P. S. Bouzinis, N. A. Mitsiou, P. D. Diamantoulakis, D. Tyrovolas and G. K. Karagiannidis, "Intelligent Over-the-Air Computing Environment," in IEEE Wireless Communications Letters, vol. 12, no. 1, pp. 134-137, Jan. 2023, doi: 10.1109/LWC.2022.3219250.
This is the authors' preprint version and not the final published version. The published version is available by IEEE at the following link: https://ieeexplore.ieee.org/abstract/document/9941007

Intelligent Over-the-Air Computing Environment

Pavlos S. Bouzinis, *Student Member, IEEE*, Nikos A. Mitsiou, *Student Member, IEEE*, Panagiotis D. Diamantoulakis, *Senior Member, IEEE*, Dimitrios Tyrovolas, *Student Member, IEEE*, and George K. Karagiannidis, *Fellow, IEEE*

***Abstract*—A key service of the sixth generation (6G) of wireless networks is envisioned to be native Artificial Intelligence, which calls for radical changes to the way the nodes communicate and perform computations, as well as the role of wireless environment. For this purpose, over-the-air computing (AirComp) is a promising technique for ultra low-latency wireless data aggregation, enabled by the waveform superposition properties of a multiple access channel. In this work, the synergy of decentralized AirComp, reconfigurable intelligent surfaces (RISs) and machine learning is proposed, to transform the wireless environment to intelligent AirComp environment (IACE), i.e., with inherent and advanced capabilities to perform computations in a fully decentralized way at the physical layer. Specifically, we minimize the AirComp error, i.e., the average mean-square errors of devices with respect to a target function, by jointly optimizing the RIS phase-shift vector and the transmission and reception scaling factors of devices. Also, to solve this challenging problem, we propose an online deep neural network (DNN) optimization approach. Finally, simulation results validate the effectiveness of IACE and the proposed DNN approach.**

***Index Terms*—AirComp, reconfigurable intelligent surfaces, deep learning**

I. Introduction

NAtive Artificial Intelligence, which refers to the provision of intelligent functionalities, such as distributed and federated learning (FL), is envisioned to be one the key services of the sixth generation (6G) of wireless networks [1]. To provide such services with respect to the requirements of the next generation Internet-of-Things [2], the use of ultra-low latency wireless data aggregation methods are needed that ideally are solely performed at the physical layer, avoiding any Open Systems Interconnection (OSI) stacks crossing. To this end, using over-the-air computing (AirComp) is a particularly promising approach [3], since it has potential to accomplish ultra-fast data aggregation, by exploiting the superposition property of a multiple access channel (MAC) to compute functions via simultaneous transmissions by all devices. In essence, AirComp fusions the communication and computation processes instead of treating them separately, while it also achieves significant bandwidth savings owing to the concurrent wireless transmissions. Notably, by appropriate processing, AirComp can be used to compute all nomographic functions, and, thus, any other function [4].

This work was supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 957406.

The authors are with the Wireless Communications and Information Processing (WCIP) Group, Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, 54636, Thessaloniki, Greece (e-mails: mpouzinis@ece.auth.gr, nmitsiou@ece.auth.gr, padiaman@ieee.org, tyrovolas@ece.auth.gr, geokarag@auth.gr).

Traditionally, studies on AirComp consider a fusion center, e.g., a central server, which collects the aggregated data and computes the function of desire through the simultaneous device transmissions. For this purpose, linear-analog modulation, channel pre-compensation at each transmitter, and post-processing at the central server are essentially used [4]. Interestingly, the channel compensation can be facilitated by acquiring control over the wireless environment, which can be efficiently achieved by using reconfigurable intelligent surfaces (RISs). RISs have been introduced as meta-surfaces connected with a controller whose properties can be real-time altered and, thus, performing a plethora of electromagnetic functions such as reflection, steering, diffusion, absorption, etc. [5], [6]. For instance, in a classical centralized AirComp framework [7], the mean square error (MSE) among the ground-truth and estimation function was minimized with the aid of a RIS. Also in [8], a multi RIS-assisted AirComp system was designed for a federated learning task, aiming to minimize the AirComp error and accelerate the federated learning convergence.

However, certain applications may imply either the absence of a central server, e.g., device-to-device (D2D) communications, autonomous vehicles and cooperative robotics. In this direction, decentralized AirComp approaches in a D2D manner have been proposed, where each device can recover the target function constructed by the transmissions of the residual devices. For instance, authors in [9]–[11], proposed the use of AirComp-based communication protocols to facilitate decentralized FL. Moreover, in [12], the decentralized AirComp is examined for use in a distributed optimization scenario, where authors conducted a beamforming design to minimize the AirComp error and also investigated its impact on the convergence of the distributed optimization algorithm. It deserves to be noted that although controlling the wireless environment through a RIS would have a transformative impact to the capabilities of decentralized AirComp, by radically increasing the available degrees-of-freedom, the latter has not been investigated in the existing literature.

To this end, in this work, the synergy of decentralized AirComp, RISs and machine learning is proposed, to transform the wireless environment to intelligent AirComp environment (IACE), i.e., with inherent and advanced capabilities to perform computations in a fully decentralized way at the physical layer. Therefore, the IACE, through the exploitation of the MAC properties and the proper adjustment of the wireless environment, aims to facilitate the efficient recovery of a target function in each network device. This capability paves the way towards realizing promising applications, such as distributed

learning, sensing and consensus. In this direction, with the goal to optimize the performance of IACE, the AirComp error is minimized, i.e., the average MSE of devices with respect to a desired target function. Specifically, we jointly optimize the transmission and reception scaling factors of devices, as well as the phase-shifts at the RIS. To efficiently solve the aforementioned challenging problem, an online deep neural network (DNN) approach is properly designed. Finally, simulation results are provided, which validate the effectiveness of IACE and the proposed DNN approach.

II. System model and Problem Formulation

We consider an IACE consisting of a set $\mathcal{N} = \{1, 2, ..., N\}$ of N devices, a RIS equipped with K passive reflective elements, and an intelligent virtual or physical entity/controller. The latter is aware of the channel state information (CSI) and responsible for the IACE orchestration, providing among others feedback to the RIS regarding its optimal configuration. Each device acts as a fusion center which is interested in receiving an aggregation of the data, e.g., arithmetic mean, of the residual network devices, under the absence of a central fusion center. Specifically, the AirComp technique is adopted, where all devices transmit simultaneously by exploiting the waveform superposition property of a MAC. At this point we clarify that the target function each device is aiming to recover through the AirComp is

$$y = \sum_{n\in\mathcal{N}} x_n, \tag{1}$$

i.e., the summation of all data among the network devices, where x_n is the information signal of device n and assumed to be zero-mean unit variance, without loss of generality. It is noted that the considered target function is ubiquitous in decentralized learning techniques, such as FL. Furthermore, we consider a full-duplex scenario, i.e., devices can simultaneously transmit and receive signals, while we also assume that devices are synchronized. Moreover, the direct links among devices are assumed to be blocked, due to unfavorable propagation environment, while at each device the transmitter is perfectly decoupled from the receiver via passive and/or active self-interference cancellation. Following that, the estimated AirComp aggregated signal received by device i is given as

$$\hat{y}_i = c_i\left(b_i\left(\boldsymbol{h}_i^\top\boldsymbol{\Theta}\boldsymbol{h}_i\right)x_i + \sum_{n\in\mathcal{N}\setminus\{i\}} b_n\left(\boldsymbol{h}_i^\top\boldsymbol{\Theta}\boldsymbol{h}_n\right)x_n + z_i\right), \tag{2}$$

where $\boldsymbol{h}_i = [h_{i,1}, h_{i,2}, ..., h_{i,K}]^\top \in \mathbb{C}^{K\times 1}$, with $h_{i,k}$ being the reciprocal channel coefficient between device i and the k-th RIS element and $\boldsymbol{\Theta} = \text{diag}\left([e^{j\theta_1}, e^{j\theta_2}, ..., e^{j\theta_K}]\right) \in \mathbb{C}^{K\times K}$ denotes the diagonal phase shift matrix of RIS with $0 \le \theta_k < 2\pi$, $\forall k$. Moreover, $b_i, c_i \in \mathbb{C}$, $\forall i \in \mathcal{N}$, are the transmission and reception scaling factors of device i, respectively, facilitating the effectiveness of AirComp. Furthermore, $z_i \sim \mathcal{CN}(0, \sigma^2)$ is the AWGN. Each device has a maximum transmit power, denoted as P_0. Thus, it holds

$$\mathbb{E}[|b_i x_i|^2] = |b_i|^2 \le P_0, \quad \forall i \in \mathcal{N}. \tag{3}$$

Note also that the post-processing factor c_i is operated on the digital domain, thus, c_i is not constrained [13]. The model of the proposed system is depicted in Fig. 1. Taking the above into consideration, the computational distortion at user i, which is reflected through the MSE between the estimated function $\hat{y}_i$ and the target function y, is given as

$$\text{MSE}_i = \mathbb{E}\left[|\hat{y}_i - y|^2\right], \quad \forall i \in \mathcal{N}, \tag{4}$$

where the expectation is taken with respect to the randomness of the original signals $\{x_n\}$ and the receiver noise z_i. After some manipulations, it is straightforward to show that the MSE_i can be written as

$$\begin{aligned}\text{MSE}_i = &\left|c_i b_i\left(\boldsymbol{h}_i^\top\boldsymbol{\Theta}\boldsymbol{h}_i\right) - 1\right|^2 \\ &+ \sum_{n\in\mathcal{N}\setminus\{i\}}\left|c_i b_n\left(\boldsymbol{h}_i^\top\boldsymbol{\Theta}\boldsymbol{h}_n\right) - 1\right|^2 + \sigma_i^2|c_i|^2.\end{aligned} \tag{5}$$

Since $\boldsymbol{\Theta}$ is a diagonal matrix and $\boldsymbol{h}_i$ is a vector, the MSE_i can be equivalently given as

$$\begin{aligned}\text{MSE}_i = &\left|c_i b_i\left(\boldsymbol{h}_i^\top\text{diag}(\boldsymbol{h}_i)\boldsymbol{v}\right) - 1\right|^2 \\ &+ \sum_{n\in\mathcal{N}\setminus\{i\}}\left|c_i b_n\left(\boldsymbol{h}_i^\top\text{diag}(\boldsymbol{h}_n)\boldsymbol{v}\right) - 1\right|^2 + \sigma_i^2|c_i|^2,\end{aligned} \tag{6}$$

where $\boldsymbol{v} = [v_1, v_2, ..., v_k]^\top$, with $v_k = e^{j\theta_k}, \forall k \in \mathcal{K}$. Our objective goal is jointly optimize the transmission, reception scalars and the phase shift vector, towards minimizing the average of MSE among users, which gives rise to the following optimization problem

$$\begin{aligned}\min_{\{b_i\},\{c_i\},\{v_k\}} \quad & \frac{1}{N}\sum_{i\in\mathcal{N}}\text{MSE}_i \\ \textbf{\textit{s.t.}} \quad & |b_i| \le \sqrt{P_0}, \quad \forall i \in \mathcal{N}, \\ & |v_k| = 1, \forall k \in \mathcal{K}, \quad v_k, b_i, c_i \in \mathbb{C},\end{aligned} \tag{7}$$

where the objective function can be also written as

$$\begin{aligned}&\frac{1}{N}\sum_{i\in\mathcal{N}}\text{MSE}_i \\ &= \frac{1}{N}\sum_{i=1}^{N}\sum_{n=1}^{N}\left|c_i b_n\left(\boldsymbol{h}_i^\top\text{diag}(\boldsymbol{h}_n)\boldsymbol{v}\right) - 1\right|^2 + \frac{1}{N}\sum_{i=1}^{N}\sigma_i^2|c_i|^2.\end{aligned} \tag{8}$$

Problem (7) is of non-convex nature, due to the coupling of the optimization variables in the objective function and the unit modulus constraint. When encountering similar problems the conventional approaches usually resort to suboptimal solutions, e.g., alternating optimization-based algorithms, which in general may suffer from high computational complexity and non-guaranteed convergence.

III. Proposed Solution

A. Online DNN

Motivated by the encouraging results of online DNN optimization in dealing with RIS-involved problems [14], a DNN is trained in an online fashion for solving problem (7). Specifically, for a given individual channel realization, the DNN is

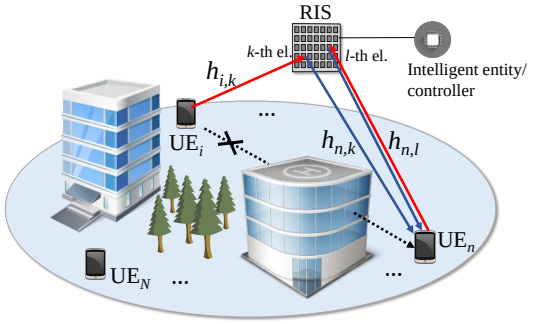


Fig. 1: System model of the IACE.

tuned to estimate the optimal variables i.e., the transmission and reception scaling factors of the devices $\boldsymbol{b}, \boldsymbol{c}$, as well as the phase shift vector $\boldsymbol{v}$ of the RIS elements. It is also clarified that unlike to offline training approach where multiple samples are used for training and afterwards the testing stage follows, online DNN has no testing stage, since for each new generated sample a dedicated network is trained, and thus, the generalization issue does not exist [14]. Following that, the device-to-RIS channel matrix $\boldsymbol{H} = [\boldsymbol{h}_1, ..., \boldsymbol{h}_N] \in \mathbb{C}^{K \times N}$ is given as an input to the DNN. More specifically, the real and imaginary part of the channels are treated as separate features. Hence, the feature vector is expressed as $\boldsymbol{F} \in \mathbb{R}^{2NK \times 1}$, where the entries of $\boldsymbol{H}$ have been concatenated to form $\boldsymbol{F}$.

The DNN's loss function is defined as the objective function of the problem (7), which can be written as

$$\text{Loss} = \frac{1}{N} \sum_{i=1}^{N} \text{MSE}_i, \tag{9}$$

while the DNN is trained to minimize the considered loss function given the input feature vector $\boldsymbol{F}$, and finally extract an estimation of $\boldsymbol{b}$, $\boldsymbol{c}$ and $\boldsymbol{v}$ as outputs. We denote the overall output estimation vector as $\hat{\boldsymbol{o}} = m(\boldsymbol{F}, \boldsymbol{\theta})$, where m is the mapping function of the DNN parameterized by its trainable parameters $\boldsymbol{\theta}$. Note that the loss function can be also expressed as $\text{Loss}(\boldsymbol{F}, \hat{\boldsymbol{o}})$. In the subsequent analysis, $\hat{\boldsymbol{o}}$ will be clearly defined. In the continue, we describe the techniques which manipulate the constraints in (7), towards properly handling the complex optimization variables and guaranteeing the feasibility of the solution extracted by the DNN.

Firstly, we consider that the activation function of the output layer is the *Sigmoid* function. Regarding the phase shift vector $\boldsymbol{v}$, the constraints impose that its complex components have unit modulus. Similarly to [15], we can express the vector $\boldsymbol{v}$ as

$$\boldsymbol{v} = \cos(2\pi\tilde{\boldsymbol{v}}) + j \cdot \sin(2\pi\tilde{\boldsymbol{v}}), \tag{10}$$

where $\tilde{\boldsymbol{v}}$ is the output of the DNN and its entries belong in the range $[0, 1]$, given that the *Sigmoid* is the activation function of the output layer. Next, to meet the constraint $|b_i| \leq \sqrt{P_0}$, we use the auxiliary variables $\tilde{b}_i, \bar{b}_i \in [0, 1]$ to express b_i as

$$b_i = \bar{b}_i \cdot \sqrt{P_0} \left(\cos(2\pi\tilde{b}_i) + j \cdot \sin(2\pi\tilde{b}_i) \right), \quad \forall i \in \mathcal{N}. \tag{11}$$

Note that the outputs of the neural network are the vectors $\tilde{\boldsymbol{b}}, \bar{\boldsymbol{b}}$, while it is obvious that with the proposed transformation the target constraint is indeed satisfied. Finally, recall that the variable c_i is unconstrained. However, to preserve the

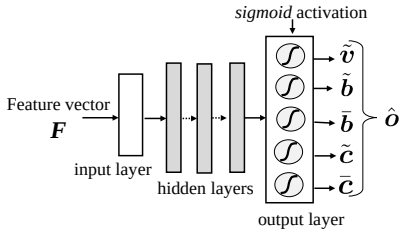


Fig. 2: DNN architecture.

consistency of the neural network architecture, i.e., use the *Sigmoid* as the activation function of each output node, and retain the value range of the output layer at small levels, we adopt a similar transformation for c_i, as follows

$$c_i = \bar{c}_i \cdot B \left(\cos(2\pi\tilde{c}_i) + j \cdot \sin(2\pi\tilde{c}_i) \right), \quad \forall i \in \mathcal{N}, \tag{12}$$

where $\tilde{c}_i, \bar{c}_i \in [0, 1], \forall i \in \mathcal{N}$ are also outputs of the DNN. Moreover, since c_i is not constrained we select B to be a very big constant, and thus, not limiting the range of $c_i, \forall i \in \mathcal{N}$. The aforementioned transformations are of significant importance, since they enable the use of a DNN and address the optimization constraints.

Following that, we highlight that for the final output vector of DNN it holds $\hat{\boldsymbol{o}} = [\tilde{\boldsymbol{v}}^\top, \tilde{\boldsymbol{b}}^\top, \bar{\boldsymbol{b}}^\top, \tilde{\boldsymbol{c}}^\top, \bar{\boldsymbol{c}}^\top]^\top \in \mathbb{L}^{(K+4N)\times 1}$, where $\mathbb{L} = \{x \in \mathbb{R} \,|\, 0 \leq x \leq 1\}$. To this end, it should be noted that the primary variables $\boldsymbol{v}, \boldsymbol{b}$ and $\boldsymbol{c}$ can be directly retrieved from their counterparts $\tilde{\boldsymbol{v}}, \tilde{\boldsymbol{b}}, \bar{\boldsymbol{b}}$ and $\tilde{\boldsymbol{c}}, \bar{\boldsymbol{c}}$, by using the equations (10),(11) and (12), respectively.

B. Baseline solution

In this subsection we propose a baseline solution, which will operate as a benchmark for comparison purposes with the online DNN approach. Hence, a simple intuitive solution to problem (7) comes as follow. The problem can be decomposed into three blocks, corresponding to each one of the optimization variables $\boldsymbol{v}, \boldsymbol{b}$ and $\boldsymbol{c}$. Therefore, we adopt an alternate optimization method, which alternatively solves for one block of variables with the rest variables taking fixed values. When treating $\boldsymbol{b}$ and $\boldsymbol{c}$ as the variables, the problem is convex with respect to $\boldsymbol{b}$ and $\boldsymbol{c}$, respectively. However, when solving for $\boldsymbol{v}$, with $\boldsymbol{b}$ and $\boldsymbol{c}$ fixed, the problem is non-convex due to the unit modulus constraint. In this case, we relax $|v_k| = 1$ to $|v_k| \leq 1, \forall k \in \mathcal{K}$, and afterwards enforce the optimal solution to satisfy the unit modulus constraint by projecting it to the unit complex circle, i.e., $u_k^* \leftarrow \frac{u_k^*}{|u_k^*|}, \forall k$.

C. Complexity Analysis

For the DNN architecture, without loss of generality, we consider a single hidden layer with L neurons. The forward-pass complexity, which is dominated by the weight matrix multiplication cost, is given as $\mathcal{O}(2KNL + L(4N + K))$ and is equivalent to $\mathcal{O}(KNL)$. Given that the backward-pass has the same computational cost with the forward-pass [16], the overall back propagation algorithm for training the online DNN is of the order $\mathcal{O}(I_E KNL)$, where I_E is the number of training epochs. As for the baseline solution, the major complexity lies in solving problem (7) with respect to $\boldsymbol{v}$. For a standard solver, e.g., the interior-point method, the complexity is $\mathcal{O}(I_0 K^3)$, where I_0 is the number of total iterations. Usually

we have $K \gg N$, which implies that the baseline algorithm's execution time per iteration will be greater. To this end, it is noted that the training of the DNN is performed on the intelligent entity/controller, which is equipped with sufficient computation capabilities. We also highlight that the execution time of the online DNN can be significantly accelerated with the usage of dedicated hardware for parallel computations, e.g., GPU, rendering the proposed approach suitable for real-time optimization.

IV. Performance Evaluation and Discussion

We consider that the network devices are uniformly distributed in a disk with radius equal to 40m. Moreover, the RIS is located at the center of the disk in height $z = 30\,\mathrm{m}$. For the device-RIS links, i.e., $\boldsymbol{H}$, we consider Rician fading with factor β, thus we have [7]

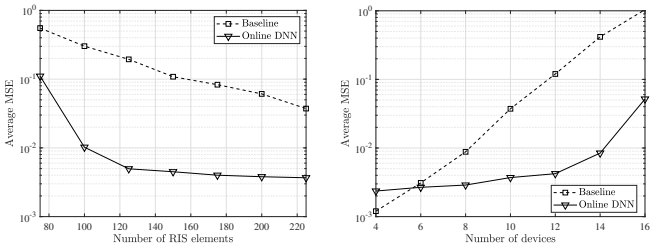
$$\boldsymbol{H} = \sqrt{T_0(d/d_0)^{-a}}\left(\sqrt{\tfrac{\beta}{\beta+1}}\boldsymbol{H}^{\mathrm{LoS}} + \sqrt{\tfrac{1}{\beta+1}}\boldsymbol{H}^{\mathrm{NLoS}}\right), \tag{13}$$

where $\boldsymbol{H}^{\mathrm{LoS}}$ and $\boldsymbol{H}^{\mathrm{NLoS}}$ denote the line-of-sight (LoS) and the non-line-of-sight (NLoS) components, respectively. Moreover, T_0 is the path loss at the reference distance $d_0 = 1$m, d denotes the distance between the transmitter and the receiver, and a is the path loss exponent. Unless specified otherwise, we set $T_0 = -25\,\mathrm{dB}$, $\beta = 10$, $a = 2.2$, $P_0 = 40\,\mathrm{dBm}$, and $\sigma^2 = -90\,\mathrm{dBm}$.

We construct a feed-forward fully connected DNN, consisting of two hidden layers with 128 and 64 nodes, respectively. The *ReLU* is selected as the activation function of all layers, except for the output layer, where the *Sigmoid* function is utilized. Furthermore, we adopt the Adam optimizer and set the initial learning rate equal to 0.1. The number of maximal epochs is set to 1500, while in order to accelerate convergence, we decay the learning rate by a factor of 0.33 when the loss does not decrease for consecutive 10 epochs.

In Fig. 3a, the impact of RISs' elements on the average MSE of devices is demonstrated. It can be observed that with the increase of the reflecting elements, the average MSE of AirComp is decreasing. This is reasonable, since more elements translate to higher degrees-of-freedom, facilitating the construction of the desired function in each devices' receiver. Moreover, the proposed online DNN approach clearly outperforms the baseline solution, exhibiting a performance gain which is higher than an order of magnitude. Finally, it can be seen that for high number of reflecting elements, the online DNN presents a slight reduction in the average MSE. This result may stem from the saturation of the system, i.e., it is hard to further reduce the average MSE even by employing additional RIS elements, owing to the presence of AWGN. Notice that the AWGN term in (8) may prevail when targeting small MSE values, hindering its further minimization.

In figure Fig. 3b, we evaluate the impact of devices' number N on the average MSE. The number of reflective elements has been set as $K = 225$. Although, for small number of devices, e.g., 4 and 6 devices, the baseline algorithm performs slightly better or equivalent to the proposed DNN approach, the performance gap among the two methods is rapidly increasing along with the number of devices. It is evident that



(a) AMSE vs the number of RIS elements. (b) AMSE vs the number of devices.

Fig. 3: AirComp average MSE (AMSE)

the online DNN is superior to the baseline scheme in achieving lower MSE values, a fact that justifies its adoption. The latters' performance degradation, can be probably attributed to the large scale of the system as the number of devices is increasing, enforcing the algorithm to stuck into local minima. Moreover, in the case of the online DNN approach, it is observed that the average MSE grows abruptly for high N values. This result implies that higher number of reflective elements have to be utilized, in order to efficiently employ the AirComp for scenarios with large number of devices.

To this end, the results corroborate the effectiveness of the proposed approach towards creating the IACE.

References

[1] W. Tong and P. Zhu, *6G, the Next Horizon: From Connected People and Things to Connected Intelligence*. Cambridge University Press, 2021.

[2] A. Brékine, M. Brynskov, F. Facca, and G. Hrasko, "Building a roadmap for the next generation Internet of Things. Research innovation and implementation 2021–2027 (scoping paper)," *M. Brynskov, FM Facca, and G. Hrasko, Eds*, 2019.

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

[4] G. Zhu, J. Xu, K. Huang, and S. Cui, "Over-the-air computing for wireless data aggregation in massive IoT," *IEEE Wireless Commun.*, vol. 28, no. 4, pp. 57–65, 2021.

[5] C. Liaskos, S. Nie, A. Tsioliaridou, A. Pitsillides, S. Ioannidis, and I. Akyildiz, "A New Wireless Communication Paradigm through Software-Controlled Metasurfaces," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 162–169, 2018.

[6] S. A. Tegos, D. Tyrovolas, P. D. Diamantoulakis, C. K. Liaskos, and G. K. Karagiannidis, "On the Distribution of the Sum of Double-Nakagami-m Random Vectors and Application in Randomly Reconfigurable Surfaces," *IEEE Trans. Veh. Technol.*, pp. 1–1, 2022.

[7] W. Fang, Y. Jiang, Y. Shi, Y. Zhou, W. Chen, and K. B. Letaief, "Over-the-air computation via reconfigurable intelligent surface," *IEEE Trans. Commun.*, vol. 69, no. 12, pp. 8612–8626, 2021.

[8] W. Ni, Y. Liu, Z. Yang, H. Tian, and X. Shen, "Federated learning in multi-ris aided systems," *IEEE Internet Things J.*, 2021.

[9] E. Ozfatura, S. Rini, and D. Gündüz, "Decentralized SGD with over-the-air computation," in *Proc. IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.

[10] H. Xing, O. Simeone, and S. Bi, "Decentralized federated learning via sgd over wireless d2d networks," in *Proc. International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2020, pp. 1–5.

[11] Y. Shi, Y. Zhou, and Y. Shi, "Over-the-air decentralized federated learning," in *Proc. IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2021, pp. 455–460.

[12] Z. Lin, Y. Gong, and K. Huang, "Distributed over-the-air computing for fast distributed optimization: Beamforming design and convergence analysis," *arXiv preprint arXiv:2204.06876*, 2022.

[13] W. Liu, X. Zang, Y. Li, and B. Vucetic, "Over-the-air computation systems: Optimization, analysis and scaling laws," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, pp. 5488–5502, 2020.

[14] J. Gao, C. Zhong, G. Y. Li, and Z. Zhang, "Online deep neural network for optimization in wireless communications," *IEEE Wireless Commun. Lett.*, vol. 11, no. 5, pp. 933–937, 2022.

[15] J. Gao, C. Zhong, X. Chen, H. Lin, and Z. Zhang, "Unsupervised learning for passive beamforming," *IEEE Commun. Lett.*, vol. 24, no. 5, pp. 1052–1056, 2020.
[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.