooling layer with pool-size of 2. The second set has with 64, 64, and 128 feature-maps followed by a max-pooling layer with pool-size of 2. The final set has 128, 128, and 256 feature-maps followed by a max-pooling layer with pool-size of 5. The output of pre-processing has 128 dimensions, which is sent to channel. In the post-processing DNN, there are three fully connected layers with hidden neurons 256, 128, and 1. The output of the final layer is fed to a Softmax layer to get the classification result. We use Adam for optimization and the batch size is 32.

Results: Fig. 4 shows detection results, where the deep over-the-air computation can effectively find the outliers with

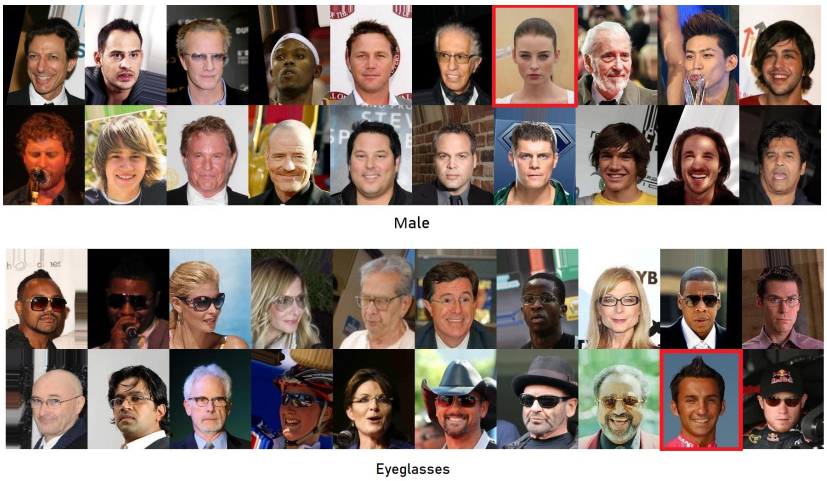


Fig. 4: Decentralized deep over-the-air computation based anomaly detection results with different attributes.

different attributions via sharing features through the wireless multi-access channel. The anomaly detection problem is cast as a binary classification problem at each device and the classification accuracy is 98.5%.

VI. Conclusion

In this paper, a deep learning enabled over-the-air computation framework has been proposed, where the DNN is used as the pre-processing and post-processing functions. Both the centralized and decentralized structures are developed for different applications. In this way, the over-the-air computation functions can be trained for many machine learning applications on the IoT. This paper provides examples that the superposition property of multi-access channels can be leveraged to develop efficient communication architectures for machine learning / artificial intelligence enabled applications. One interesting future direction is how to extend the framework to general wireless channels such as MIMO channels. In addition, how to exploit the superposition property to develop more types of learning algorithms that are suitable for the wireless networks also need to be further explored.

References

[1] A. Zappone M. Di Renzo M. Debbah, "Wireless networks design in the era of deep learning: Model-based AI-based or both?" *IEEE Trans. Commun.* vol. 67, no. 10, pp. 7331-7376, Oct. 2019.

[2] B. Nazer and M. Gastpar, "Computation over multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3498-3516, 2007.

[3] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. Salakhutdinov, and A. Smola,"Deep sets," in *Proc. NeurIPS*, Dec. 2017, pp. 3394–3404.

[4] Z. Qin, G. Li, and H. Ye, "Federated learning and wireless communications," submitted to *IEEE Wireless Commun.*

[5] M. Gastpar, "Uncoded transmission is exactly optimal for a simple Gaussian sensor network," *IEEE Trans. Inf. Theory,* vol. 54, no. 11, pp. 5247–5251, Nov 2008.

[6] J. O'Shea, T. Erpek, and T. C. Clancy, "Deep learning based MIMO communications," *arXiv preprint arXiv:1707.07980,* 2017.

[7] J. Xiao, S. Cui, Z. Q. Luo, and A. J. Goldsmith, "Linear coherent decentralized estimation," *IEEE Trans. Signal Process.*, vol. 56, no. 2, pp. 757–770, Feb 2008.

[8] M. Goldenbaum, and S. Stańczak, "Robust analog function computation via wireless multiple-access channels," *IEEE Trans. Commun.* vol. 61, no. 9, pp. 3863-3877, Aug. 2013.

[9] G. Zhu and K. Huang, "MIMO over-the-air computation for highmobility multi-modal sensing," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6089-6103, Aug. 2019.

[10] M. Goldenbaum, H. Boche, and S. Stańczak, "Harnessing interferencefor analog function computation in wireless sensor networks,", *IEEETrans. Signal Process.*, vol. 61, no. 20, pp. 4893-4906, Oct 2013.

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

[12] M. Amiri, D. Gündüz, "Federated learning over wireless fading channels," *IEEE Trans. Wireless Communi., Early Access*, Feb. 2020.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez,Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Neuips*, 2017, pp. 5998-6008.

[14] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *arXiv preprint arXiv:1812.08434.*, 2018.

[15] C. Qi, H. Su, K. Mo, and L. Guibas, "PointNet: Deep learning on point sets for 3d classification and segmentation," in *Proc. IEEE CVPR* Jul.2017, pp. 652-660.

[16] J. Oliva, B. Poczos, and J. Schneider, "Distribution to distribution regression," in *Proc. ICML*, Jun. 2013, pp.1049-1057.

[17] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.,* vol. 3, no. 4, pp. 563-575, Dec. 2017.

[18] A. Felix, *et al.*, "OFDM autoencoder for end-to-end learning of communications systems," in *Proc. IEEE Int. Workshop Signal Proc. Adv. Wireless Commun.(SPAWC)*, Jun. 2018.

[19] F. Aoudia and J. Hoydis, "End-to-end learning of communications systems without a channel model," *preprint arXiv:1804.02276*, 2018.

[20] H. Ye, G. Li, B.-H. Juang, and K. Sivanesan, "Channel agnostic endto-end learning based communication systems with conditional GAN," in *Proc. IEEE Globecom Workshops*, Dec. 2018, pp. 1–5.

[21] H. Ye, L. Liang, G. Li, and B.-H. Juang, "Deep learning based end-to-end wireless communication systems with conditional GAN as unknown channel," in *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3133–3143, May 2020.

[22] G. Loosli, S. Canu, and L. Bottou, "Training invariant support vector machines using selective sampling," in *Large Scale Kernel Machines*, L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, Eds. Cambridge, MA, USA: MIT Press, 2007, pp. 301–320. [Online]. Available: *http://leon.bottou.org/papers/loosli-canu-bottou-2006*

[23] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. ICCV*, Dec. 2015, pp. 3730-3738.