Federated Dictionary Learning from Non-IID Data

Alexandros Gkillas
Department of Computer Engineering and Informatics,
University of Patras,
Patras, Greece
Email: st1003586@ceid.upatras.gr

Dimitris Ampeliotis
Department of Digital Media and Communication,
Ionian University,
Argostoli, Greece
Email: ampeliotis@ionio.gr

Kostas Berberidis
Department of Computer Engineering and Informatics,
University of Patras,
Patras, Greece
Email: berberid@ceid.upatras.gr

***Abstract*—The problem of learning a common dictionary by following a federated learning framework, in a network where each edge user may have statistically different data, is considered. In such a challenging setting, the Federated Averaging solution is shown to exhibit poor performance. To alleviate the drawbacks of this approach, two more elaborate schemes are proposed. The new schemes are designed so as to offer increased performance in the case of non-i.i.d. data, that is the focus in this work. Extensive simulation results are presented for an image processing scenario, that demonstrate the effectiveness of the proposed methods.**

I. Introduction

The recent groundbreaking success of Machine Learning (ML) techniques, especially in problems that, until recently, were considered too sophisticated to be solved by some automatic procedure[1], has transformed it into a valuable tool able to confront various challenging engineering problems in many and diverse fields, ranging from wireless communications [2] to social networks [3]. The demand for training more and more complex ML models, however, requires significant computational resources and immensely big training datasets [4]. Such requirements constitute centralized computing architectures inadequate, and decentralized approaches must be employed to offer a possible solution. However, distributed ML is not a straightforward task and many researchers have started working towards engineering efficient algorithms to achieve this goal. Distributed ML approaches can be categorized mainly into three categories. In the first category, distributed edge users (devices) communicate with a fusion server that gathers data for centralized processing [5], [6]. On the other hand, fully decentralized strategies that rely upon local information exchange among neighboring nodes have been developed over the last years [7], [8]. More recently, a third category has emerged: Federated Learning (FL) [9], [10], [11] stands somewhere in between the above two categories, proposing a paradigm in which data is collected locally at the edge users and some processing is also performed locally, while global information is shared between a central server and the dispersed devices (or users).This approach alleviates the main drawbacks of the other two categories, namely, the limited storage and computation resources of fusion center-based methods and the intense communication requirements of fully distributed architectures. FL poses new statistical and systems challenges in training ML models over distributed networks of edge devices. Mobile phones, wearable devices, and smart homes are just a few of the modern distributed networks generating massive amounts of data each day. Due to the growing storage and computational power of devices in these networks, it is increasingly attractive to store data locally and push more network computation to the edge. The nascent field of FL explores training statistical models directly on edge devices. FL approaches can be categorized into three main categories [12], namely horizontal FL, vertical FL and federated transfer learning. These categories arise for identical or not feature and/or index spaces of the distributed datasets.

In this work, the focus is on FL methods for solving the so-called dictionary learning problem [13], that can be utilised in many applications, e.g. [14]. While many efficient centralized algorithms exist for this problem, e.g., [15], the work on decentralized methods has mainly focused on fully distributed methods, for example [16], [17], [18]. Very recently, FL based approaches for dictionary learning have appeared [19]. Different from previous works on decentralized methods for dictionary learning focusing on i.i.d. scenarios, in this work, we examine an even more realistic and challenging scenario, i.e., the non i.i.d case where the data on the nodes of the network may follow different statistical distributions. In light of this, we fill this gap by providing federated dictionary learning frameworks able to effectively handle the statistical heterogeneity of the data.

II. Problem Formulation

To accurately formulate the considered Federated Dictionary Learning (FDL) problem, we consider a set of N edge users (in the sequel, we also use the terms users, edge devices, nodes to refer to these entities), where each user $n \in \mathcal{N} = \{1, 2, \ldots, N\}$ owns a local private dataset represented by the matrix

$$\boldsymbol{Y}_n \in \mathbb{R}^{d \times m_n}, \qquad n \in \mathcal{N}\ , \tag{1}$$

where d denotes the dimension of the local data samples and m_n is the size of the dataset of user n. As an example, the data of each node may correspond to image patches [20], extracted from some local image database. Note that in this study, we assume that the local datasets follow different statistical distribution, thus exploring a non i.i.d scenario. The edge users are interested to compute a *common dictionary* matrix $\boldsymbol{D} \in \mathbb{R}^{d \times K}$ that is suitable for the sparse representation of the entire dataset, i.e. the concatenation of the datasets of all the users. Under the Federated Learning protocol, the collaboration of the users for computing such a common dictionary is

This work was supported in part by the University of Patras and the RPF, Cyprus, under the project INFRASTRUCTURES/1216/0017 (IRIDA).

978-1-6654-7822-9/22/$31.00 ©2022 IEEE

performed via a central server, that is able to communicate with all the edge devices and follows an iterative procedure. In particular, in each communication round, each user computes local dictionaries, sends them to the central server that applies some aggregation rule, and the aggregate model (dictionary) is sent to all the users for using it to proceed to the next round. The details of the FL flow, particularised for the considered problem, are given in the following section.

III. Federated Dictionary Learning Flow

The Federated Dictionary Learning (FDL) framework is an iterative procedure that employs several communication rounds. At every communication round t, each edge device (or edge user) n employs its private dataset $\boldsymbol{Y}_n$ to compute a local dictionary $\boldsymbol{D}_n^{(t)} \in \mathbb{R}^{d\times K}$ by solving the following optimization problem

$$\boldsymbol{D}_n^{(t)} = arg \min_{\boldsymbol{D}_n, \boldsymbol{G}_n} \frac{1}{2} \left\| \boldsymbol{Y}_n - \boldsymbol{D}_n \boldsymbol{G}_n \right\|_{\mathrm{F}}^2 + \lambda \left\| \boldsymbol{G}_n \right\|_1, \quad (2)$$

where $\boldsymbol{G}_n \in \mathbb{R}^{K\times m_n}$ stands for the sparse representation matrix of the n-th user. It should be highlighted that the edge devices rely solely on their local data, thus indicating that the dataset $\boldsymbol{Y}_n$ may be insufficient for learning an accurate dictionary that is able to generalize on data generated from different distributions. In particular, a dictionary $\boldsymbol{D}_n^{(t)}$ learnt via the optimization problem in (2) is likely to perform poorly when applied to data different from $\boldsymbol{Y}_n$. The above-mentioned challenging issue may be effectively addressed via the collaboration of edge users.

In light of this, in our study, we propose a novel federated learning framework to solve the dictionary learning problem, allowing users to collaboratively learn a global (common) dictionary under the coordination of a central server, without sharing any sensitive information concerning their private data. Thus, the server targets to solve the following optimization problem in order to compute the global dictionary, say $\boldsymbol{D}_*$,

$$\boldsymbol{D}_* = \arg\min_{\boldsymbol{D}} \sum_{n=1}^{N} \left(\frac{1}{2} \left\| \boldsymbol{Y}_n - \boldsymbol{D}\boldsymbol{G}_n \right\|_{\mathrm{F}}^2 + \lambda \left\| \boldsymbol{G}_n \right\|_1 \right). \quad (3)$$

In general, the server requires access to all the individual datasets, i.e., $\{\boldsymbol{Y}_n\}_{n=1}^N$, in order to obtain the global dictionary from the optimization problem in (3). However, in this study, two novel dictionary learning methods are derived based on the federated learning framework in order to learn a global dictionary from edge users that keep their local data private and share only their local dictionaries. In more detail, the iterative procedure consists of a number of communication rounds where each round consist of the following four steps:

1) Each node n computes/updates a local copy of the dictionary, denoted as $\boldsymbol{D}_n^{(t)}$, using its local data $\boldsymbol{Y}_n$ and any information sent by the central server at the previous iterations
2) Each node n sends $\boldsymbol{D}_n^{(t)}$ to the central server (in this work, we extend this step by letting the nodes send some additional information to the central server)
3) The central server applies some *model fusion* rule, to derive one common dictionary $\boldsymbol{D}^{(t)}$ from the N dictionaries that it receives
4) The central server sends the common dictionary $\boldsymbol{D}^{(t)}$ to all the nodes, and the next iteration $t+1$ is performed

IV. Proposed Federated Dictionary learning methods

In this section the proposed federated dictionary learning methods are described.

A. Federated Averaging Dictionary Learning (FedAvg-DL)

Concretely, to formulate the proposed learning scheme, each edge user updates its current local dictionary aiming to solve the optimization problem described in (2). To effectively tackle this non-convex problem [13], the users employ an iterative alternating optimization (AO) scheme of I iterations, splitting the local dictionary problem into two sub-problems, namely the sparse coding and dictionary update [13]. Specifically, the following equation are employed

$$\begin{aligned} \boldsymbol{G}_n^{(t,i+1)} &= \mathcal{F}\left(\boldsymbol{D}_n^{(t,i)}, \boldsymbol{G}_n^{(t,i)}, \boldsymbol{Y}_n\right) \\ \boldsymbol{D}_n^{(t,i+1)} &= \boldsymbol{D}_n^{(t,i)} + \mu \left(\boldsymbol{Y}_n - \boldsymbol{D}_n^{(t,i)} \boldsymbol{G}_n^{(t,i+1)}\right) (\boldsymbol{G}_n^{(t,i+1)})^T \end{aligned} \quad (4)$$

where t denotes the t-th communication round, i stands for the i-th local iteration, $\mu > 0$ is the gradient step-size for the update of the local dictionary, and $\mathcal{F}(\cdot)$ denotes some proper sparse coding algorithm (e.g., the Orthogonal Matching Pursuit - OMP). After I iterations of the form (4), the users send the resulting local dictionaries to the server.

On the server-side, the server aims to compute the global dictionary $\boldsymbol{D}$ by adopting an averaging fusion rule that aggregates all the received local dictionaries from the participated users. In particular, the server computes an average of the local dictionaries, i.e.,

$$\boldsymbol{D}^{(t)} = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{D}_n^{(t)}, \quad (5)$$

$$\boldsymbol{D}^{(t)} = \arg\min_{\boldsymbol{D}} \sum_{n=1}^{N} \left\| \boldsymbol{D} - \boldsymbol{D}_n^{(t)} \right\|_{\mathrm{F}}^2. \quad (6)$$

In the sequel, the central server sends the aggregated dictionary to all the users. The users set their local dictionaries equal to the aggregated dictionary and use the update equations in (4) to update it according to their local data. The above mentioned procedure is repeated for T communication rounds until the global dictionary reaches convergence.

Although, this proposed scheme is expected to perform well in the case where all nodes have similiar (e.g., i.i.d.) data, in more realistic scenarios where the datasets of the users follow different distributions (i.e., non-i.i.d. data), the proposed averaging strategy of the local dictionaries may affect heavily the performance of the resulting global dictionary. This phenomenon is known in literature as *model divergence*, where the averaged local model parameters may also be far away from the ideal global model parameters (the model obtained when the data on the local devices is i.i.d.) [21]. In other words, the global model may fail to generalize on each user's data, thus resulting in a global model biased to

some data distribution of a specific user [11], or even, the aggregate dictionary may not be suitable for the data of any user at all. In view of this, in the following sections, we explore novel model/dictionary aggregation rules and local update mechanisms that lead to schemes able to handle the statistical heterogeneity of the data.

B. Extended Federated Dictionary Learning - (E-FedAvg-DL)

In order to handle the case in which the datasets of the users follow significantly different distributions, a novel approach is proposed by modifying both the local learning stages and the centralized aggregation stage of the previous federated dictionary learning framework (i.e., FedAvg-DL). In more detail, we argue that more efficient federated dictionary learning methodologies can be derived by introducing the concept of *atom selection probabilities* at each node. In particular, at each node n, we define the vector $\boldsymbol{p}_n \in \mathbb{R}^K$, that holds the estimated probabilities (i.e., relative frequencies) for the use of each of the K local dictionary atoms in the representation of the local data $\boldsymbol{Y}_n$. These probabilities are estimated at every communication round, after the local dictionary update steps have been completed, and in the sequel, they are sent to the central server along the current local dictionaries, so as to help the server implement some more elaborate aggregation rule, as compared to the simple averaging operation in (5).

Thus, at each communication round t, the central server has access to the N local dictionaries $\{\boldsymbol{D}_n^{(t)}\}_{n=1}^N$, and the corresponding atom selection probability vectors $\{\boldsymbol{p}_n^{(t)}\}_{n=1}^N$ for each node. It should be noted at this point that the transmission of the vectors $\boldsymbol{p}_n^{(t)}$ from the edge users to the central server induces only a small communication overhead, with respect to the transmission of the local dictionaries, especially when the dimension d of the involved dictionaries is large. Taking into account the information that is available at the central server, a new, weighted average, fusion rule is proposed that utilizes effectively the additional information, i.e.,

$$\boldsymbol{D}^{(t)}[k] = \sum_{n=1}^{N} \boldsymbol{q}_n^{(t)}[k] \cdot \boldsymbol{D}_n^{(t)}[k] \ , \ k = 1, 2, \ldots, K \ , \qquad (7)$$

where $\boldsymbol{q}_n^{(t)} \in \mathbb{R}^K$ is a vector of combination weights whose k-th element is defined by

$$\boldsymbol{q}_n^{(t)}[k] = \frac{\boldsymbol{p}_n^{(t)}[k]}{\sum_{n=1}^{N} \boldsymbol{p}_n^{(t)}[k]} \ , \ k = 1, 2, \ldots, K \ ,$$

and $\boldsymbol{D}^{(t)}[k]$ is the k-th column (atom) of the dictionary matrix $\boldsymbol{D}^{(t)}$. This fusion rule guarantees that the atoms of the local dictionaries that are frequently used by the users will obtain high weights during this weighted averaging scheme, thus preserving atoms that encapture valuable information for the local datasets of the users.

Focusing, now, on the dictionary update stage of users, instead of solving the optimization problem in (2) to derive the local updated dictionaries, we modify the objective function by introducing a proper regularization term to overcome the statistical diversity among users. In particular, the modified optimization problem is described by the cost function

$$\frac{1}{2} \left\| \boldsymbol{Y}_n - \boldsymbol{D}_n \boldsymbol{G}_n \right\|_{\mathrm{F}}^2 + \lambda \left\| \boldsymbol{G}_n \right\|_1 + \gamma \left\| (\boldsymbol{D} - \boldsymbol{D}_n) \cdot \boldsymbol{P}_n \right\|_F^2 , \qquad (8)$$

where $\boldsymbol{D}_n$ denotes the local dictionary of user n, $\boldsymbol{D}$ is the global dictionary sent by the central server, and γ stands for the penalty parameter that balances the relative impact of the global dictionary on the learning stage of the local dictionary. Furthermore, the matrix $\boldsymbol{P}_n = diag(1 - \boldsymbol{p}_n)$ is diagonal with size $K \times K$ containing the vector $\boldsymbol{r}_n = 1 - \boldsymbol{p}_n$ on its main diagonal. The motivation for this idea is to allow the users obtaining a local dictionary, which is able to accurately describe their local data, with an additional constraint that the resulting dictionaries need to remain close to global dictionary. In more detail, due to the diagonal matrix $\boldsymbol{P}_n$, the atoms that are rarely used during the local learning stage (i.e., that have small atom selection probabilities) should not diverge significantly from the respective atoms of the global dictionary (that is, large weight 1-p in the last term of the cost function in (8)), since these atoms, although they are not used for the representation of the local dataset, may be beneficial for other users with different data distributions. Thus, the rarely used atoms should remain unchanged. On the other hand, the most *popular* atoms used frequently for the representation of the local dataset should be allowed to be altered more, since they capture the local information of the dataset $\boldsymbol{Y_n}$. The above simple, yet effective, procedure aims to alleviate the model divergence problem described in the previous section.

Similar to Section IV-A, each edge user n can efficiently address the proposed problem in (8) via an alternating optimization procedure, thus leading to the following iterative scheme

$$\begin{aligned} \boldsymbol{G}_n^{(t,i+1)} &= \mathcal{F}\left(\boldsymbol{D}_n^{(t,i)}, \boldsymbol{G}_n^{(t,i)}, \boldsymbol{Y}_n\right) \\ \boldsymbol{D}_n^{(t,i+1)} &= \boldsymbol{D}_n^{(t,i)} + \mu\left(\boldsymbol{Y}_n - \boldsymbol{D}_n^{(t,i)} \boldsymbol{G}_n^{(t,i+1)}\right) (\boldsymbol{G}_n^{(t,i+1)})^T \\ &\quad + \mu\left(\gamma(\boldsymbol{D}^t - \boldsymbol{D}_n^{(t,i)}) \boldsymbol{P}\boldsymbol{P}^T\right) \end{aligned} \qquad (9)$$

where t denotes the t-th communication round and i is the i-th local iteration.

C. Federated Dictionary learning with personalized atoms - (FedAvg-DL-PerA)

Motivated by the so-called personalization deep learning approaches, which allow the users to have personalized layers in the neural network models [22], [23], in this study, we extend this idea by introducing the concept of the personalization to the considered federated dictionary learning problem. Focusing on the server-side, we modify the previous fusion rule in (7), proposing a new one that incorporates the notion of the personalization. Based on the atom selection probabilities of the local dictionaries, the server is able to generate a global dictionary with a proper number of personalized atoms, namely atoms that describe accurately more detailed structures of the local datasets, while the rest atoms of the global dictionary aim to capture the global structures across all the users datasets.

The proposed fusion rule at each t communication round consists of two main phases. During the **first phase**, the

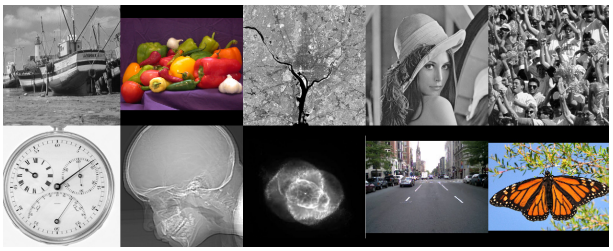

Fig. 1. The images used for generating the input data at the $N = 10$ edge users considered in the experimental part.

server computes the global dictionary $\boldsymbol{D}^{(t)}$ using the fusion rule described in equation (7). Having obtained the global dictionary, during the **second phase**, the server exploits the atom selection probabilities (i.e., the vectors $\{p_n^{(t)}\}_{n=1}^N$) of the local dictionaries (i.e., $\{\boldsymbol{D}_n^{(t)}\}_{n=1}^N$) in order to identify the most frequently used atoms in each local dictionary. In more detail, we employ a simple condition that examines if the selection probability of each atom in each local dictionary is sufficiently high. In greater detail, we examine if

$$p_n^{(t)}(j) > T_p \quad \forall\, j = 1, ..., K \quad \& \quad n = 1, ..., N \tag{10}$$

where $p_n^{(t)}(j)$ denotes the selection probability of j-th atom of the n-th local dictionary $\boldsymbol{D}_n$ and T_p is some properly defined threshold. All atoms from all the local dictionaries that meet the above criteria can be used as personalized atoms, thus resulting in a new global dictionary given by the following relation

$$\boldsymbol{D}_{new}^{(t)} = \left[\boldsymbol{D}^{(t)} \;\; \boldsymbol{D}_{per}^{(t)}\right] \tag{11}$$

where $\boldsymbol{D}^{(t)} \in \mathbb{R}^{d\times K}$ is the global dictionary computed from equation (7), $\boldsymbol{D}_{per}^{(t)} \in \mathbb{R}^{d\times K_{per}}$ denotes a matrix containing the personalized atoms, i.e., the atoms that meet the condition in (10) and $\boldsymbol{D}_{new}^{(t)} \in \mathbb{R}^{d\times (K+K_{per})}$ is the new derived global dictionary. To control the size of the resulting new global dictionary, we employ a pruning strategy according to which if two atoms from the dictionary $\boldsymbol{D}_{new}^{(t)}$ are too coherent (similar) we remove one of them. Concerning the local updates of dictionaries on the users-side, this stage remains unchanged as described in previous section. The proposed federated dictionary learning framework, called FedAvg-DL-PerA is able to effectively surmount the statistical heterogeneity of the local datasets, since it contains some highly representative personalized atoms that are widely used by users to describe their respective local datasets.

V. Numerical Results

To demonstrate the efficacy and applicability of the proposed federated dictionary learning approaches extensive numerical experiments were conducted in the context of image processing. In particular, a network with $N = 10$ edge users (nodes) is considered, where each user contains data derived from some real world image. The considered images (see, Figure 1) were selected carefully so that their distributions are quite different, thus corresponding to a non i.i.d. scenario. The images with size 256×256 were processed into non-overlapping patches of 8×8 pixels, hence generating the users local datasets $\{Y_n\}_{i=1}^{10}$. The goal is to compute a global dictionary $\boldsymbol{D}$ that represent accurately all the local datasets (images) of the users.

Implementation Details: In all experiments, the users employed the OMP algorithm, in place of the sparse coding function $\mathcal{F}(\cdot)$ defined in (4) and (9) with sparsity equal to 10. For all cases, the local dictionaries employed $K = 128$ atoms, thus leading to dictionaries with size 64×128. The threshold value that controls the atom selection probabilities of the frequently used atoms defined in (10) was set to $T_p = 0.75$. Finally, the parameter γ in (8) was set to 0.9.

Evaluation metric: In the following we present the performance of the 3 proposed FDL methods (i.e., FedAvg-DL, E-FedAvg-DL, FedAvg-DL-PerA) against the centralized problem (3) that has access to all the datasets of the users. To quantify the performance accuracy of the proposed models, the representation error (Global error) for all the data of the users is adopted, defined as follows $e^{(t)} = \left\|Y - D^{(t)}G^{(t)}\right\|_F^2$, where $\boldsymbol{Y} = \cup_{n=1}^N \boldsymbol{Y}_n$ is the matrix of all data, $\boldsymbol{D}^{(t)}$ denotes the global dictionary derived from the proposed methods at the t-th round and $\boldsymbol{G}^{(t)}$ is the sparse representation matrix.

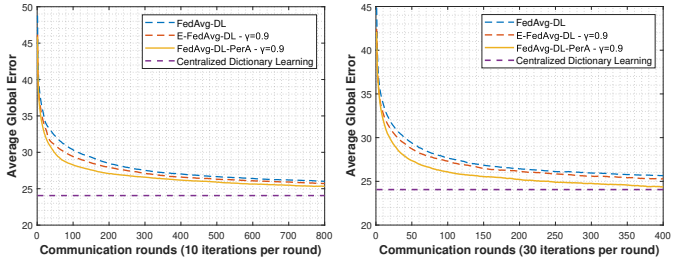

Fig. 2. Average representation error for the 3 proposed FDL approaches.

According to Figure 2, it is evident that the proposed methods, i.e., E-FedAvg-DL, FedAvg-DL-PerA exhibits superior performance as compared to the FedAvg-DL approach that employs only a simple averaging rule, which ignores the highly non i.i.d nature of the considered datasets. Focusing on the two best performing approaches, the FedAvg-DL-PerA method that utilizes the idea of the personalized atoms outperforms the E-FedAvg-DL that employs only the concept of the selection probabilities. This remark can be attributed to the fact that the first approach is able to surmount the statistical heterogeneity of the datasets more efficiently, since apart from the selection probability idea, the derived global dictionary contains some highly representative personalized atoms that are widely used by users to describe their respective local datasets. The superiority of the FedAvg-DL-PerA approach can be justified further, since not only converges faster than the other approaches but more importantly requires considerably less communication rounds to reach the performance of the centralized dictionary algorithm. Considering the high communication cost that introduces the repeated exchange of the models between the edge users and the server, the above finding renders the FedAvg-DL-PerA methodology ideal for real-time applications.

VI. Conclusion

In this study the problem of dictionary learning from the federated learning perspective, was studied. Considering a particularly challenging scenario in which the edge users contain data derived from different distributions, more elaborate federated dictionary learning schemes were proposed,

as compared to the federated averaging approach in order to handle effectively the statistical heterogeneity of the local datasets. Simulations results were conducted in the context of image processing to verify the efficacy and applicability of the proposed methods.

REFERENCES

[1] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[2] L. Liang, H. Ye, and G. Y. Li, "Toward intelligent vehicular networks: A machine learning framework," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 124–135, 2018.

[3] S. Deng, L. Huang, G. Xu, X. Wu, and Z. Wu, "On deep learning for trust-aware recommendations in social networks," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 5, pp. 1164–1177, 2016.

[4] X.-W. Chen and X. Lin, "Big data deep learning: challenges and perspectives," *IEEE access*, vol. 2, pp. 514–525, 2014.

[5] A. Agarwal and J. C. Duchi, "Distributed delayed stochastic optimization," *Advances in neural information processing systems*, vol. 24, 2011.

[6] M. Zinkevich, M. Weimer, L. Li, and A. Smola, "Parallelized stochastic gradient descent," *Advances in neural information processing systems*, vol. 23, 2010.

[7] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. ARTICLE, pp. 311–801, 2014.

[8] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[9] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[10] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *arXiv preprint arXiv:1602.05629*, 2016.

[11] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[12] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[13] I. Tošić and P. Frossard, "Dictionary learning," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 27–38, 2011.

[14] A. Gkillas, D. Ampeliotis, and K. Berberidis, "Efficient coupled dictionary learning and sparse coding for noisy piecewise-smooth signals: Application to hyperspectral imaging," in *IEEE International Conference on Image Processing (ICIP 2020)*, Abu Dhabi, United Arab Emirates, 25-28 October 2020.

[15] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[16] J. Chen, Z. J. Towfic, and A. H. Sayed, "Dictionary learning over distributed models," *IEEE Transactions on Signal Processing*, vol. 63, no. 4, pp. 1001–1016, 2014.

[17] A. Daneshmand, G. Scutari, and F. Facchinei, "Distributed dictionary learning," in *2016 50th Asilomar Conference on Signals, Systems and Computers*. IEEE, 2016, pp. 1001–1005.

[18] D. Ampeliotis, C. Mavrokefalidis, and K. Berberidis, "Distributed dictionary learning via projections onto convex sets," in *European Signal Processing Conference (EUSIPCO 2017)*, Kos Island, Greece, August 28 - September 2 2017.

[19] K. Huang, X. Liu, F. Li, C. Yang, O. Kaynak, and T. Huang, "A federated dictionary learning method for process monitoring with industrial applications," *IEEE Transactions on Artificial Intelligence*, 2022.

[20] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[21] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-IID data: A survey," *Neurocomputing*, vol. 465, pp. 371–390, nov 2021.

[22] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," 2019. [Online]. Available: https://arxiv.org/abs/1912.00818

[23] P. P. Liang, T. Liu, L. Ziyin, N. B. Allen, R. P. Auerbach, D. Brent, R. Salakhutdinov, and L.-P. Morency, "Think locally, act globally: Federated learning with local and global representations," 2020. [Online]. Available: https://arxiv.org/abs/2001.01523