Transform-Domain Federated Learning for Edge-Enabled IoT Intelligence

Lei Zhao, *Student Member, IEEE*, Lin Cai, *Fellow, IEEE*, and Wu-Sheng Lu, *Life Fellow, IEEE*

***Abstract*—Federated learning (FL) deployed in the edge network environment is a promising approach for combining the separated training results based on the isolated local data sensed by various Internet of Things (IoT) devices. However, the limited computing resources for the training of various application models in each edge server and the communication burden among the edge server and numerous IoT devices greatly impact the realization of IoT intelligence. In this article, we propose transform-domain FL schemes based on discrete cosine transform (DCT-FA) and discrete wavelet transform (DWT-FA) to achieve better training efficiency and reduce the communication burden for IoT devices. Furthermore, when the amount of training data is limited, we propose to combine time-domain features and frequency-domain features in FL (CDCT-FA) that turns out to achieve much higher test accuracy. From the experimental results, the transform-domain FL schemes are shown to be promising, given the different constraints and requirements of various IoT intelligence applications.**

***Index Terms*—Federated learning (FL), Internet of Things (IoT) intelligence applications, transform-domain features.**

I. Introduction

TO SUPPORT smart building, intelligent transportation, ubiquitous e-healthcare, and smart home [1], massive Internet of Things (IoT) devices, such as sensors, wearable devices, and mobile devices are growing in both power and popularity. Numerous data collected by various IoT devices are key to unlock the potential of artificial intelligence in our daily lives [2], [3]. However, IoT devices often have limited resources and energy supply. To fully unleash the potential of the data sensed by IoT devices, more computation resources are needed for data processing and learning. However, the conventional approach to send data to remote cloud involves high-volume transmissions which can be costly and lead to long delay. Furthermore, privacy can also be a major concern when transporting sensitive data across public networks [4].

Therefore, storing and processing data locally with the assistance of edge servers is more desirable [5], [6]. However, it is a challenge to achieve high processing accuracy by individual edge server, given the limitations of local data sets. Federated learning (FL) provides a framework to train models in a distributed fashion [7] to address the challenge by involving both local processing in edge servers and remote coordination in the cloud data center [8]. Efforts have been focused on designing advanced FL algorithms to achieve a better learning performance, including privacy preservation [9], [10] and learning efficiency [11].

There are, however, significant challenges for FL. First, since each IoT device collects its own local data, statistical heterogeneity is a common issue among the collected data from IoT devices located in different regions with different environments. FL also involves a multitude of edge servers with diverse coverage availability and hardware, such as storage, computational, and communication capabilities. An FL process relies on local models trained by edge servers, consideration of the heterogeneity among edge servers leads to quite different local objective functions corresponding to different local optimums. As a result of these, the local model trained on each edge server may be biased, and in effect the heterogeneity may cause significant local training drifts. Moreover, efforts in the local training can be neutralized in the conventional strategy based on averaging the trained local models which makes the global model very hard to converge. In addition, the exchange of updated models between the edge servers and the central cloud server implies that the time required to tune the global model depends not only on the number of training iterations but also on the delay induced by transmitting the model updates at each FL iteration. Each edge server needs to wait till receiving the global model update and then resumes the training for the next iteration. Here, the communication procedure can be a bottleneck affecting the training time of global models. Clearly, reduction of the communication time will greatly improve the efficiency of the entire training procedure. The work presented in this article is motivated by the points made above, where we focus our investigation on an FL approach to address issues concerning statistical heterogeneity of the model, reduction of communication cost, and ensuring processing accuracy given limited time and computing resources at edges.

Specifically, we propose to explore the training based on the data samples in their transformed domains which may reduce the communication burden of IoT devices and in some cases enhance the reliability and accuracy of federated learned models. The proposed transform-domain FL algorithms include 1-D and 2-D discrete cosine transform with different preserve rates (DCT-FA), discrete wavelet transform with different decomposition levels (DWT-FA), and the

Manuscript received 27 September 2022; accepted 10 November 2022. Date of publication 16 November 2022; date of current version 24 March 2023. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), Compute Canada, and British Columbia Knowledge Development Fund (BCKDF). *(Corresponding author: Lin Cai.)*

The authors are with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8P 5C2, Canada (e-mail: leizhao@uvic.ca; cai@ece.uvic.ca; wslu@ece.uvic.ca).

Digital Object Identifier 10.1109/JIOT.2022.3222842

scheme combines the time-domain features and frequency-domain features (CDCT-FA). Both DCT-FA and DWT-FA are shown to reduce the computation burden in edge servers and communication cost without sacrificing the application model accuracy due to their leading ability to compact the most important information of the raw data samples into fewer features. In addition, we examine a transform-domain scheme, CDCT-FA, which takes both the advantage of time-domain and frequency-domain features, that is, shown to achieve considerably improved processing accuracy, especially, when the data set is of limited volume. We have conducted extensive simulations to evaluate the performance of the transform-domain FL schemes. Our numerical results show that the proposed transform-domain FL algorithms are superior to the popular FL algorithms, in both overall training time and learning accuracy.

The remainder of this article is organized as follows. Related works are provided in Section II. Section III illustrates the system model and formulates the federated training problem. The proposed transform-domain FL algorithms are elaborated in Section IV. The theoretical analysis is provided in Section V. Simulation results are presented in Section VI to illustrate the training efficiency with different settings, followed by the concluding remarks in Section VII.

II. Related Works

Although task offloading, workload scheduling, and service migration for IoT systems have been heavily investigated, IoT intelligence is still in its infancy stage. Pushing the AI frontiers to the individual IoT devices is promising to fully unleash the potential of the zillion bytes of data generated by billions of IoT devices per year. However, IoT intelligence is obtained from the heterogeneous local data sets with the federated setting in the training procedure of IoT devices. Real-world data samples collected by individual IoT devices contain a mixture of many effects, and how to deal with the cross device differences in real-world partitioned data sets for efficient federated training is an important open question.

To tackle the issues of communication cost and delay of FL, it was proposed that clients perform multiple local model updates before communicating with the central server [12]. One of the most popular FL techniques is the Federated Averaging (FedAvg) algorithm [13]. For homogeneous clients, FedAvg coincides with the parallel stochastic gradient descent (SGD) analyzed in [14], and its asymptotic convergence has been proven [15]. Empirically, the FedAvg is found working well when the local data sets are independently identically distributed (IID) and local SGD updates are averaged because because in this case the local gradient provides an unbiased estimate of the global gradient [16].

However, a client may differ from its peers in multiple aspects [7], [17] and statistical heterogeneity is common with data being nonindependent and identically distributed (non-IID) among clients. Many authors have proposed non-IID objective models to address the data variation [7], [18], [19], [20], [21], where the FedAvg is shown to provide substantially degraded performance due to data heterogeneity because with non-IID local data sets the local stochastic gradient becomes a biased estimate of the global gradient. Zhao et al. [22] was among the first to observe the challenges facing FedAvg when dealing with heterogeneous local data. Wang et al. [8], Li et al. [18], and Yu et al. [23] applied the bounded gradients and analyze how it affects the training drift due to the use of non-IID local data. Analysis of the FedAvg that quantifies how data heterogeneity degrades the convergence rate in this scenario can be found in [19], [20], and [21].

To improve the performance of FL with data heterogeneity, FedProx proposed in [7] can be viewed as a generalization and reparametrization of FedAvg by adding a proximal term to local objective functions. Another promising direction to address the challenge arising from data heterogeneity is to apply variance reduction techniques into FL [24]. SCAFFOLD was proposed to use variance reduction to correct the client-drift in its local updates [21]. By adapting an arbitrary centralized optimization algorithm to the cross-device FL setting, MIME is proposed to use a combination of control-variates and momentum at each client-update step to ensure that each local update mimics that of the centralized method running on independent identically distributed data [25].

To reduce the communication cost within each training round and make the collaboration more flexible among dynamic client environment, a line of works assumed that the server can arbitrarily sample a set of clients to collect responses accordingly in every communication round [7], [20], [21], [26]. This stochastic client selection is desirable in many practical scenarios as it can reduce the communication cost in each training round and handle the problem of arbitrary device availability [27].

In this article, our focus is on developing methods to deal with heterogeneous data sets with a synchronous architecture as recommended in [13]. Regarding to the training procedure in IoT devices, we apply periodic decentralized SGD (PD-SGD) updating to carry out the training procedure in each IoT device with multiple local updates [28], [29], [30]. The proposed methods are based on the observation that compressed frequency-domain features are of considerable help in mitigating the heterogeneity of the local data sets and reducing model size as well as the cost of communication between the central cloud and edge servers as well as the communication cost among edge servers and IoT devices, while providing sufficient information of the original data necessary to maintain satisfactory convergence.

III. System Model and Problem Formulation

A. System Model

In this section, we propose an FL framework for edge-enabled IoT intelligence as illustrated in Fig. 1. Massive data are collected by various IoT devices, such as temperature and humidity sensors in smart buildings, road surveillance cameras and carbon dioxide sensors in smart transportation, wearable devices in smart healthcare, and networking home devices in smart homes. However, due to the limited battery life, processing capacity, and storage space, each individual IoT device is

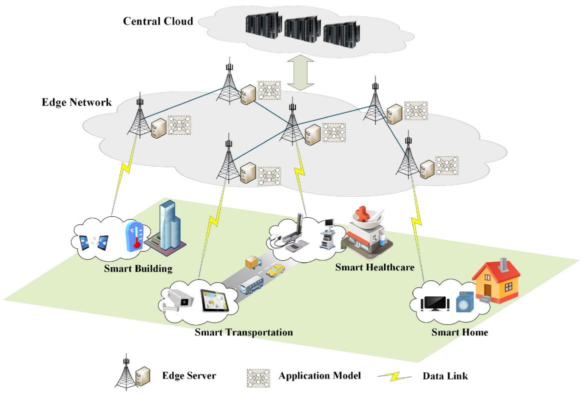


Fig. 1. FL framework for edge-enabled IoT intelligence.

mainly used to collect data and conduct only simple computing tasks. To train the AI models, numerous sensed data need to deliver to edge servers close to the end IoT devices.

The edge server alleviates the training burden on individual IoT devices, however, the coverage of stationary edge server is also limited leading to the limited amount of local training data. If massive data are sent from all edge servers to the central cloud, it will introduce a heavy burden to the network and more privacy concerns. To achieve good performance for supporting AI applications, multiple edge servers cooperate to finish the application model training procedure. We apply the FL framework to conduct the cooperation among edge servers. Each edge server conducts a few training steps on its local model parameters then transports its local model to the central cloud. The central cloud will aggregate all the locally updated models in each round by taking a weighted average of the local model parameters in proportion to the size of local data sets [31] and then distributes the aggregated global model to the edge network for further local parameter updating.

B. Federated Optimization Problem Setup

There are E edge servers denoted as $\{s_i, i = 1, 2, \dots, E\}$. Under the coverage of the service region of edge server s_i, there are a set of IoT devices I_i collecting and delivering data to s_i. The edge servers receive the data from their service region and maintain local data sets to train their local model parameters. We use $\boldsymbol{w}$ to represent the global model parameters and it is shared with all edge servers as the initialization of each local model. We use D_i to denote the local data set for edge server s_i with n_i local training samples for $i = 1, 2, \dots, E$ and the overall sample number $n = \sum_{i=1}^{N} n_i$. There is no overlap among different local data sets, i.e., $D_i \cap D_j = \emptyset$ whenever $i \neq j$. All data samples in the local data set D_i of edge server s_i construct the local objective function $\boldsymbol{f_i}(\boldsymbol{w})$. The optimization problem in an FL objective is formulated as

$$\underset{\boldsymbol{w}}{\text{minimize}} \quad \boldsymbol{f}(\boldsymbol{w}) = \sum_{i=1}^{N} \frac{n_i}{n} f_i(\boldsymbol{w}). \qquad (1)$$

It clearly shows that the central objective function $f(\boldsymbol{w})$ is a convex combination of the local objective $\boldsymbol{f_i}(\boldsymbol{w})$. For the local training procedure in edge server s_i, it tries to minimize its own objective function $\boldsymbol{f_i}(\boldsymbol{w})$ which will lead to a local optimal solution. Due to the heterogeneity of the local training data sets, different local objective functions will be very different from each other. The local training procedure will update the model into different directions which leading it very hard to converge from the global view. Therefore, minimizing $\boldsymbol{f_i}(\boldsymbol{w})$ and average the results cannot provide a promising solution to the global objective $f(\boldsymbol{w})$, unless the local functions are all the same, i.e., local solutions $\boldsymbol{w_i^*}$ for $i = 1, 2, \dots, E$ are all the same which is highly unlikely in practice.

C. Role of Features in Local Objectives

To clearly show the problem of heterogeneity, we formulated the connection of the data features and the local objective function as follows. For generality, we assume the local objective function $\boldsymbol{f_i}(\boldsymbol{w})$ is formulated by one L layer neural network combined with softmax regression loss. The features of the data sample are transmitted to multiple neurons by linearly combining the link weights that connect each input node and the neurons to obtain their preactivation value and compute the post-activation value by the activation function in each neuron. Then, the successive neuron layers feed the post-activation value into one another until the output layer.

We define each layer contains $p_1, p_2, \dots, p_L$ neurons. The post-activation outputs of hidden layers are denoted by L vectors $\boldsymbol{h_1}, \boldsymbol{h_2}, \dots, \boldsymbol{h_L}$ with dimension $p_1, p_2, \dots, p_L$, respectively. The weights between the lth hidden layer and the $(l + 1)$th hidden layer are denoted by a connection matrix $\boldsymbol{W_l} \in R^{p_l \times p_{l+1}}$, and the forward computation part of this transformation between hidden layers are denoted as $\boldsymbol{h_{l+1}} = \Phi(\boldsymbol{W_{l+1}^T h_l})$, where $\Phi(\cdot)$ represents the activation function and $\forall l \in \{1, \dots, L-1\}$. For $(\boldsymbol{x}, y) \in D_i$, the data sample $\boldsymbol{x}$ is fed into the neural network at the input layer, where the connection matrix $\boldsymbol{W_0} \in R^{N \times p_1}$ linearly combining the input features and deliver the result through the activation function to the first hidden layer formulated as $\boldsymbol{h_1} = \Phi(\boldsymbol{W_0^T x})$. We define the connection matrix $\boldsymbol{W_{L+1}} \in R^{p_L \times C}$ connecting the Lth hidden layer and the output layer, the forward computation is formulated as

$$\boldsymbol{o} = \Phi\left(\boldsymbol{W_{L+1}^T} \boldsymbol{\Phi}\left(\boldsymbol{W_L^T}, \dots \boldsymbol{\Phi}\left(\boldsymbol{W_0^T x}\right)\right)\right) \qquad (2)$$

where $\boldsymbol{o} \in R^{C \times 1}$ denotes the output vector in the output layer. Then, by applying the softmax regression loss, the detailed formulation for the local objective function is written as

$$f_i(\boldsymbol{w}) = \frac{1}{n_i} \sum_{(\boldsymbol{x}, y) \in D_i} \log\left(\sum_{j=1}^{C} e^{o_i}\right) - o_y, \quad i = 1, 2, \dots, E \qquad (3)$$

where we use $\boldsymbol{w}$ to represent all the parameters in the neural network for simplicity. We assume that the neural network structures are the same, then it is clearly shown that the features in local data set determines the local objective function. Due to the heterogeneity of the local data samples, the optimal solution of local objective functions are quite different from each other. Since the local training efforts update the model parameters toward their own minimizers, the problem of the data heterogeneity directly shows up.

D. One Round Updating in Federated Optimization

In round r, we sample $S^r \subseteq [E]$ edge servers at random and $|S^r| = S$, the global model $\boldsymbol{w^{r-1}}$ is shared to the selected local parameter $\boldsymbol{\hat{w}^r_{i,0}} = \boldsymbol{w^{r-1}}$ and update the local parameters with local step size α_l for $k \in K$ local iterations

$$\boldsymbol{\hat{w}^r_{i,k}} = \boldsymbol{\hat{w}^r_{i,k-1}} - \alpha_l \boldsymbol{g_i}\left(\boldsymbol{\hat{w}^r_{i,k-1}}\right) \tag{4}$$

$$\boldsymbol{\hat{w}^r_{i,K}} = \boldsymbol{w^{r-1}} - \sum_{k=1}^{K} \alpha_l \boldsymbol{g_i}\left(\boldsymbol{\hat{w}^r_{i,k-1}}\right). \tag{5}$$

The local updating in the ith edge server applies stochastic gradient directions. Let $f_i(\boldsymbol{w}) = E_{\xi_i}[\boldsymbol{f_i}(\boldsymbol{w}; \xi_i)]$, and $\boldsymbol{g_i}(\boldsymbol{w}) = \triangledown \boldsymbol{f_i}(\boldsymbol{w}; \xi_i)$ which is an unbiased gradient estimation of f_i with variance bounded by σ^2. The new global parameters are updated with global step size α_g as

$$\boldsymbol{w^r} = \boldsymbol{w^{r-1}} + \frac{\alpha_g}{S} \sum_{i \in S^r} \left(\boldsymbol{\hat{w}^r_{i,K}} - \boldsymbol{w^{r-1}}\right). \tag{6}$$

We define the effective step-size $\tilde{\alpha} = K\alpha_l\alpha_g$, so that the expectation of server update in round r is written as

$$\boldsymbol{\delta^{r-1}} = -\frac{\tilde{\alpha}}{KS} \sum_{i=1}^{S} \sum_{k=1}^{K} \boldsymbol{g_i}\left(\boldsymbol{\hat{w}^r_{i,k-1}}\right).$$

IV. Transform-Domain Federated Learning

At the beginning of the training, i.e., during the first iteration of the loop, the central cloud initializes the model parameters of the required application and transmits them to the edge network. The training procedure in each edge server minimizes the local loss function formulated by the model and the local training data set, in that the cloud server frequently participates in the training procedure by periodically aggregating the local updated parameters into the global model. Below, we propose the transform-domain FL techniques to improve the training efficiency as well as the computing and communication cost for the local training procedure. The transform-domain federated averaging algorithm and the supervised aggregation scheme are illustrated in Algorithm 1 and Procedure 1, respectively.

Algorithm 1 Transform-Domain Federated Averaging Algorithm

```
Initialize global model parameters $\boldsymbol{w^0}$ in central cloud
Initialize $\{D_i\}_{i=1}^E = \varnothing$ and $\{\tilde{D}_i\}_{i=1}^E = \varnothing$
for $r \leftarrow 1$ to $R$ do
  Distribute global model to the edge network
  $S^r \leftarrow$ random set of $S$ edge servers
  for $i \in S^r$ do
    Initialize local mode $\boldsymbol{\hat{w}^r_{i,0}} = \boldsymbol{w^{r-1}}$
    for $k \leftarrow 1$ to $K$ do
      Sample batch $\{(\boldsymbol{x_j}, y_j)\}_{j=1}^m$ from local data set
      Initialize transform-domain feature batch $\hat{D} = \varnothing$
      for $j \leftarrow 1$ to $m$ do
        if $(\boldsymbol{x_j}, y_j) \notin \tilde{D}_i$ then
          Generate transform-domain features $\boldsymbol{\hat{z}_j}$ of $\boldsymbol{x_j}$
          $D_i \leftarrow D_i \cup (\boldsymbol{\hat{z}_j}, y_j)$ and $\tilde{D}_i \leftarrow \tilde{D}_i \cup (\boldsymbol{x_j}, y_j)$
        end if
        Fetch $(\boldsymbol{\hat{z}_j}, y_j)$ from $D_i$ and $\hat{D}_i \leftarrow \hat{D}_i \cup (\boldsymbol{\hat{z}_j}, y_j)$
      end for
      Compute $\boldsymbol{g_i}(\boldsymbol{\hat{w}^r_{i,k-1}})$ based on $\hat{D}_i$
      $\boldsymbol{\hat{w}^r_{i,k}} = \boldsymbol{\hat{w}^r_{i,k-1}} - \alpha_l \boldsymbol{g_i}(\boldsymbol{\hat{w}^r_{i,k-1}})$
    end for
  end for
  Model aggregation following Supervised Aggregation
  Scheme in Procedure 1.
end for
```

A. Usefulness of Frequency Features

Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two nonzero samples with the same dimension. We define the *cosine similarity* between $\boldsymbol{x}$ and $\boldsymbol{y}$ as

$$S(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x^T y}}{||\boldsymbol{x}||_2 \cdot ||\boldsymbol{y}||_2}.$$

Note that $S(\boldsymbol{x}, \boldsymbol{y})$ is precisely the cosine of the angle between vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, and it is a similarity measure because it is merely the normalized inner product which quantifies the correlation between the two samples. It is known that, to a large extent, the information of the original input data are well represented by a small number of DCT coefficients in the low-frequency region [32]. Utilizing these compressed frequency features is found to have increased cosine similarity relative to that of the original features. For example, for the two images in Fig. 2, the cosine similarity between the bitmap features is 0.9055. If we focus on the first 1000 frequency features out of the entire 132 300 DCT features, the cosine similarity increases dramatically to 0.9802. This provides an intuitive comparison of the original features and the corresponding transfer domain features from different traffic surveillance cameras which are representative IoT devices in practice. Since each local objective function is determined by its local training feature vectors, increasing feature vector similarity makes the landscape of the local objective functions more similar to each other. When local objective functions get more similar, the local gradient drift is reduced along the updating trajectory, which in turn leads to less heterogeneity among different local objective functions. As a result, the use of DCT features is shown to make the training procedure more efficient and, hence, converge faster. Furthermore, the substantially compressed sample features also lead to improved local training speed and reduced communication cost due to reduced model parameter space.

Besides the intuitive explanation, below we provide an analytical argument concerning the usefulness of frequency-domain features. As common practice in optimization, considerable training progress is made in a small number of initial iterations. Thus, most of the training effort is spent in searching the local area near the solution. To conduct a local analysis, assuming the ith local objective function f_i is a β_i-smooth and μ_i-strongly convex function, $\boldsymbol{w}$ is the current global model and $\boldsymbol{w^*}$ is the optimal global model. From Lemma 1 (see the Appendix), we have

$$f_i(\boldsymbol{w}) - f_i\left(\boldsymbol{w^*}\right) \geq \frac{1}{2\beta_i} \left\| \triangledown \boldsymbol{f_i}(\boldsymbol{w}) - \triangledown \boldsymbol{f_i}\left(\boldsymbol{w^*}\right) \right\|^2 + \triangledown \boldsymbol{f_i}\left(\boldsymbol{w^*}\right)^{\boldsymbol{T}} \left(\boldsymbol{w} - \boldsymbol{w^*}\right).$$

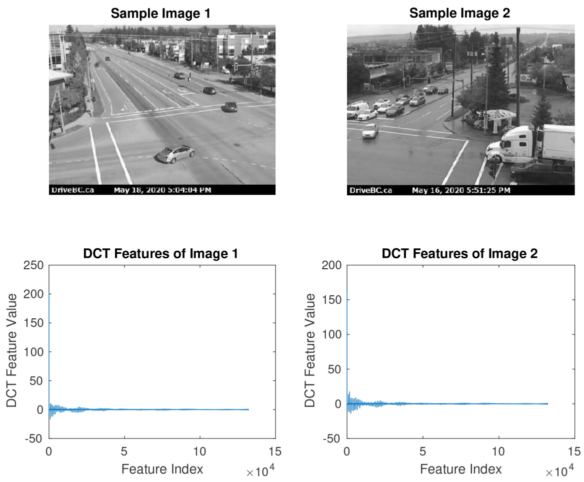


Fig. 2. Examples of DCT features.

Then, we sum over all local objectives and take the average on both sides. Since $\nabla \boldsymbol{f}(\boldsymbol{w}^*) = 0$, we obtain that

$$f(\boldsymbol{w}) - f(\boldsymbol{w}^*) \geq \frac{1}{E}\sum_{i=1}^{E}\frac{1}{2\beta_i}\left\|\nabla \boldsymbol{f_i}(\boldsymbol{w}) - \nabla \boldsymbol{f_i}(\boldsymbol{w}^*)\right\|^{\mathbf{2}}. \quad (7)$$

By defining $\hat{\beta} = \max\{\beta_i\}_{i=1}^{E}$, (7) implies that

$$2\hat{\beta}\left(f(\boldsymbol{w}) - f(\boldsymbol{w}^*)\right) \geq \frac{1}{E}\sum_{i=1}^{E}\left\|\nabla \boldsymbol{f_i}(\boldsymbol{w}) - \nabla \boldsymbol{f_i}(\boldsymbol{w}^*)\right\|^{\mathbf{2}}. \quad (8)$$

Based on (8) and Lemma 2, we bound the local gradients as

$$\begin{aligned}
&\frac{1}{E}\sum_{i=1}^{E}\left\|\nabla \boldsymbol{f_i}(\boldsymbol{w})\right\|^{\mathbf{2}} \\
&\leq \frac{2}{E}\sum_{i=1}^{E}\left\|\nabla \boldsymbol{f_i}(\boldsymbol{w}) - \nabla \boldsymbol{f_i}(\boldsymbol{w}^*)\right\|^{\mathbf{2}} + \frac{2}{E}\sum_{i=1}^{E}\left\|\nabla \boldsymbol{f_i}(\boldsymbol{w}^*)\right\|^{\mathbf{2}} \\
&\leq 4\hat{\beta}\left(f(\boldsymbol{w}) - f(\boldsymbol{w}^*)\right) + \frac{2}{E}\sum_{i=1}^{E}\left\|\nabla \boldsymbol{f_i}(\boldsymbol{w}^*)\right\|^{\mathbf{2}}. \quad (9)
\end{aligned}$$

Utilizing the frequency features, it is possible to make the local objectives more similar to each other. As a result, the optimal global model gets closer to the optimal points of the local objectives, resulting in a smaller average magnitude of the local gradients with respect to $\boldsymbol{w^*}$ defined by B

$$B = \frac{1}{E}\sum_{i=1}^{E}\left\|\nabla \boldsymbol{f_i}(\boldsymbol{w}^*)\right\|^{\mathbf{2}}.$$

From the convergence analysis provided in Section V, it will become clear that a smaller B implies a tighter upper bound of the closeness of the model learned to the optimal model after a given rounds of iterations. To demonstrate the idea of utilizing frequency features to decrease the value of B, we computed B with the MNIST data set, with 10% of the low-frequency DCT features, achieving $B = 4.7106$. This compares favorably with a $B = 55.8658$ when only the original features were used.

B. Discrete Cosine Transform-Based Federated Averaging Algorithm

The discrete cosine transform (DCT) which is known to provide compressed frequency-domain features [32] is employed as a part of our feature engineering to preprocess the data sample at each IoT device. Assuming each data sample has N features, i.e., $\boldsymbol{x} \in R^{N\times 1}$, we transform each sample into a frequency space by defining a frequency-related coordinate system. The number of dimensions indicates the resolution of the spectrum and each dimension in this spectral space represents one frequency we select. To ensure that the frequency-related basis vectors are orthogonal, we set the number of the basis vectors as the same as the feature number of time-domain samples. By designing the frequency-related basis vectors to be orthonormal vectors, the transformation of data sample features from the time domain into the frequency domain is regarded as a projection of the sample on the frequency-related basis vectors.

The frequency-related coordinate system is denoted by $\boldsymbol{U} = [\boldsymbol{u_1}, \boldsymbol{u_2}, \ldots, \boldsymbol{u_N}] \in R^{N\times N}$. Here, we select the DCT as such a system due to its excellent energy compaction and compression ability [32], and, in this case, the kth basis vector is given by

$$\boldsymbol{u_k} = \alpha_k\sqrt{\frac{2}{N}}\left[\cos\left(\frac{1\cdot k\pi}{2N}\right) \quad \cdots \quad \cos\left(\frac{(2N-1)\cdot k\pi}{2N}\right)\right]^T \quad (10)$$

where $k = 0, 1, 2, \ldots, N-1$ and

$$\alpha_k = \begin{cases} \frac{1}{\sqrt{2}}, & k = 0 \\ 1, & k = 1, 2, \ldots, N-1. \end{cases}$$

The orthonormality of frequency basis vectors is guaranteed where the magnitude of other basis vectors are $||\boldsymbol{u_k}||_2^2 = 1$ for $k = 0, 1, 2, \ldots, N-1$ and the basis vectors are also orthogonal to each other $\boldsymbol{u_i^T u_j} = 0$ for $i, j = 0, 1, 2, \ldots, N-1$, and $i \neq j$.

Each basis vector $\boldsymbol{u_i} \in R^{N\times 1}$ combines the N features in the original sample $\boldsymbol{x}$ to generate a spectral feature which is regarded as the projection of sample $\boldsymbol{x}$ on the frequency-domain basis vector $\boldsymbol{u_i^T x}$. Let $z_k = \boldsymbol{u_k^T x}$ be the kth frequency-domain feature, namely,

$$z_k = \boldsymbol{u_k^T x} = \alpha_k\sqrt{\frac{2}{N}}\sum_{n=0}^{N-1} x_n \cos^2\left(\frac{(2n+1)\cdot k\pi}{2N}\right)$$

where $k = 0, 1, 2, \ldots, N-1$. This transformation procedure is directly applied to the raw data, and can be done in IoT devices. We now define a preserve rate p for DCT-based feature extraction as the ratio of the number of the most significant components in $\boldsymbol{z}$ that are to be retained to the length of $\boldsymbol{z}$. Since p is rather small (typical in the range between 0.1 and 0.3), using $N \times p$ most significant components of $\boldsymbol{z}$ as features implies a big reduction of the input space, and, hence, reduced cost in training and communication.

Alternatively, we may apply 2-D DCT on the sensed data in IoT devices. For a 2-D sample $\boldsymbol{X} \in R^{N\times N}$, the 2-D DCT coefficient matrix is regarded as we project all the rows of $\boldsymbol{X}$ into frequency space $\boldsymbol{U}$, i.e., $\boldsymbol{XU}$, and then projecting all the columns of $\boldsymbol{XU}$ into $\boldsymbol{U}$ again, i.e., $\boldsymbol{Z} = \boldsymbol{U^T XU}$. We rearrange the frequency features from the 2-D DCT coefficient matrix $\boldsymbol{Z}$ following the zig-zag searching scheme into a feature vector $\boldsymbol{z}$, and then apply the preserve rate into the feature vector and transmit the select portion to the edge server for local training.

C. Discrete Wavelet Transform-Based Federated Averaging Algorithm

Effective features of reduced dimensionality may also be acquired using discrete wavelet transform (DWT). Here, we apply the Haar wavelet which is characterized by the two-tap low-pass and high-pass filters $\boldsymbol{h_0} = \left[(1/\sqrt{2})\ \ (1/\sqrt{2})\right]^T$ and $\boldsymbol{h_1} = \left[-(1/\sqrt{2})\ \ (1/\sqrt{2})\right]^T$, respectively.

For 1-D DWT in IoT devices with sensed data sample $\boldsymbol{x} \in R^{N\times 1}$, an input sample is projected onto the basis vector $\boldsymbol{h_0}$ with block length 2 when passing through the low-pass filter. We denote the ith feature block as $\hat{\boldsymbol{x}}_{\boldsymbol{i}} \in R^{2\times 1}$, and there are also N feature blocks in total for time domain sample $\boldsymbol{x}$ where there is one feature overlap between every two adjacent feature blocks, e.g., $\hat{\boldsymbol{x}}_{\boldsymbol{i}} = [x_i, x_{i+1}]$ and $\hat{\boldsymbol{x}}_{\boldsymbol{i+1}} = [x_{i+1}, x_{i+2}]$, and there are zero padding for the first feature block $\hat{\boldsymbol{x}}_{\boldsymbol{1}}$ and the last feature block $\hat{\boldsymbol{x}}_{\boldsymbol{N}}$.

The output of the low-pass filter is one vector with length N where the ith term is the result of the inner product of $\boldsymbol{h}_{\boldsymbol{0}}^{\boldsymbol{T}}\hat{\boldsymbol{x}}_{\boldsymbol{i}}$. To eliminate the feature overlap impact, the results from the low-pass filter are downsampled with factor 2, and the result after the downsampling denoted by $\boldsymbol{a_1}$ is called the approximation information of DWT with first-level decomposition. The same procedure is also applied to the high-pass filter whose outputs are called the details of DWT with first-level decomposition.

In DWT-FA, we use the approximation information of the DWT results with different decomposition levels. Note that the approximation features in the first-level decomposition are of length $N/2$, $\boldsymbol{a_1} \in R^{N/2\times 1}$, due to the downsampling. The approximation features from the first-level decomposition are used as the input to the second-level decomposition where the transform is carried out in the same way. First, $\boldsymbol{a_1}$ is fed into the low-pass filter with impulse response $\boldsymbol{h_0}$ whose output is in turn downsampled by factor 2 to obtain the approximation information of the second-level decomposition $\boldsymbol{a_2} \in R^{N/2^2\times 1}$. Clearly, the length of the ith level approximation features is of $N/2^i$.

For 2-D DWT, the input samples are matrices, e.g., $\boldsymbol{X} \in R^{N\times M}$, the transform procedure by passing the low-pass filter with impulse response $\boldsymbol{h_0}$ and down-sampling-by-2 operation will be applied on the row vectors of $\boldsymbol{X}$ first, and then apply all column vectors of the previous results into the same procedure to get the approximation information of the first-level decomposition $\boldsymbol{A_1} \in R^{N/2\times M/2}$. For multiple-level decomposition, the feature block projection and downsampling-by-2 operation are applied to all rows and then all columns of each data sample collected by IoT devices in each level decomposition. Thus, with the same decomposition level, 2-D DWT preserves only half of the size of that in 1-D DWT which reduces more communication time at the cost of possible degradation in training accuracy.

D. Combined Discrete Cosine Transform-Based Federated Averaging Algorithm

The features of the samples sensed by IoT devices may be insufficient to guarantee target training accuracy due to the limited sensing capability of individual IoT devices. Furthermore, the quality of the sensed feature may also be insufficient to make effective local training. Therefore, we propose combined DCT-based federated averaging algorithm (CDCT-FA) which is an extension of the DCT-FA algorithm by combining the time-domain features and the frequency-domain features. Since the primary information carried by the data sample remains after DCT transformation and the frequency features in high frequencies can be removed due to little energy in that part, the additional frequency-domain features are shown to greatly enhance the performance of the FL over the edge network with massive IoT devices without causing much extra training time.

The advantages of more efficient training with DCT or DWT features may be described from two perspectives: first, the transform-domain features are able to increase the sample-vector similarity which leads to more similar local objective functions and, hence, reduced local gradient drift. This results in reduced heterogeneity and faster convergence. Second, the compressed features (by DCT or DWT) lead to reduced dimension of model parameters and, hence, increased training speed and reduced communication cost. Unlike the technique that uses DCT or DWT alone for the sake of dimensionality reduction with compressed features, the CDCT method employs features of increased dimensionality for the reason that the features are now much enriched as they cover both fundamental domains of space (or time) and frequency. The IoT sensors send the original features to the edge servers which further calculate the DCT values for each sample. The CDCT method does not increase the communication cost between the sensors and the edge servers because the sensors only need to send the original feature vectors to the edge servers. When the feature calculation is conducted in the edge servers, CDCT features are obtained by combining the original features and the low-frequency features. When the compressed frequency features are combined with the original data, there is no gain in the mode parameter reduction, however, it yields a data set with several advantages: the combined data contain enriched features from two fundamental and complimentary domains (i.e., the spatial (or time depending on the application) domain and frequency domain) and, hence, is expected to produce improved performance; moreover, data samples combined with low-frequency features are found to have increased cosine similarity relative to that of the original features. The increased cosine similarity of features based on CDCT in the example in Fig. 2 achieved 0.9401 with 1000 low-frequency features in addition to the original features, while the cosine similarity between the original features was found to be 0.9055. This demonstrates that the feature similarity based on CDCT technique is higher than that of the original sample features, but lower than that based only on DCT-compressed features. There is a tradeoff between increasing the feature similarity and the test accuracy. Low feature similarity leads to severe drift in the local updating which becomes even worse in the nonoverlap local data set scenario. High feature similarity mitigates the heterogeneity, however, it may cause some loss of critical information to distinguish different samples. The CDCT features make a tradeoff

Algorithm 1 Supervised Aggregation Scheme

```
Initialize current best global model parameters $\boldsymbol{w}_{\boldsymbol{g}}^{\boldsymbol{*}}$ and the corresponding valid accuracy $\psi_{g^*}$.
for $t \leftarrow 1$ to $T$ do
    if $t < \psi_r$ then
        Weighted aggregate $\boldsymbol{w} \leftarrow \sum_{i=1}^{E} \rho_i \hat{\boldsymbol{w}}_{\boldsymbol{i}}$
        Evaluate the valid accuracy $\psi_g$ of $\boldsymbol{w}$
        if $\psi_{g^*} < \psi_g$ then
            Update $\boldsymbol{w^*} \leftarrow \boldsymbol{w}$ and $\psi_{g^*} \leftarrow \psi_g$
        end if
    else
        Weighted aggregate $\boldsymbol{w} \leftarrow \sum_{i=1}^{E} \rho_i \hat{\boldsymbol{w}}_{\boldsymbol{i}}$
        Evaluate the valid accuracy $\psi_g$ of $\boldsymbol{w}$
        if $\psi_{g^*} < \psi_g$ then
            Update $\boldsymbol{w}_{\boldsymbol{g}}^{\boldsymbol{*}} \leftarrow \boldsymbol{w}$ and $\psi_{g^*} \leftarrow \psi_g$
        end if
        if $\psi_g \le \psi_a$ then
            Replace global model by $\boldsymbol{w} \leftarrow \boldsymbol{w}_{\boldsymbol{g}}^{\boldsymbol{*}}$
        end if
    end if
end for
```

between the two, which appears to be the reason of its higher accuracy.

E. Supervised Aggregation Scheme

The supervised aggregation scheme we propose is shown in Procedure 1. To control the model flipping from traditional average aggregation mode to the supervised aggregation mode, we define two parameters, namely, the accuracy threshold ψ_a and the round threshold ψ_r. The round threshold ψ_r refers to the round number from where the supervised aggregation mode starts. Before ψ_rth round, the locally trained model parameters are aggregated by weighted average. After the ψ_rth training round, we record the current best global model $\boldsymbol{w}_{\boldsymbol{g}}^{\boldsymbol{*}}$. At each training round, first, the local models are aggregated by the weighted average, then the validate accuracy achieved by the current weighted aggregated global model is compared with the predefined accuracy threshold ψ_a: if the performance is lower than the accuracy threshold ψ_a, then the current global model is replaced by the recorded best performance global model $\boldsymbol{w}_{\boldsymbol{g}}^{\boldsymbol{*}}$; otherwise, the current aggregated global model will be distributed to the edge network for the next training round.

V. Convergence Analysis

The performance improvement from the $(r-1)$th round to the rth round is measured by

$$E\left\|\boldsymbol{w^r} - \boldsymbol{w^*}\right\|^{\boldsymbol{2}} = E\left\|\boldsymbol{w^{r-1}} - \boldsymbol{w^*}\right\|^{\boldsymbol{2}} + 2E\left(\boldsymbol{w^{r-1}} - \boldsymbol{w^*}\right)^{\boldsymbol{T}} \boldsymbol{\delta^{r-1}} + E\left\|\boldsymbol{\delta^{r-1}}\right\|^{\boldsymbol{2}} \qquad (11)$$

where $\boldsymbol{w^*}$ denotes the global optimal solution. We need to upper bound the terms $(\boldsymbol{w^{r-1}} - \boldsymbol{w^*})^{\boldsymbol{T}} \boldsymbol{\delta^{r-1}}$ and $||\boldsymbol{\delta^{r-1}}||^{\boldsymbol{2}}$ to estimate the improvement that one round provides. First, we use the updating rule to write

$$E\left[\left(\boldsymbol{w^{r-1}} - \boldsymbol{w^*}\right)^{\boldsymbol{T}} \boldsymbol{\delta^{r-1}}\right] = -\frac{\tilde{\alpha}}{KE} \sum_{i=1}^{E} \sum_{k=1}^{K} \nabla \boldsymbol{f_i}\left(\hat{\boldsymbol{w}}_{\boldsymbol{i,k-1}}^{\boldsymbol{r}}\right)^{\boldsymbol{T}} \left(\boldsymbol{w^{r-1}} - \boldsymbol{w^*}\right).$$

Assuming that the local objective function f_i is also μ_i-strongly convex. Then, Lemma 4 (see the Appendix) implies that

$$\nabla \boldsymbol{f_i}\left(\hat{\boldsymbol{w}}_{\boldsymbol{i,k-1}}^{\boldsymbol{r}}\right)^{\boldsymbol{T}} \left(\boldsymbol{w^{r-1}} - \boldsymbol{w^*}\right) \ge f_i\left(\boldsymbol{w^{r-1}}\right) - f_i\left(\boldsymbol{w^*}\right) + \frac{\mu_i}{4}\left\|\boldsymbol{w^{r-1}} - \boldsymbol{w^*}\right\|^{\boldsymbol{2}} - \beta_i\left\|\hat{\boldsymbol{w}}_{\boldsymbol{i,k-1}}^{\boldsymbol{r}} - \boldsymbol{w^{r-1}}\right\|^{\boldsymbol{2}}.$$

By defining $\bar{\mu} = (1/E)\sum_{i=1}^{E} \mu_i$ and $\bar{\beta} = (1/E)\sum_{i=1}^{E} \beta_i$, we obtain

$$E\left[\left(\boldsymbol{w^{r-1}} - \boldsymbol{w^*}\right)^{\boldsymbol{T}} \boldsymbol{\delta^{r-1}}\right] \le -\tilde{\alpha}\left(f\left(\boldsymbol{w^{r-1}}\right) - f\left(\boldsymbol{w^*}\right) + \frac{\bar{\mu}}{4}\left\|\boldsymbol{w^{r-1}} - \boldsymbol{w^*}\right\|^{\boldsymbol{2}}\right) + \tilde{\alpha}\bar{\beta}\varepsilon \qquad (12)$$

where

$$\varepsilon = \frac{1}{KE} \sum_{i=1}^{E} \sum_{k=1}^{K} \left\|\hat{\boldsymbol{w}}_{\boldsymbol{i,k-1}}^{\boldsymbol{r}} - \boldsymbol{w^{r-1}}\right\|^{\boldsymbol{2}}$$

represents the drift of the local model from the current global model.

A. Averaged Local Model Drift in One Round

Below, we analyze the upper bound of ε. According to the local updating and Lemma 2, we have

$$\left\|\hat{\boldsymbol{w}}_{\boldsymbol{i,k}}^{\boldsymbol{r}} - \boldsymbol{w^{r-1}}\right\|^{\boldsymbol{2}} \le (1+a)\left\|\hat{\boldsymbol{w}}_{\boldsymbol{i,k-1}}^{\boldsymbol{r}} - \boldsymbol{w^{r-1}}\right\|^{2} + \left(1 + \frac{1}{a}\right)\alpha_l^2 \left\|\boldsymbol{g_i}\left(\hat{\boldsymbol{w}}_{\boldsymbol{i,k-1}}^{\boldsymbol{r}}\right)\right\|^{\boldsymbol{2}}. \qquad (13)$$

Since the local updating is stochastic, and we have defined the variance from the sampled gradient to the full local gradient as σ^2, we have $\boldsymbol{E}||\boldsymbol{g_i(w)}||^{\boldsymbol{2}} = ||\nabla \boldsymbol{f_i(w)}||^{\boldsymbol{2}} + \sigma^2$, which in conjunction with (13) leads to an upper bound of the expectation as

$$\boldsymbol{E}||\hat{\boldsymbol{w}}_{\boldsymbol{i,k}}^{\boldsymbol{r}} - \boldsymbol{w^{r-1}}||^{\boldsymbol{2}} \le \left(1 + \frac{1}{K-1}\right)\boldsymbol{E}||\hat{\boldsymbol{w}}_{\boldsymbol{i,k-1}}^{\boldsymbol{r}} - \boldsymbol{w^{r-1}}||^{2} + K\alpha_l^2||\nabla \boldsymbol{f_i}\left(\hat{\boldsymbol{w}}_{\boldsymbol{i,k-1}}^{\boldsymbol{r}}\right)||^{\boldsymbol{2}} + K\alpha_l^2\sigma^2 \qquad (14)$$

where $a = (1/K - 1)$. To deal with the term $\nabla \boldsymbol{f_i}(\hat{\boldsymbol{w}}_{\boldsymbol{i,k-1}}^{\boldsymbol{r}})$ in (14), first we apply Lemma 2 to write

$$\left\|\nabla f_i\left(\hat{\boldsymbol{w}}_{\boldsymbol{i,k-1}}^{\boldsymbol{r}}\right)\right\|^{\boldsymbol{2}} \le 2\left\|\nabla f_i\left(\hat{\boldsymbol{w}}_{\boldsymbol{i,k-1}}^{\boldsymbol{r}}\right) - \nabla \boldsymbol{f_i}\left(\boldsymbol{w^{r-1}}\right)\right\|^{\boldsymbol{2}} + 2\left\|\nabla \boldsymbol{f_i}\left(\boldsymbol{w^{r-1}}\right)\right\|^{\boldsymbol{2}}.$$

Next, we use the fact that function f_i has Lipschitz continuous gradient to bound the drift of the local gradient as

$$\left\|\nabla \boldsymbol{f_i}\left(\hat{\boldsymbol{w}}_{\boldsymbol{i,k-1}}^{\boldsymbol{r}}\right) - \nabla \boldsymbol{f_i}\left(\boldsymbol{w^{r-1}}\right)\right\|^{\boldsymbol{2}} \le \beta_i^2\left\|\hat{\boldsymbol{w}}_{\boldsymbol{i,k-1}}^{\boldsymbol{r}} - \boldsymbol{w^{r-1}}\right\|^{\boldsymbol{2}}.$$

It now follows that:

$$\begin{aligned}&\boldsymbol{E}\left\|\hat{\boldsymbol{w}}_{\boldsymbol{i},\boldsymbol{k}}^{\boldsymbol{r}}-\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}\right\|^{\mathbf{2}}\\ &\quad\leq\left(1+\frac{1}{K-1}+2K\alpha_l^2\beta_i^2\right)\boldsymbol{E}\left\|\hat{\boldsymbol{w}}_{\boldsymbol{i},\boldsymbol{k}-\mathbf{1}}^{\boldsymbol{r}}-\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}\right\|^2\\ &\qquad+2K\alpha_l^2\left\|\nabla\boldsymbol{f}_{\boldsymbol{i}}\left(\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}\right)\right\|^{\mathbf{2}}+K\alpha_l^2\sigma^2.\end{aligned}$$

To upper bound the drift over K local updates, we unroll the recursion from $\hat{\boldsymbol{w}}_{\boldsymbol{i},\mathbf{0}}^{\boldsymbol{r}}$ to $\hat{\boldsymbol{w}}_{\boldsymbol{i},\boldsymbol{k}-\mathbf{1}}^{\boldsymbol{r}}$. Since $\hat{\boldsymbol{w}}_{\boldsymbol{i},\mathbf{0}}^{\boldsymbol{r}}=\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}$, we have

$$\begin{aligned}&\boldsymbol{E}\left\|\hat{\boldsymbol{w}}_{\boldsymbol{i},\boldsymbol{K}}^{\boldsymbol{r}}-\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}\right\|^{\mathbf{2}}\\ &\quad\leq\sum_{k=0}^{K-1}\left(1+\frac{1}{K-1}+2K\alpha_l^2\beta_i^2\right)^k\\ &\qquad\times\left(2K\alpha_l^2\left\|\nabla\boldsymbol{f}_{\boldsymbol{i}}\left(\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}\right)\right\|^{\mathbf{2}}+K\alpha_l^2\sigma^2\right)\end{aligned}$$

which involves a geometric series. This upper bound can be also written as

$$\boldsymbol{E}\left\|\hat{\boldsymbol{w}}_{\boldsymbol{i},\boldsymbol{K}}^{\boldsymbol{r}}-\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}\right\|^{\mathbf{2}}\leq q\left(2K\alpha_l^2\left\|\nabla\boldsymbol{f}_{\boldsymbol{i}}\left(\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}\right)\right\|^{\mathbf{2}}+K\alpha_l^2\sigma^2\right)$$

where q is a constant with fixed local learning rate α_l and local updating iterations K defined as

$$q=\frac{1-\left(1+\frac{1}{K-1}+2K\alpha_l^2\beta_i^2\right)^K}{1-\left(1+\frac{1}{K-1}+2K\alpha_l^2\beta_i^2\right)}.$$

Consequently, the average drift over E clients is upper bounded by

$$\varepsilon\leq\frac{1}{E}\sum_{i=1}^{E}q\left(2K\alpha_l^2\left\|\nabla\boldsymbol{f}_{\boldsymbol{i}}\left(\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}\right)\right\|^{\mathbf{2}}+K\alpha_l^2\sigma^2\right).$$

Note that from (9), we upper bound ε as

$$\varepsilon\leq 8qK\alpha_l^2\hat{\beta}\left(f\left(\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}\right)-f\left(\boldsymbol{w}^{\boldsymbol{*}}\right)\right)+qK\alpha_l^2\left(4B+\sigma^2\right).\quad(15)$$

B. Global Model Improvement in One Round

By applying Lemma 3, we bound the expectation of $||\boldsymbol{\delta}^{\boldsymbol{r}-\mathbf{1}}||^{\mathbf{2}}$ as

$$E\left[\left\|\boldsymbol{\delta}^{\boldsymbol{r}-\mathbf{1}}\right\|^{\mathbf{2}}\right]\leq\left\|\frac{\tilde{\alpha}}{KE}\sum_{i=1}^{E}\sum_{k=1}^{K}\nabla\boldsymbol{f}_{\boldsymbol{i}}\left(\hat{\boldsymbol{w}}_{\boldsymbol{i},\boldsymbol{k}-\mathbf{1}}^{\boldsymbol{r}}\right)\right\|^2+\frac{\tilde{\alpha}^2\sigma^2}{KE}\quad(16)$$

and using Lemma 2, we have

$$\begin{aligned}&\left\|\frac{\tilde{\alpha}}{KE}\sum_{i=1}^{E}\sum_{k=1}^{K}\nabla\boldsymbol{f}_{\boldsymbol{i}}\left(\hat{\boldsymbol{w}}_{\boldsymbol{i},\boldsymbol{k}-\mathbf{1}}^{\boldsymbol{r}}\right)\right\|^2\\ &\quad\leq 2\left\|\frac{\tilde{\alpha}}{KE}\sum_{i=1}^{E}\sum_{k=1}^{K}\left(\nabla\boldsymbol{f}_{\boldsymbol{i}}\left(\hat{\boldsymbol{w}}_{\boldsymbol{i},\boldsymbol{k}-\mathbf{1}}^{\boldsymbol{r}}\right)-\nabla\boldsymbol{f}_{\boldsymbol{i}}\left(\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}\right)\right)\right\|^{\mathbf{2}}\\ &\qquad+2\left\|\frac{\tilde{\alpha}}{E}\sum_{i=1}^{E}\nabla\boldsymbol{f}_{\boldsymbol{i}}\left(\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}\right)\right\|^{\mathbf{2}}.\end{aligned}\quad(17)$$

Then, by Jensen's inequality and $\{\beta_i\}_{1=1}^{E}$ smoothing

$$\begin{aligned}&\left\|\frac{\tilde{\alpha}}{KE}\sum_{i=1}^{E}\sum_{k=1}^{K}\left(\nabla\boldsymbol{f}_{\boldsymbol{i}}\left(\hat{\boldsymbol{w}}_{\boldsymbol{i},\boldsymbol{k}-\mathbf{1}}^{\boldsymbol{r}}\right)-\nabla\boldsymbol{f}_{\boldsymbol{i}}\left(\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}\right)\right)\right\|^{\mathbf{2}}\\ &\quad\leq\frac{\tilde{\alpha}}{KE}\sum_{i=1}^{E}\sum_{k=1}^{K}\left\|\nabla\boldsymbol{f}_{\boldsymbol{i}}\left(\hat{\boldsymbol{w}}_{\boldsymbol{i},\boldsymbol{k}-\mathbf{1}}^{\boldsymbol{r}}\right)-\nabla\boldsymbol{f}_{\boldsymbol{i}}\left(\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}\right)\right\|^{\mathbf{2}}\\ &\quad\leq\frac{\tilde{\alpha}}{KE}\sum_{i=1}^{E}\sum_{k=1}^{K}\beta_i^2\left\|\hat{\boldsymbol{w}}_{\boldsymbol{i},\boldsymbol{k}-\mathbf{1}}^{\boldsymbol{r}}-\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}\right\|^{\mathbf{2}}=\tilde{\alpha}||\boldsymbol{\beta}||^{\mathbf{2}}\varepsilon\end{aligned}$$

where $\boldsymbol{\beta}=\{\beta_i\}_{1=1}^{E}$. Now, using Jensen's inequality again and (9), we have

$$\begin{aligned}&\left\|\frac{\tilde{\alpha}}{E}\sum_{i=1}^{E}\nabla\boldsymbol{f}_{\boldsymbol{i}}\left(\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}\right)\right\|^{\mathbf{2}}\leq\frac{\tilde{\alpha}^2}{E}\sum_{i=1}^{E}\left\|\nabla\boldsymbol{f}_{\boldsymbol{i}}\left(\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}\right)\right\|^{\mathbf{2}}\\ &\quad\leq 4\tilde{\alpha}^2\hat{\beta}\left(f\left(\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}\right)-f\left(\boldsymbol{w}^{\boldsymbol{*}}\right)\right)+2\tilde{\alpha}^2B\end{aligned}$$

which leads us to

$$\begin{aligned}E\left[\left\|\boldsymbol{\delta}^{\boldsymbol{r}-\mathbf{1}}\right\|^{\mathbf{2}}\right]&\leq 2\tilde{\alpha}||\boldsymbol{\beta}||^{\mathbf{2}}\varepsilon+4\tilde{\alpha}^2B\\ &\quad+8\hat{\beta}\tilde{\alpha}^2\left(f\left(\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}\right)-f\left(\boldsymbol{w}^{\boldsymbol{*}}\right)\right)+\frac{\tilde{\alpha}^2\sigma^2}{KE}.\end{aligned}$$

Synthesizing the above analysis, the improvement provided by the proposed technique in one round is estimated by the upper bound of $E[||\boldsymbol{w}^{\boldsymbol{r}}-\boldsymbol{w}^{\boldsymbol{*}}||^{\mathbf{2}}]$

$$\begin{aligned}E\left[\left\|\boldsymbol{w}^{\boldsymbol{r}}-\boldsymbol{w}^{\boldsymbol{*}}\right\|^{\mathbf{2}}\right]&\leq\left(1-\frac{\tilde{\alpha}\bar{\mu}}{4}\right)E\left\|\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}-\boldsymbol{w}^{\boldsymbol{*}}\right\|^{\mathbf{2}}\\ &\quad+c_2E\left(f\left(\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}\right)-f\left(\boldsymbol{w}^{\boldsymbol{*}}\right)\right)+4\tilde{\alpha}(c_1+\tilde{\alpha})B+\left(c_1+\frac{\tilde{\alpha}^2}{KE}\right)\sigma^2\end{aligned}\quad(18)$$

where

$$\begin{aligned}c_1&=qK\alpha_l^2\left(\bar{\beta}+2||\boldsymbol{\beta}||^{\mathbf{2}}\right)\\ c_2&=8\hat{\beta}\tilde{\alpha}(\tilde{\alpha}+c_1)-\tilde{\alpha}.\end{aligned}$$

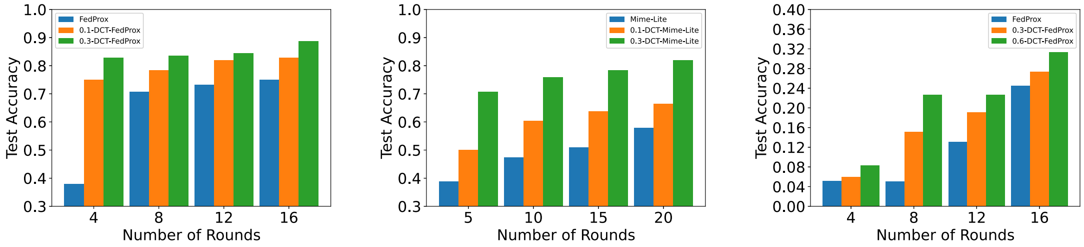
C. Convergence Analysis in Multiple Rounds

By rearranging the terms in (18), we obtain

$$\begin{aligned}E\left[f\left(\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}\right)-f\left(\boldsymbol{w}^{\boldsymbol{*}}\right)\right]&\leq E\left[\frac{1}{c_2}\left(1-\frac{\tilde{\alpha}\bar{\mu}}{4}\right)\left\|\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}-\boldsymbol{w}^{\boldsymbol{*}}\right\|^{\mathbf{2}}\right.\\ &\quad\left.-\frac{1}{c_2}\left\|\boldsymbol{w}^{\boldsymbol{r}}-\boldsymbol{w}^{\boldsymbol{*}}\right\|^{\mathbf{2}}\right]+\frac{4\tilde{\alpha}}{c_2}(c_1+\tilde{\alpha})B+\left(c_1+\frac{\tilde{\alpha}^2}{KE}\right)\frac{\sigma^2}{c_2}.\end{aligned}$$

Applying a weighted summation to the above inequality and letting $\lambda_r=(1-(\bar{\mu}\tilde{\alpha}/4))^{1-r}$ and $\Lambda_R=\sum_{r=1}^{R+1}\lambda_r$, we obtain

$$\begin{aligned}&E\left[\left(1-\frac{\tilde{\alpha}\bar{\mu}}{4}\right)\left\|\boldsymbol{w}^{\boldsymbol{r}-\mathbf{1}}-\boldsymbol{w}^{*}\right\|^{\mathbf{2}}-\left\|\boldsymbol{w}^{\boldsymbol{r}}-\boldsymbol{w}^{*}\right\|^{\mathbf{2}}\right]\\ &\quad=\frac{\left(1-\frac{\tilde{\alpha}\bar{\mu}}{4}\right)\left\|\boldsymbol{w}^{\mathbf{0}}-\boldsymbol{w}^{*}\right\|^{\mathbf{2}}}{\Lambda_R}-\frac{\left(1-\frac{\tilde{\alpha}\bar{\mu}}{4}\right)^{-R}\left\|\boldsymbol{w}^{\boldsymbol{R}+\mathbf{1}}-\boldsymbol{w}^{*}\right\|^{\mathbf{2}}}{\Lambda_R}.\end{aligned}$$



(a) (b) (c)

Fig. 3. Advantage of frequency features over different algorithms and different data sets. (a) FedProx with EMNIST. (b) Mime-Lite with EMNIST. (c) FedProx with CIFAR-100.

By choosing the step size $\tilde{\alpha}$ from the region $(0, (4/\bar{\mu})]$, we reach the upper bound

$$E\left[\left(1-\frac{\tilde{\alpha}\bar{\mu}}{4}\right)\left\|\boldsymbol{w}^{\boldsymbol{r-1}}-\boldsymbol{w}^{\boldsymbol{*}}\right\|^{\mathbf{2}}-\left\|\boldsymbol{w}^{\boldsymbol{r}}-\boldsymbol{w}^{*}\right\|^{\mathbf{2}}\right] \le \frac{\left(1-\frac{\tilde{\alpha}\bar{\mu}}{4}\right)\left\|\boldsymbol{w}^{\mathbf{0}}-\boldsymbol{w}^{\boldsymbol{*}}\right\|^{\mathbf{2}}}{\Lambda_R}. \tag{19}$$

If we set the federated training round $R \ge (4/\tilde{\alpha}\bar{\mu})$, we obtain a lower bound for the summed weights Λ_R in R rounds

$$\Lambda_R \ge 4\left(1-\frac{\tilde{\alpha}\bar{\mu}}{4}\right)^{-R}\frac{1-e^{-R\frac{\tilde{\alpha}\bar{\mu}}{4}}}{\bar{\mu}\tilde{\alpha}}.$$

Since $(R\mu\tilde{\alpha}/4) \ge 1$ and $e^{-1} < (2/3)$

$$\Lambda_R \ge \frac{4\left(1-\frac{\tilde{\alpha}\bar{\mu}}{4}\right)^{-R}}{3\mu\tilde{\alpha}}$$

which leads the estimate in (19) to

$$E\left[\left(1-\frac{\tilde{\alpha}\bar{\mu}}{4}\right)\left\|\boldsymbol{w}^{\boldsymbol{r-1}}-\boldsymbol{w}^{*}\right\|^{\mathbf{2}}-\left\|\boldsymbol{w}^{\boldsymbol{r}}-\boldsymbol{w}^{*}\right\|^{\mathbf{2}}\right] \le \frac{3\tilde{\alpha}\bar{\mu}}{4}\left(1-\frac{\tilde{\alpha}\bar{\mu}}{4}\right)^{R+1}\left\|\boldsymbol{w}^{\mathbf{0}}-\boldsymbol{w}^{*}\right\|^{\mathbf{2}}$$

and, thus

$$E\left[f\left(\boldsymbol{w}^{\boldsymbol{R}}\right)-f\left(\boldsymbol{w}^{\boldsymbol{*}}\right)\right] \le \frac{3\tilde{\alpha}\bar{\mu}e^{-\frac{(R+1)\bar{\mu}\tilde{\alpha}}{4}}}{4c_2}\left\|\boldsymbol{w}^{\mathbf{0}}-\boldsymbol{w}^{*}\right\|^{\mathbf{2}} + \frac{4\tilde{\alpha}}{c_2}(c_1+\tilde{\alpha})B+\left(c_1+\frac{\tilde{\alpha}^2}{KE}\right)\frac{\sigma^2}{c_2}. \tag{20}$$

From (20), we conclude that the quality of model $\boldsymbol{w}^{\boldsymbol{R}}$ measured by its closeness to the optimal $\boldsymbol{w}^{\boldsymbol{*}}$ in terms of the mean $E[f(\boldsymbol{w}^{\boldsymbol{R}})-f(\boldsymbol{w}^{\boldsymbol{*}})]$ is determined by three factors, namely, R: the number of rounds, B: the average magnitude of local gradients at global model $\boldsymbol{w}^{\boldsymbol{*}}$, and σ^2: the variance from the sampled gradient to the full local gradient. The upper bound provided in (20) clearly indicates that model $\boldsymbol{w}^{\boldsymbol{R}}$ is expected to be satisfactory if R is sufficiently large so as to keep the first term of the upper bound small, and both B and σ^2 are small to keep the other two terms of the bound reasonably small as well.

In summary, since the transform-domain features can increase the similarity among local objective functions, namely, the landscape of local objective functions tend to be more similar to each other and the local optimizers get much closer to each other. Therefore, the drift in the local models during the federated training procedure can be reduced, which in turn improves the convergence.

VI. Simulation Results and Discussion

In this section, the proposed method with different SOTA algorithms are examined by applying them to several popular data sets. First we provide a case study to demonstrate the advantages of the proposed frequency features in FL compared with FedProx and Mime-Lite, where the algorithms are implemented with federated-EMNIST and federated-CIFAR-100 data sets. Due to the different complexity of these two data sets, we have applied different local models in the federated training.

Next, we verify the usefulness of the proposed method in various scenarios with a case study via FedAvg. We proposed various the transform-domain FL schemes based on FedAvg, i.e., one-dimensional DCT (1D-DCT-FA) and the combination with time-domain features (1D-CDCT-FA); two-dimensional DCT (2D-DCT-FA) and the combination with the time-domain and frequency-domain features (2D-CDCT-FA); one-dimensional DWT (1D-DWT-FA); and two-dimensional DWT (2D-DWT-FA). For both DCT and DWT-based techniques, in addition to the preserve rate there are several parameters that are adjustable (e.g., the decomposition levels of DWT) to meet specific requirements in applications.

A. Advantage of Frequency Features

First, the proposed transform-domain technique was incorporated into the FedProx algorithm which behaves practically the same way as FedAvg except that it includes a proximal regularization term that prevents clients from drifting too far from the global model. The test accuracy results in comparison with the original FedProx and the frequency-feature-enabled FedProx when applied to the federated EMNIST data set are shown in Fig. 3(a). For FedProx, the proximal strength parameter, which controls the regularization level, was set to 0.1. The local model used, here, was one simple neural network with one hidden layer and softmax regression loss, which was optimized using Adam in updating the local model, where the learning rate was set to 0.02. For the Adam local optimizer, the decay for tracking previous gradients and their second moments was set to 0.9 and 0.999, respectively. For each iteration at an edge server in the local training procedure, the batch size was set to 20 and there were five local training epochs in one federated training round. The results shown in Fig. 3(a) were obtained after four rounds of federated training, where 0.1-DCT-FedProx and 0.3-DCT-FedProx are meant to utilize only 10% and 30% of frequency features. The proposed method was also integrated into the Mime-Lite algorithm with the same setting (except that the local learning rate was set to 0.001), and the results obtained are shown in Fig. 3(b). Clearly, in both cases, the results have demonstrated the ability of the proposed method to quickly converge to fairly accurate solutions with 70%–90% reduction of the input dimensionality.

In addition, the proposed technique was also applied to the CIFAR-100, a more complex data set in terms of the number of categories. In this case study, our method also outperforms the original FedProx. Since there are 100 different classes in CIFAR-100, a relatively more complex local model known as one VGG block [33] was employed to track the convergence trends. As shown in Fig. 3(c), our proposed frequency feature-enabled FedProx converges faster than the original FedProx when applied to the CIFAR-100.

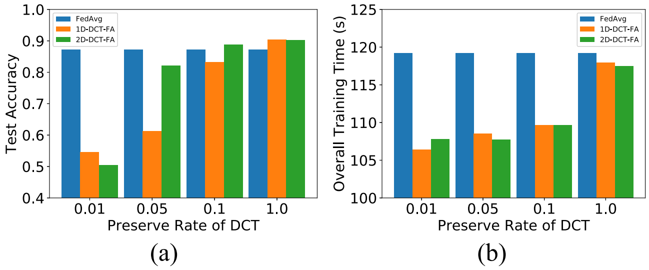


Fig. 4. Performance of 1D-DCT and 2D-DCT-based FL of different preserve rate with IID local data sets. (a) Test accuracy. (b) Overall training time.

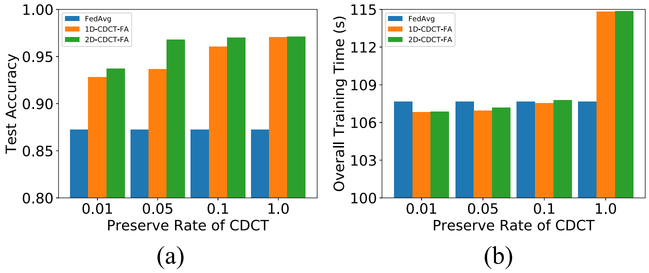


Fig. 5. Performance of 1D-CDCT and 2D-CDCT-based FL of different preserve rates with IID local data sets. (a) Test accuracy. (b) Overall training time.

B. Case Study With FedAvg

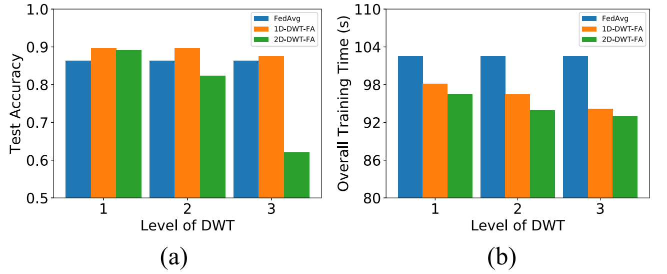
To investigate the diversified data distribution over IoT devices covered by different edge servers, we divided the MNIST [34] data sets and allocated them to randomly selected ten edge servers. The local model applied the one-hidden layer neural network with softmax loss function. The performance of the proposed transform-domain FL schemes was compared with FedAvg over different heterogeneous level of local data sets to illustrate the advantages and robustness of the proposed schemes over the FedAvg in various IoT application scenarios.

For local training, the batch size was set as 0.5 of the local training data set and each edge server only conducted one epoch in each round and there were 200 training rounds for all following experiments. In this case study, we applied SGD algorithm as the local training optimizer. For all DCT-FA except for the measurement of the impact of different preserve rates, the preserve rate was set to 0.2. For all DWT-FA except for the measurement of the impact of different decomposition levels, we applied the first-level DWT into the FL procedure. The train–test rate which means the percentage of training data size in the test data size was set to 10 except the experiment measures the impact of the train–test rate.

1) Simulation Results and Analysis With IID Local Data Sets: First, we measured the performance of the DCT transform-domain FL. Fig. 4 shows the test accuracy of FedAvg, 1D-DCT-FA, and 2D-DCT-FA with the increasing preserve rate. From the results shown in Fig. 4(a), the performance of both 1D-DCT-FA and 2D-DCT-FA is improved rapidly with the increasing preserve rate at the beginning. When the preserve rate reaches 0.05, 2D-DCT-FA achieved good performance and the improvement of the performance slows down compared to that with the original features when the preserve rate is 0.1. When the preserve rate is smaller than 1, 1D-DCT-FA has much worse performance than that with 2D-DCT-FA. We conclude that 2D-DCT-FA needs less information, i.e., features in the frequency domain, to obtain a comparable performance. The performance of both 1D-DCT-FA and 2D-DCT-FA are comparative to the performance of FedAvg after the preserve rate achieving 0.1. The advantages of 1D-DCT-FA and 2D-DCT-FA are shown in Fig. 4(b). Both 1D-DCT-FA and 2D-DCT-FA need much less overall training time to achieve a comparable performance which is shown to be promising property with time-sensitive IoT applications.

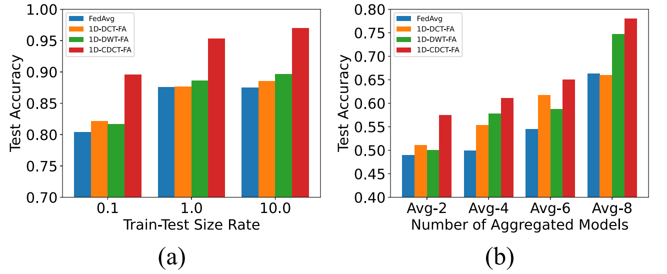
Furthermore, when combining the time-domain features and the frequency-domain ones, the test accuracy achieved for both 1D-CDCT-FA and 2D-CDCT-FA is much better than that of FedAvg even with a very low preserve rate as shown in Fig. 5(a) where 1D-CDCT-FA improves 6% and 2D-CDCT-FA improves 7% of the test accuracy compared with FedAvg. It also shows that 2D-CDCT-FA needs less information to converge to a better test accuracy compared with 1D-CDCT-FA. Furthermore, the overall training time of both 1D-CDCT-FA and 2D-CDCT-FA are also comparative to the overall training time consumed by FedAvg with a low preserve rate as shown in Fig. 5(b). According to the improvement of accuracy performance, this additional overall training time cost is quite tolerable. We conclude that when combining a few frequency-domain features to the time-domain features, the testing performance is greatly improved.

As illustrated in Fig. 6, we measured the performance of 1D-DWT-FA and 2D-DWT-FA with different decomposition levels and we extracted the approximation information at each level for training. There were less information at a higher decomposition level. The results from Fig. 6(a) show that both 1D-DWT-FA and 2D-DWT-FA achieve their best performance with first-level decomposition, and with



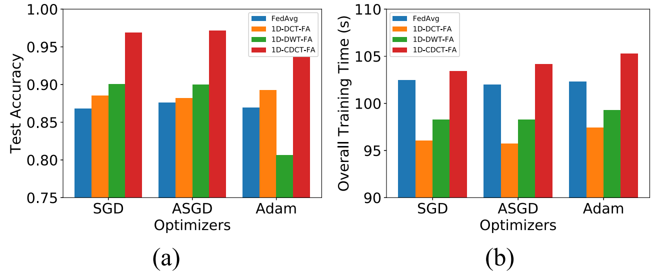
(a) (b)

Fig. 6. Performance of 1D-DWT and 2D-DWT-based FL of approximation information in different decomposition levels with IID local data sets. (a) Test accuracy. (b) Overall training time.



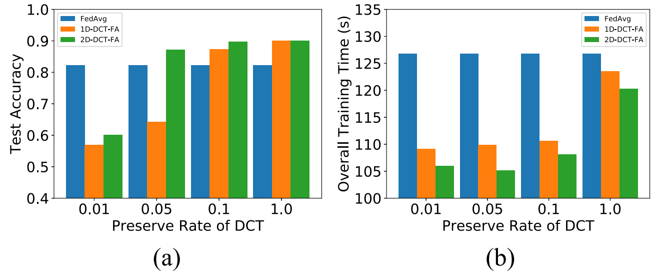
(a) (b)

Fig. 7. Performance comparison of different participation scenarios with IID local data sets. (a) Rate of train–test data size. (b) Aggregated local model number.



(a) (b)

Fig. 8. Performance of the FL of different optimizers with IID local data sets. (a) Test accuracy. (b) Overall training time.



(a) (b)

Fig. 9. Performance of 1D-DCT and 2D-DCT-based FL of different preserve rate with non-IID local data sets. (a) Test accuracy. (b) Overall training time.

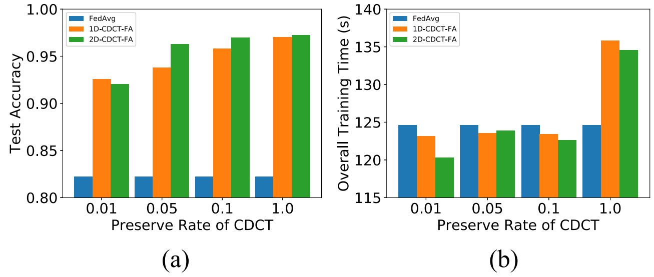
increasing decomposition levels, their performance both deteriorate. However, the performance of 1D-DWT-FA is more robust with the increasing decomposition levels compared with 2D-DWT-FA. Even in the third level decomposition where much less information is preserved, the performance of 1D-DWT-FA is still better than that of FedAvg with much less overall training time as shown in Fig. 6(b). It is because the approximation information of 1D-DWT-FA is a little more than that of 2D-DWT-FA at the same level. As shown in Fig. 6(b), 2D-DWT-FA needs a little more overall training time than that of 1D-DWT-FA. Both of them are more efficient than the FedAvg with all decomposition levels.

To compare the performance of different transform-domain FL schemes, we measured the test accuracy over 200 training runs with different training data sizes. In this experiment, the size of the test data sets was the same, and we controlled the size of the training data by the train–test rate. As illustrated in Fig. 7(a), when the train–test rate is 0.1, the performance of 1D-DCT-FA, 1D-DWT-FA, and 1D-CDCT-FA are all better than that of FedAvg. This limited training data size setting is commonly occurring in real IoT applications where each IoT device observes very limited data samples. This result shows the advantage of the transform-domain FL to deal with the limited local training data sets.

In real FL applications over the edge network, it is very hard to gather sufficient qualified edge servers for the specific application in each training round. Transform-domain FL still achieves promising performance with very limited edge servers participating in each training round. To evaluate the advantages of transform-domain FL when encountering the situation where both available edge servers and the local training data samples were limited, we set the number of local training samples for each digit class as 50, and there were 200 samples in each class for testing. In each federated training round, we randomly selected 2, 4, 6, or 8 edge servers to conduct the local model aggregation which perfectly simulated the scenario of many real IoT applications. The results in Fig. 7(b) show that with increasing qualified edge servers participating in the federated training, the performance of all schemes are improved, and the transform-domain FL schemes always show better performance with limited participating edge servers in each federated training round.

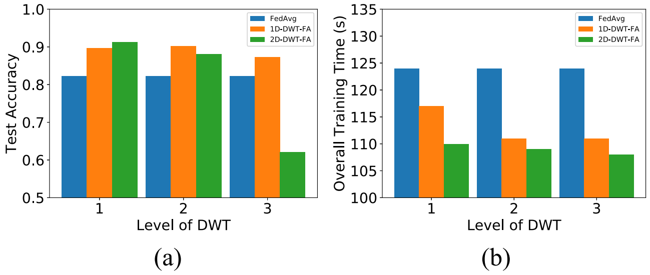
We also have checked the performance of transform-domain FL schemes with different optimizer in the local training. The results are shown in Fig. 8 where the performance with optimizer SGD is quite similar with the performance with optimizer ASGD in both test accuracy achieved after 200 training rounds and the consumed overall training time. However, the performance with optimizer Adam is not always good as the others. It is due to that both SGD and ASGD apply only the gradients from the current round to update the local model parameters, however, Adam applies the momentum of the gradients which needs to considering the historical gradients to adjust the current local parameter. This property becomes a drawback in the FL framework due to the local model aggregation procedure.

2) Simulation Results and Analysis With Non-IID Local Data Sets: Fig. 9 shows the test accuracy of FedAvg, 1D-DCT-FA, and 2D-DCT-FA with increasing preserve rate over the non-IID local data sets with heterogeneous level 0.1. Compared with the results over the IID local data sets shown in Fig. 4, the test accuracy of the FedAvg shown in Fig. 9(a) has been reduced due to the heterogeneous local data sets. However, the performance of both 1D-DCT-FA and



(a) (b)

Fig. 10. Performance of 1D-CDCT and 2D-CDCT-based FL of different preserve rates with non-IID local data sets. (a) Test accuracy. (b) Overall training time.



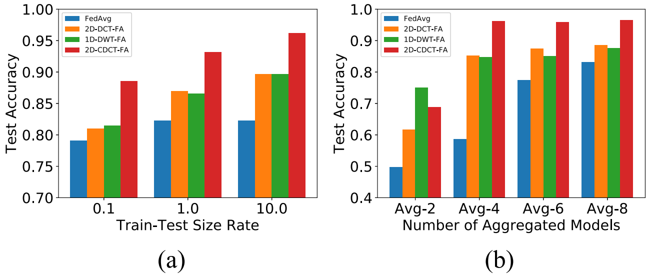
(a) (b)

Fig. 11. Performance of 1D-DWT and 2D-DWT-based FL of approximation information in different decomposition levels with non-IID local data sets. (a) Test accuracy. (b) Overall training time.

2D-DCT-FA are improved rapidly with the increasing preserve rate at the beginning and 2D-DCT-FA will out perform FedAvg when the preserve rate reaches 0.05 which means that 2D-DCT-FA needs only 5% features of that for FedAvg to achieve a better test accuracy. With the increasing preserve rate, the performance of 2D-DCT-FA converges much faster than that of 1D-DCT-FA where 1D-DCT-FA achieves better performance than FedAvg until preserve rate reaches 0.1 as illustrated in Fig. 9(a). Both 1D-DCT-FA and 2D-DCT-FA need much less overall training time to achieve a better performance over the non-IID local data sets where the overall training time varies very little with increasing preserver rate from 0.01 to 0.1. More interestingly, 2D-DCT-FA costs less overall training time compared with both FedAvg and 1D-DCT-FA as shown in Fig. 9(b). Both 1D-DCT-FA and 2D-DCT-FA achieve better performance with only 10% features compared with that of FedAvg at the same time with much less overall training time.

With non-IID local data sets, the schemes of combining the time-domain features and the frequency-domain ones, i.e., 1D-CDCT-FA and 2D-CDCT-FA, not only achieve much better test accuracy than that of FedAvg, but also less overall training time when the preserve rate is low as illustrated in Fig. 10. With low preserve rate 0.01, both 1D-CDCT-FA and 2D-CDCT-FA combine very limited frequency-domain features with the time domain features achieve more than 10% test accuracy and there is no compromise in the overall training time. When the preserve rate is 0.01, the performance of 1D-CDCT-FA is little better than that of 2D-CDCT-FA, but 2D-CDCT-FA has more advantage in the overall training time. The test accuracy of 2D-CDCT-FA is greatly improved when the preserve rate achieves 0.05 which is much better than that of 1D-CDCT-FA. Furthermore, when the preserve rate is no larger than 0.1, both 1D-CDCT-FA and 2D-CDCT-FA achieve promising overall training time compared with FedAvg.

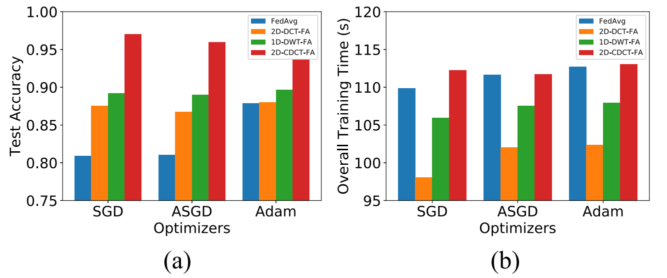
Both 1D-DWT-FA and 2D-DWT-FA achieve much efficient performance, i.e., higher test accuracy and lower overall training time, when the decomposition level is no larger than 2 with non-IID local data sets as illustrated in Fig. 11. With increasing decomposition levels, the test accuracy achieved by both 1D-DWT-FA and 2D-DWT-FA is reduced due to the decreasing information reserved as shown in Fig. 11(a). But there is advantage on the overall training time with higher decomposition level as illustrated in Fig. 11(b). The 2D-DWT-FA outperforms 1D-DWT-FA with first-level decomposition, however, the performance of 2D-DWT-FA declines rapidly with the increasing decomposition levels compared with 1D-DWT-FA. It is due to that the approximation information extracted from 2D-DWT-FA is much less than that of 1D-DWT-FA. With first-level decomposition, there is still sufficient approximation details reserved by 2D-DWT-FA which ensures the best performance compared with the other two schemes. However, when the decomposition level increases to 3, there is too less information of the sample features left in 2D-DWT-FA, the performance will be greatly deteriorated. As illustrated in Fig. 11(b), although there are advantages in overall training time with higher decomposition level, the benefit is not cost-effective with respect to the deterioration in test accuracy.



(a) (b)

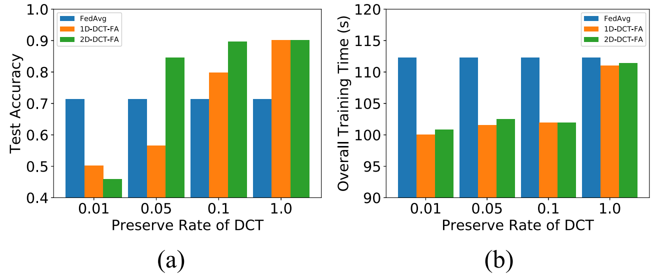
Fig. 12. Performance comparison of different participation scenarios with non-IID local data sets. (a) Rate of train–test data size. (b) Aggregated local model number.

In the experiment with different training data sizes over non-IID local data sets, we compared the performance of 2D-DCT-FA, 1D-DWT-FA, 2D-CDCT-FA with the FedAvg. The size of the test data sets were the same for each train–test size rate. As illustrated in Fig. 12(a), the performance of the transform-domain FL schemes all outperform FedAvg with limited training data size. Especially, 2D-CDCT-FA achieves very high test accuracy when train–test size rate is small compared with others. As illustrated in Fig. 12(a), the test accuracy for 2D-CDCT-FA achieves near 90% with very small local training data sets, i.e., train–test size rate is 0.1, where the FedAvg can not even get close to the test accuracy of 80% with the same local training data sets setting. Furthermore, for the medium local training data sets with train–test size rate as 1.0, 2D-DCT-FA achieves very high test accuracy near 95%, however, the test accuracy for FedAvg is still around 80%. This experiment result shows the benefit of transform-domain



(a) (b)

Fig. 13. Performance of the FL of different optimizers with non-IID local data sets. (a) Test accuracy. (b) Overall training time.



(a) (b)

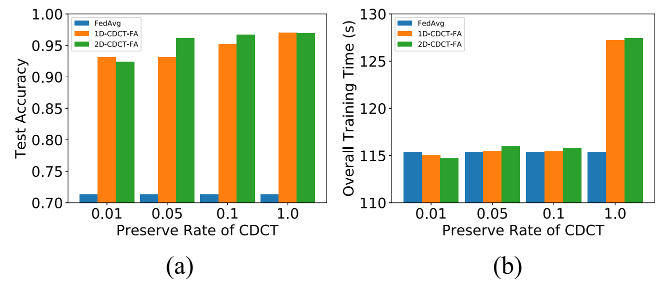
Fig. 14. Performance of 1D-DCT and 2D-DCT-based FL of different preserve rate with non-overlapping local data sets. (a) Test accuracy. (b) Overall training time.

FL also achieves very promising results with the limited and heterogeneous local training data sets.

Fig. 12(b) shows the advantages of transform-domain FL schemes over the FedAvg with very limited edge servers participating in each training round over the non-IID local data. The 2D-DCT-FA easily achieves the test accuracy around 95% when their are only 4 edge servers participating into the FL in each round where FedAvg only achieves the test accuracy less than 60%. Furthermore, both 2D-DCT-FA and 1D-DWT-FA achieve the test accuracy near 90% when the number of edge servers participating in each training round is more than 4. The performance of transform-domain FL schemes is very promising with limited participating edge servers in each federated training round over the non-IID local data sets.

As illustrated in Fig. 13, the transform-domain FL schemes all achieve better performance than the FedAvg with different optimizers over the non-IID local data sets. With heterogeneous local data sets, the performance of optimizer Adam on the test accuracy is a little better than both SGD and ASGD with respect to FedAVg, 2D-DCT-FA, and 1D-DWT-FA. Although, the performance of 2D-CDCT-FA with Adam is a little worse than the other two optimizers, it is still the best among the other schemes. Both 2D-DCT-FA and 1D-DWT-FA still achieve promising overall training time among all the schemes over different optimizers, and 2D-CDCT-FA also achieves comparable efficiency with FedAvg in overall training time. Furthermore, it is also observed that the impact of different optimizers over the non-IID local data sets on the transform-domain FL is very little compared with that of FedAvg in both test accuracy and overall training time.

3) Simulation Results and Analysis With Nonoverlap Local Data Sets: Fig. 14(a) shows the test accuracy of FedAvg, 1D-DCT-FA, and 2D-DCT-FA with increasing preserve rate over the nonoverlap local data sets. When the preserve rate is very small as 0.01, 1D-DCT-FA slightly outperforms 2D-DCT-FA, however, both of them can not achieve the same performance of FedAvg with test accuracy 70%. When the preserve rate increases to 0.05, which is still small, the performance of 2D-DCT-FA is greatly improved to achieve test accuracy near 85%, however, the performance of 1D-DCT-FA is still very poor with test accuracy around 55%. With increasing preserve rate, the performance of 1D-DCT-FA will be continuously improved, but it only achieves the same performance as 2D-DCT-FA with maximum preserve rate 1.0. Furthermore, 2D-DCT-FA achieves test accuracy of 90% with preserve rate 0.1 which is much efficient compared with that of FedAvg. As shown in Fig. 14(b), when the preserve rate is no more than 0.1, the overall training time of both 1D-DCT-FA and 2D-DCT-FA are much smaller than that of FedAvg. Furthermore, the impact of different preserve rates on the overall training time of both 1D-DCT-FA and 2D-DCT-FA is very limited. When set the preserve rate as 0.1, it achieves much better performance on the test accuracy but also remain small cost on the overall training time.

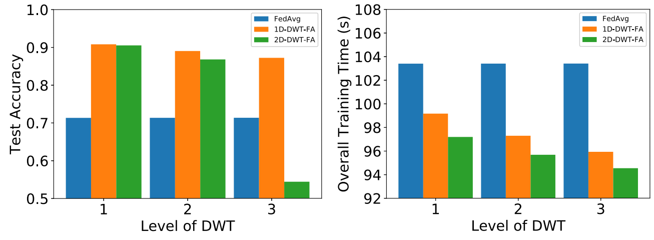


(a) (b)

Fig. 15. Performance of 1D-CDCT and 2D-CDCT-based FL of different preserve rates with non-overlapping local data sets. (a) Test accuracy. (b) Overall training time.

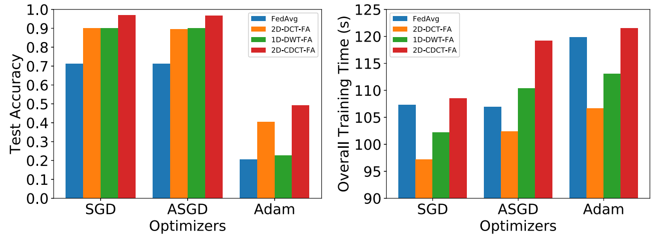
Both 1D-CDCT-FA and 2D-CDCT-FA achieve far better performance compared with FedAvg over the nonoverlap local data sets. As shown in Fig. 15(a), both 1D-CDCT-FA and 2D-CDCT-FA improve more than 20% of the test accuracy compared with FedAvg. When the preserve rate is 0.01, the performance of 1D-CDCT-FA maybe a little better than that of 2D-CDCT-FA, however, the improvement of 2D-CDCT-FA is faster than that of 1D-CDCT-FA with the increasing preserve rate. When the preserve rate reaches 0.05, 2D-CDCT-FA already achieves test accuracy over 95% where the 1D-CDCT-FA can only achieve the same test accuracy with the maximum preserve rate 1.0. Furthermore, the overall training time of both 1D-CDCT-FA and 2D-CDCT-FA also are comparative to that of FedAvg when preserve rate is no more than 0.01 as shown in Fig. 15(b). It shows that when combining a few frequency-domain features to the time-domain features, the testing performance is greatly improved over the nonoverlap local data sets.

When we extract the approximation information at different decomposition levels for training, the performance of both 1D-DWT-FA and 2D-DWT-FA can outperform FedAvg when the decomposition level is no greater than 2 over the nonoverlap local data sets as illustrated in Fig. 16(a). However, the performance of 2D-DWT-FA becomes much worse when



(a) (b)

Fig. 16. Performance of 1D-DWT and 2D-DWT-based FL of approximation information in different decomposition levels with non-overlapping local data sets. (a) Test accuracy. (b) Overall training time.



(a) (b)

Fig. 17. Performance of the FL of different optimizers with non-overlapping local data sets. (a) Test accuracy. (b) Overall training time.

the decomposition level reaches 3 due to less information at a higher decomposition level. Both 1D-DWT-FA and 2D-DWT-FA achieve their best performance with first-level decomposition. With the increasing decomposition level, the performance of 1D-DWT-FA will be slightly deteriorated compared with the large test accuracy reduction of 2D-DWT-FA. Although, the performance of 1D-DWT-FA is better than 2D-DWT-FA on test accuracy, 2D-DWT-FA have advantages on the overall training time and both of them outperform the FedAvg as shown in Fig. 16(b). Furthermore, the 1D-DWT-FA achieves the test accuracy near 90% in the third level decomposition with very little overall training time which is much efficient than that of FedAvg.

As shown in Fig. 17(a), the performance of applying optimizer Adam becomes much worse with nonoverlap local data sets for all FL schemes, and the transform-domain FL schemes achieve much better performance on test accuracy with optimizers SGD and ASGD, which are 20% higher than that of FedAvg. The performance deterioration with Adam over the nonoverlap local data sets is due to the property of Adam considering the historical local gradients to adjust the current local parameter which is highly unsuitable for the scenario where the local data sets are totally different with each other. The historical local gradient information can not be used to adjust the model parameters which try to learn the knowledge over all local data sets. Furthermore, due to the sophisticated updating mechanism, the optimizer Adam also does not have advantages on the overall training time as shown in Fig. 17(b), the overall training time of both FedAvg and 2D-CDCT-FA with Adam grow rapidly compared with optimizers SGD and ASGD, however, the overall training time growth of 2D-DCT-FA and 1D-DWT-FA are very slow.

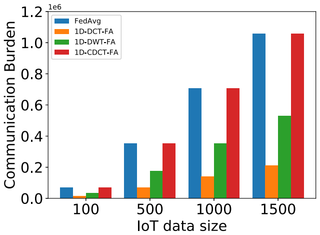


Fig. 18. Communication advantages.

As expected, transform-domain FL also provides considerable help in alleviating the overall communication burden among IoT devices and edge servers. As shown in Fig. 18, where the communication burden is defined as the amount of sample features needed to be transmitted from IoT devices to edge servers where each feature value needs one unit of communication resources. When we set the preserve rate of DCT-FA as 0.1, it only needs to transmit 10% features compared with that of the FedAvg applying the original sample features. From Fig. 18, we observe that DWT-FA also vastly relieves the communication burden compared with FedAvg due to the signal compression ability of the Haar wavelets. In this experiment, we apply the first-level DWT decomposition which holds the largest number of approximation features but still causing much less communication burden compared with that of FedAvg. In consideration of time-domain and transform-domain features combination in CDCT-FA, there is no extra communication burden for the IoT devices, since the feature transformation and combination is conducted in edge servers.

VII. Conclusion

In this article, based on the application scenarios of edge-enabled IoT intelligence, we proposed transform-domain FL algorithms to improve the federated training efficiency among multiple edge servers, which provides promising application service for various IoT devices with limited local data and resources. Since the transform-domain features provide sufficient information of the original data when reducing the dimensionality, the satisfactory convergence is maintained. Furthermore, the compressed frequency-domain features increases the similarity among different local objectives, which is important to address the heterogeneous challenges facing FL. Thus, the proposed transform-domain technique leads to faster convergence and reduced communication cost. The performance of the proposed method was also analyzed from a series of experiments and demonstrated its advantages over the SOTA FL algorithms with popular data sets.

Appendix

Lemma 1: Let $f(\boldsymbol{x})$ be a β-smooth convex function, then

$$(\nabla \boldsymbol{f}(\boldsymbol{x}) - \nabla \boldsymbol{f}(\boldsymbol{y}))^{\boldsymbol{T}}(\boldsymbol{x} - \boldsymbol{y}) \geq \frac{1}{2\beta}\|\nabla \boldsymbol{f}(\boldsymbol{x}) - \nabla \boldsymbol{f}(\boldsymbol{y})\|^{\mathbf{2}}. \quad (21)$$

Proof: Due to the quadratic upper bound and the linear lower bound of the β-smooth convex function, we obtain the inequality as

$$\begin{aligned} f(\boldsymbol{x}) - f(\boldsymbol{y}) &= f(\boldsymbol{x}) - f(z) + f(z) - f(\boldsymbol{y}) \\ &\leq \nabla \boldsymbol{f}(\boldsymbol{x})^{\boldsymbol{T}}(\boldsymbol{x} - z) + \nabla \boldsymbol{f}(\boldsymbol{y})^{\boldsymbol{T}}(z - \boldsymbol{y}) + \frac{\beta}{2}||z - \boldsymbol{y}||^2 \\ &= \nabla \boldsymbol{f}(\boldsymbol{x})^{\boldsymbol{T}}(\boldsymbol{x} - \boldsymbol{y}) + (\nabla \boldsymbol{f}(\boldsymbol{x}) - \nabla \boldsymbol{f}(\boldsymbol{y}))^{\boldsymbol{T}}(\boldsymbol{y} - z) \\ &\quad + \frac{\beta}{2}||z - \boldsymbol{y}||^2. \end{aligned}$$

If we define

$$z = \boldsymbol{y} - \frac{1}{\beta}(\nabla \boldsymbol{f}(\boldsymbol{y}) - \nabla \boldsymbol{f}(\boldsymbol{x}))$$

then

$$\begin{aligned} &(\nabla \boldsymbol{f}(\boldsymbol{x}) - \nabla \boldsymbol{f}(\boldsymbol{y}))^{\boldsymbol{T}}(\boldsymbol{y} - z) = -\frac{1}{\beta}\|\nabla \boldsymbol{f}(\boldsymbol{x}) - \nabla \boldsymbol{f}(\boldsymbol{y})\|^2 \\ &\frac{\beta}{2}||z - \boldsymbol{y}||^2 = \frac{1}{2\beta}\|\nabla \boldsymbol{f}(\boldsymbol{x}) - \nabla \boldsymbol{f}(\boldsymbol{y})\|^2 \end{aligned}$$

and, hence

$$f(\boldsymbol{x}) - f(\boldsymbol{y}) \leq \nabla \boldsymbol{f}(\boldsymbol{x})^{\boldsymbol{T}}(\boldsymbol{x} - \boldsymbol{y}) - \frac{1}{2\beta}||\nabla \boldsymbol{f}(\boldsymbol{x}) - \nabla \boldsymbol{f}(\boldsymbol{y})||^2$$

which leads to

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) - \nabla \boldsymbol{f}(\boldsymbol{x})^{\boldsymbol{T}}(\boldsymbol{y} - \boldsymbol{x}) \geq \frac{1}{2\beta}||\nabla \boldsymbol{f}(\boldsymbol{x}) - \nabla \boldsymbol{f}(\boldsymbol{y})||^2.$$

Using the linear bound of $f(\boldsymbol{x})$, we have

$$\begin{aligned} \nabla \boldsymbol{f}(\boldsymbol{y})^{\boldsymbol{T}}(\boldsymbol{x} - \boldsymbol{y}) &\leq f(\boldsymbol{x}) - f(\boldsymbol{y}) \\ &\leq \nabla \boldsymbol{f}(\boldsymbol{x})^{\boldsymbol{T}}(\boldsymbol{x} - \boldsymbol{y}) - \frac{1}{2\beta}||\nabla \boldsymbol{f}(\boldsymbol{x}) - \nabla \boldsymbol{f}(\boldsymbol{y})||^2 \end{aligned}$$

which leads to (21). ■

Lemma 2: For any positive number a, we get the relaxed Triangle inequalities as

$$||\boldsymbol{x} + \boldsymbol{y}||^2 \leq (1 + a)||\boldsymbol{x}||^2 + \left(1 + \frac{1}{a}\right)||\boldsymbol{y}||^2 \tag{22}$$

$$2\left(||\boldsymbol{x}||^2 + ||\boldsymbol{y}||^2\right) \geq ||\boldsymbol{x} + \boldsymbol{y}||^2. \tag{23}$$

Proof: To get the relaxed Triangle inequality in (22), we follow the derivation as

$$\begin{aligned} ||\boldsymbol{x} + \boldsymbol{y}||^2 &= (1 + a)||\boldsymbol{x}||^2 + \left(1 + \frac{1}{a}\right)||\boldsymbol{y}||^2 - \left\|\sqrt{a}\boldsymbol{x} + \frac{1}{\sqrt{a}}\boldsymbol{y}\right\|^2 \\ &\leq (1 + a)||\boldsymbol{x}||^2 + \left(1 + \frac{1}{a}\right)||\boldsymbol{y}||^2. \end{aligned}$$

When we set $a = 1$, we can obtain the result in (23). ■

Lemma 3: Let σ^2 be an upper bound of the variance of a sequence of random vectors $\{\boldsymbol{v_t}\}_{\boldsymbol{t=1}}^{\boldsymbol{T}}$, then

$$\boldsymbol{E}\left[\left\|\sum_{\boldsymbol{t=1}}^{\boldsymbol{T}}\boldsymbol{v_t}\right\|^2\right] \leq \left\|\sum_{\boldsymbol{t=1}}^{\boldsymbol{T}}\boldsymbol{E}[\boldsymbol{v_t}]\right\|^2 + T\sigma^2. \tag{24}$$

Proof: The variance of sequence $\{\boldsymbol{v_t}\}_{\boldsymbol{t=1}}^{\boldsymbol{T}}$ is defined by

$$\begin{aligned} \boldsymbol{E}\Big[\|\boldsymbol{v_t} - \boldsymbol{E}[\boldsymbol{v_t}]\|^2\Big] &= \boldsymbol{E}\Big[||\boldsymbol{v_t}||^2\Big] - 2||\boldsymbol{E}[\boldsymbol{v_t}]||^2 \\ &\quad + ||\boldsymbol{E}[\boldsymbol{v_t}]||^2 = \boldsymbol{E}\Big[||\boldsymbol{v_t}||^2\Big] - ||\boldsymbol{E}[\boldsymbol{v_t}]||^2. \end{aligned}$$

Similarly,

$$\boldsymbol{E}\left[\left\|\sum_{\boldsymbol{t=1}}^{\boldsymbol{T}}(\boldsymbol{v_t} - \boldsymbol{E}[\boldsymbol{v_t}])\right\|^2\right] = \boldsymbol{E}\left[\left\|\sum_{\boldsymbol{t=1}}^{\boldsymbol{T}}\boldsymbol{v_t}\right\|^2\right] - \left\|\sum_{\boldsymbol{t=1}}^{\boldsymbol{T}}\boldsymbol{E}[\boldsymbol{v_t}]\right\|^2.$$

Using Jensen's inequality, we have

$$\left\|\sum_{\boldsymbol{t=1}}^{\boldsymbol{T}}(\boldsymbol{v_t} - \boldsymbol{E}[\boldsymbol{v_t}])\right\|^2 \leq \sum_{\boldsymbol{t=1}}^{\boldsymbol{T}}\|\boldsymbol{v_t} - \boldsymbol{E}[\boldsymbol{v_t}]\|^2$$

and the linearity of the expectation gives

$$E\left[\left\|\sum_{\boldsymbol{t=1}}^{\boldsymbol{T}}(\boldsymbol{v_t} - \boldsymbol{E}[\boldsymbol{v_t}])\right\|^2\right] \leq \sum_{\boldsymbol{t=1}}^{\boldsymbol{T}}\boldsymbol{E}\Big[\|\boldsymbol{v_t} - \boldsymbol{E}[\boldsymbol{v_t}]\|^2\Big] \leq T\sigma^2$$

which immediately leads to (24). ■

Lemma 4: If $f(\boldsymbol{x})$ is a β-smooth and μ-strongly convex function, thus, admitting

$$\begin{aligned} f(z) &\leq f(\boldsymbol{x}) + \nabla \boldsymbol{f}(\boldsymbol{x})^{\boldsymbol{T}}(z - \boldsymbol{x}) + \frac{\beta}{2}||z - \boldsymbol{x}||^2 \\ f(\boldsymbol{y}) &\geq f(\boldsymbol{x}) + \nabla \boldsymbol{f}(\boldsymbol{x})^{\boldsymbol{T}}(\boldsymbol{y} - \boldsymbol{x}) + \frac{\mu}{2}||\boldsymbol{y} - \boldsymbol{x}||^2 \end{aligned}$$

then

$$\nabla \boldsymbol{f}(\boldsymbol{x})^{\boldsymbol{T}}(z - \boldsymbol{y}) \geq f(z) - f(\boldsymbol{y}) + \frac{\mu}{4}||\boldsymbol{y} - z||^2 - \beta||z - \boldsymbol{x}||^2. \tag{25}$$

Proof: Using the Cauchy–Schwarz inequality, we have

$$\nabla \boldsymbol{f}(\boldsymbol{x})^{\boldsymbol{T}}(z - \boldsymbol{y}) \geq f(z) - f(\boldsymbol{y}) + \frac{\mu}{2}||\boldsymbol{y} - \boldsymbol{x}||^2 - \frac{\beta}{2}||z - \boldsymbol{x}||^2$$

which in conjunction with the upper and lower bounds with $\beta \geq \mu$, we obtain (25). ■

REFERENCES

[1] C.-H. Chen, M.-Y. Lin, and C.-C. Liu, "Edge computing gateway of the Industrial Internet of Things using multiple collaborative microcontrollers," *IEEE Netw.*, vol. 32, no. 1, pp. 24–32, Jan./Feb. 2018.

[2] J. Ni, X. Lin, and X. S. Shen, "Toward edge-assisted Internet of Things: From security and efficiency perspectives," *IEEE Netw.*, vol. 33, no. 2, pp. 50–57, Mar./Apr. 2019.

[3] L. Zhao, J. Wang, J. Liu, and N. Kato, "Optimal edge resource allocation in IoT-based smart cities," *IEEE Netw.*, vol. 33, no. 2, pp. 30–35, Mar./Apr. 2019.

[4] Y. Meng, W. Zhang, H. Zhu, and X. S. Shen, "Securing consumer IoT in the smart home: Architecture, challenges, and countermeasures," *IEEE Wireless Commun.*, vol. 25, no. 6, pp. 53–59, Dec. 2018.

[5] R. Kelly, *Internet of Things Data to Top 1.6 Zettabytes by 2020*, Campus Technol., Los Angeles, CA, USA, 2015.

[6] L. Zhao, X. Lan, L. Cai, and J. Pan, "Adaptive content placement in edge networks based on hybrid user preference learning," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2019, pp. 1–6.

[7] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, vol. 2, 2020, pp. 429–450.

[8] S. Wang et al., "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[9] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4177–4186, Jun. 2020.

[10] 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.

[11] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Differentially private asynchronous federated learning for mobile edge computing in urban informatics," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 2134–2143, Mar. 2020.

[12] D. Basu, D. Data, C. Karakus, and S. Diggavi, "Qsparse-local-SGD: Distributed SGD with quantization, sparsification and local computations," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

[13] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Stat.*, 2017, pp. 1273–1282.

[14] M. Zinkevich, M. Weimer, L. Li, and A. Smola, "Parallelized stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23, 2010, pp. 1–9.

[15] S. U. Stich, "Local SGD converges fast and communicates little," 2018, *arXiv:1805.09767*.

[16] X. Lian, Y. Huang, Y. Li, and J. Liu, "Asynchronous parallel stochastic gradient for nonconvex optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 2737–2745.

[17] P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, nos. 1–2, pp. 1–210, 2021.

[18] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-iid data," 2019, *arXiv:1907.02189*.

[19] A. Khaled, K. Mishchenko, and P. Richtárik, "Tighter theory for local SGD on identical and heterogeneous data," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2020, pp. 4519–4529.

[20] H. Yang, M. Fang, and J. Liu, "Achieving linear speedup with partial worker participation in non-iid federated learning," 2021, *arXiv:2101.11203*.

[21] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5132–5143.

[22] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," 2018, *arXiv:1806.00582*.

[23] H. Yu, S. Yang, and S. Zhu, "Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 5693–5700.

[24] X. Liang, S. Shen, J. Liu, Z. Pan, E. Chen, and Y. Cheng, "Variance reduced local SGD with lower communication complexity," 2019, *arXiv:1912.12844*.

[25] S. P. Karimireddy et al., "Breaking the centralized barrier for cross-device federated learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 28663–28676.

[26] E. Gorbunov, K. P. Burlachenko, Z. Li, and P. Richtárik, "MARINA: Faster non-convex distributed learning with compression," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 3788–3798.

[27] X. Gu, K. Huang, J. Zhang, and L. Huang, "Fast federated learning in the presence of arbitrary device unavailability," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 1–13.

[28] J. Wang and G. Joshi, "Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms," 2018, *arXiv:1808.07576*.

[29] X. Li, W. Yang, S. Wang, and Z. Zhang, "Communication efficient decentralized training with multiple local updates," 2019, *arXiv:1910.09126*.

[30] J. Wang, A. K. Sahu, Z. Yang, G. Joshi, and S. Kar, "MATCHA: Speeding up decentralized SGD via matching decomposition sampling," 2019, *arXiv:1905.09435*.

[31] H. B. McMahan, E. Moore, D. Ramage, S. Hampson et al., "Communication-efficient learning of deep networks from decentralized data," 2016, *arXiv:1602.05629*.

[32] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Boston, MA, USA: Academic, 2014.

[33] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[34] Y. LeCun, C. Cortes, and C. J. Burges. "The MNIST database of handwritten digits." 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist

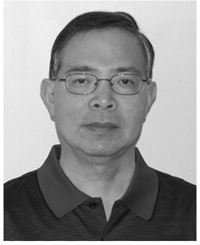
Lei Zhao (Student Member, IEEE) received the B.S. and M.A.Sc. degrees in computer science and technology from Xidian University, Xi'an, China, in 2015 and 2018, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada.

His current research interests include federated learning and optimization.

Lin Cai (Fellow, IEEE) received the M.A.Sc. and Ph.D. degrees (awarded Outstanding Achievement in Graduate Studies) in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2002 and 2005, respectively.

She has been with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada, since 2005, where she is currently a Professor. Her research interests span several areas in communications and networking, with a focus on network protocol and architecture design supporting emerging multimedia traffic and Internet of Things.

Dr. Cai was elected as a member of the Royal Society of Canada's College of New Scholars, Artists and Scientists in 2020, and the 2020 "Star in Computer Networking and Communications" by N2Women. She has co-founded and chaired the IEEE Victoria Section Vehicular Technology and Communications Joint Societies Chapter. She has been elected to serve the IEEE Vehicular Technology Society Board of Governors, from 2019 to 2024. She has served as an Associate Editor-in-Chief for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, a member of the Steering Committee of the IEEE TRANSACTIONS ON BIG DATA and IEEE TRANSACTIONS ON CLOUD COMPUTING, an Associate Editor for the IEEE INTERNET OF THINGS JOURNAL, IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON COMMUNICATIONS, *EURASIP Journal on Wireless Communications and Networking*, *International Journal of Sensor Networks*, and *Journal of Communications and Networks*, and as the Distinguished Lecturer of the IEEE VTS and ComSoc Societies. She is a Fellow of the NSERC E.W.R. Steacie Memorial and Engineering Institute of Canada.



Wu-Sheng Lu (Life Fellow, IEEE) received the B.Sc. degree in mathematics from Fudan University, Shanghai, China, in 1964, and the M.S. degree in electrical engineering and the Ph.D. degree in control science from the University of Minnesota, Minneapolis, MN, USA, in 1983 and 1984, respectively.

He has been with the University of Victoria, Victoria, BC, Canada, since 1987, where he is currently a Professor Emeritus. He has coauthored with A. Antoniou of the *Two-Dimensional Digital Filters* (Marcel Dekker, 1992) and *Practical Optimization: Algorithms and Engineering Applications* (2nd ed., Springer, 2021), and with E. K. P. Chong and S. H. Zak of *An Introduction to Optimization* (5th ed., Wiley, 2023).

Dr. Lu served as an Editor for the *Canadian Journal of Electrical and Computer Engineering* and an Associate Editor for several journals, including the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART I: REGULAR PAPERS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II: EXPRESS BRIEFS, *International Journal of Multidimensional Systems and Signal Processing*, and *Journal of Circuits, Systems, and Signal Processing*.