

Energy-Efficient Personalized Federated Learning for Establishing Green IoT

Haowen Xu[†], Yingchi Mao[†], Haotian Zheng[†], Xiaoming He[‡], Yi Rong[†], Mingkai Chen[§], and Saba Al-Rubaye^{††}

[†]College of Computer Science and Software Engineering, Hohai University, Nanjing, China

[‡]College of Internet of Things, Nanjing University of Posts and Telecommunication, Nanjing, China.

[§]School of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China

^{††}Faculty of Engineering & Applied Science, Cranfield University, Bedford, UK

Abstract—Green Internet of Things (Green IoT) is a technique that intends to reduce energy consumption and carbon emissions of Internet of Things (IoT) devices by optimizing hardware design, communication protocols, and data processing. One of the most promising schemes to realize Green IoT is personalized Federated Learning (pFL). Unfortunately, existing pFL methods still need further improvement in achieving Green IoT from the following aspects. 1) Computational energy consumption: model training on IoT devices generates a substantial amount of computational energy consumption. 2) Model performance: the dynamic role differences in each layer of the trained deep neural network need to be considered. Jointly considering these aspects, we present a novel pFL framework named Energy-Efficient personalized Federated Learning (EE-pFL) for establishing Green IoT. Specifically, an IoT device serves as an edge server. Each IoT device produces a customized model through a *model training* phase and a *model aggregation* phase. In the model training phase, a threshold-based sparsification strategy is introduced to reduce the computational energy consumption of IoT devices by selectively executing parameter updates. In the model aggregation phase, layer aggregation and an Adaptive Weight Calculation (AWC) mechanism are proposed to capture dynamic role differences in different layers of a deep neural network. Experimental results demonstrate that EE-pFL shows lower computational energy consumption and higher classification accuracy than advanced benchmarks.

Index Terms—Green IoT, energy-efficient, pFL

I. INTRODUCTION

The Internet of Things (IoT) connects various physical devices through the Internet to enable communication and information processing [1]–[3]. Currently, with the continuous emergence of smart devices, the number of connected IoT devices is experiencing exponential growth, thus generating massive data. This faces challenges in energy consumption. Due to the increasing attention in IoT, its sustainability cannot be neglected. Hence, some scholars begin to focus on the Green Internet of Things (Green IoT), which aims to reduce energy consumption of IoT devices through optimizing hardware design, communication protocols, and data processing [4].

To achieve Green IoT, personalized Federated Learning (pFL) is regarded as one of the most promising schemes. pFL is an improved version of Federated Learning (FL). The key point of FL is to combine all local models deployed on IoT devices to implement a more generalized global model. Different from FL, pFL learns customized models corresponding to individual IoT devices. The customized models adapt to data distribution

of corresponding IoT devices. Traditional machine learning methods typically employ centralized learning to process data and train models, *i.e.*, data from IoT devices is transmitted to a distant cloud server for training. The centralized learning consumes a lot of energy due to transmitting massive data from IoT devices [5]–[7]. Fortunately, pFL flips the centralized learning through training at the device level, instead of training at the cloud server. There is an extreme reduction in data volume that requires to be transmitted from IoT devices to the cloud server. Despite the huge potential in establishing Green IoT, existing pFL methods still necessitate further enhancement from the following two aspects.

1) Computational Energy Consumption: Since existing pFL methods need to train customized models on IoT devices, this brings about computational load to IoT devices, thus consuming energy. We refer to the energy consumption caused by model training on IoT devices as computational energy consumption. Several studies show that computational energy consumption in pFL is comparable to energy consumption for the wireless transmissions between IoT devices and the cloud server, especially when the trained model is a deep learning model (*e.g.*, Alexnet) [8]. Hence, a reliable design of the energy-efficient pFL oughts to simultaneously take computational energy consumption and communication-related energy consumption. Nonetheless, most works in local model training mainly focus on computation resource allocation under the pFL convergence limitations, while ignoring the computational energy consumption in model parameter update.

2) Model Performance: In a deep neural network, different layers play various roles. Given a Convolutional Neural Network (CNN) model involving two Convolutional (Conv) layers and two Fully Connected (FC) layers, the initial two Conv layers primarily focus on feature extraction, while the latter FC layers are responsible for integrating features and making decisions. Additionally, the role of each layer in the deep neural network is dynamic and continuous changes in iterations of training round. Most pFL methods treat the model as a whole when aggregating, while neglecting the different roles of each layer in the deep neural network [9]–[12]. Few studies have investigated layer aggregation, but there is a lack of adaptive aggregation weights to capture the dynamic changes of layers' roles, leading to inaccurate personalization [13]–[15].

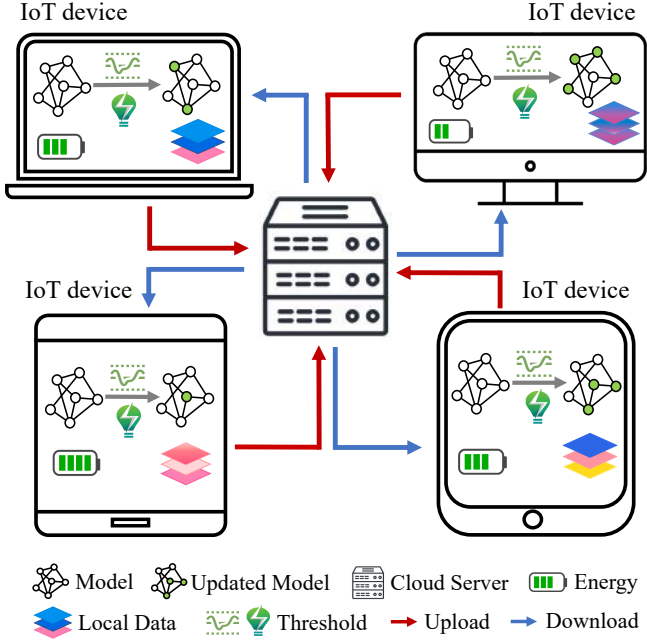


Fig. 1. EE-pFL for establishing Green IoT. EE-pFL is composed of a cloud server and several IoT devices. IoT devices are powered by batteries. In each communication round, each IoT device first downloads the global model from the cloud server. Then, each IoT device trains its customized model using local data and global model (local model). Meanwhile, the local model is updated on the IoT device through local data. During the training, the threshold-based sparsification strategy is adopted to reduce computational energy consumption. Finally, the local model is uploaded to the cloud server for updating the global model.

To address the above challenges, We present a novel pFL framework named Energy-Efficient personalized Federated Learning (EE-pFL), to effectively establish Green IoT. Specifically, the IoT device serves as an edge server. Each IoT device generates a customized model through two phases, a *model training* phase and a *model aggregation* phase. In the model training phase, we design an energy-efficient parameter update scheme, which introduces a threshold-based sparsification strategy to reduce the computational energy consumption of IoT devices through selectively updating parameters in each iteration. In the model aggregation phase, layer aggregation is first used to implement layer-level aggregation with aggregation weights. The Adaptive Weight Calculation (AWC) mechanism is then used to adaptively measure and update aggregation weights based on the dynamic role differences of model layers. The main idea of EE-pFL for establishing Green IoT is presented in Fig. 1. The primary contributions of this work are summarized as follows:

- We present a novel pFL framework named EE-pFL for establishing Green IoT. To reduce computational energy consumption during model training, we develop an energy-efficient parameter update scheme using a threshold-based sparsification strategy. Specifically, a sparse threshold value is first set. Then, we filter out the differential parameters that their absolute values are less than the sparse threshold value, while the remaining differential

parameters are updated.

- To capture dynamic role differences in model layers, in each IoT device, we design a novel model aggregation strategy with layer aggregation and an AWC mechanism. The former enables a layer-wised aggregation of the local model and the customized model for each IoT device based on aggregation weights. The latter aims to adaptively update the aggregation weights in accordance with the dynamic role differences in model layer.
- We conduct simulation experiments on three public and representative datasets (e.g., CIFAR-100, Tiny-ImageNet, and AG News). Experimental results prove that our proposed model is superior to advanced benchmarks in computational energy consumption and classification accuracy.

II. EE-pFL DESIGN

A. Problem Formulation

In the work, our objective is to train customized models on IoT devices with heterogeneous IoT device data for establishing Green IoT. It is worth noting that an edge server is an IoT device. We select a multi-layer deep neural network as the model to be trained. We identify a classification task as the task that the model strives to accomplish.

Specifically, given N IoT devices, their classification task datasets are $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_N$, respectively. These datasets are non-Independent and Identically Distributed (non-IID) with statistical heterogeneity. Let $\mathcal{D}_i = \{(x_{i,b}, y_{i,b})\}_{b=1}^{m_i}$, $i \in [1, N]$ denote the dataset of the i^{th} IoT device, where $x_{i,b}$ and $y_{i,b}$ stand for the b^{th} sample and the corresponding label in the i^{th} IoT device's dataset. m_i is the total number of samples in the i^{th} IoT device's dataset. The total size of all IoT device datasets is $M = \sum_{i=1}^N m_i$. We collaboratively train customized models $\bar{v}_1, \bar{v}_2, \dots, \bar{v}_N$ through $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_N$, where $\bar{v}_i$ is a customized model for the i^{th} IoT device. The optimal customized models are acquired through minimizing the global loss:

$$(\bar{v}_1, \bar{v}_2, \dots, \bar{v}_N) = \arg \min_{\bar{v}} \sum_{i=1}^N \frac{m_i}{M} \mathcal{L}_i, \quad (1)$$

$$\mathcal{L}_i = \mathcal{L}(\mathcal{D}_i; \bar{v}_i; w), \forall i \in [N], \quad (2)$$

where $\mathcal{L}(\cdot)$ stands for the global loss function. $\mathcal{L}_i$ represents as the loss function of the i^{th} IoT device related to dataset $\mathcal{D}_i$, measuring the discrepancies between the predicted value and the real label of m_i data samples. w stands for the global model, which provides useful information to train the i^{th} IoT device's customized model.

B. System Overview

In order to address the above problem, we present a pFL framework named EE-pFL. In each communication round, EE-pFL can be decomposed into two phases: a model training phase and a model aggregation phase. Fig. 2 shows the workflow of EE-pFL in the t^{th} round of communications. A total number of N sensing devices are involved in the communication round. Taking the i^{th} IoT device as an example, the IoT device stores three models, containing the local model w_i^t , the intermediate

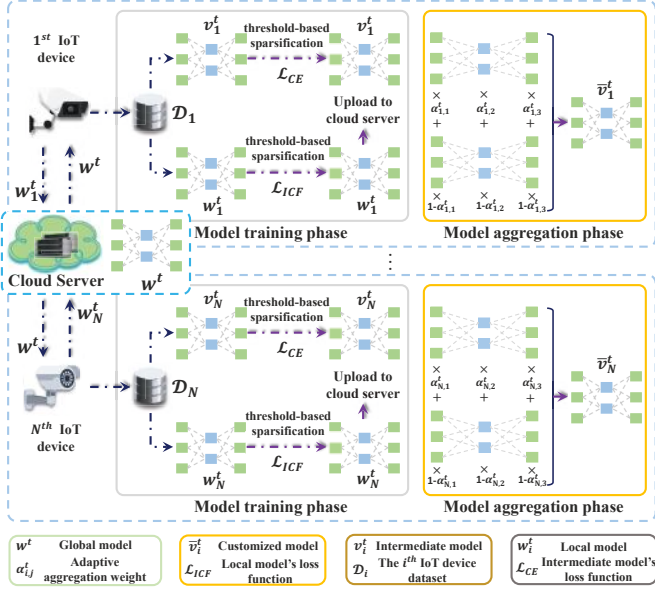


Fig. 2. The workflow of EE-pFL.

version of the customized model called intermediate model v_i^t , and the customized model $\bar{v}_i^t$. The initial intermediate model v_i^t is derived from the customized model $\bar{v}_i^{t-1}$ output in the $t-1$ th round for the i th sensing device. The initial local model w_i^t is distributed by the global model w^{t-1} generated through local model aggregations before the t th round of communication on the cloud server. Note that all N local models are aggregated using synchronization. Each IoT device stores these three models.

In the model training phase, dual loss functions are adopted to learn the local model w_i^t and the intermediate model v_i^t . For the training of w_i^t , the cross-entropy loss function is adopted. Moreover, to alleviate the low performance of the local model on the minority class in IoT device data, we introduce ICF into the cross-entropy loss function for updating the local model. We refer to the new loss function as $\mathcal{L}_{ICF}$. For the parameter update of w_i^t in the t th round, we design a threshold-based sparsification strategy to reduce computational energy consumption. The threshold-based sparsification strategy can selectively update parameters in the round, thus reducing the computation load on the IoT device. After training, we upload the local model to a cloud server for synchronous aggregation, which produces a new global model for the next communication round. Additionally, for the training of v_i^t , we directly use the cross-entropy loss function, which can be denoted as $\mathcal{L}_{CE}$. Similarly, we use the threshold-based sparsification strategy to update the v_i^t 's parameters.

In the model aggregation phase, to capture the dynamic role differences of model layers, the layer aggregation and an AWC mechanism are introduced to implement the aggregation of the local model w_i^t and the intermediate model v_i^t on the i th IoT device.

All IoT devices execute the model training phase and the model aggregation phase. Next, we describe these two phases

in greater detail.

C. Model Training Phase

As shown in Fig. 2, we design the loss function $\mathcal{L}_{ICF}$ and the loss function $\mathcal{L}_{CE}$ to train w_i^t and v_i^t , respectively. Below, we detail how each loss function is designed to train corresponding models for the i th IoT device in the t th communication round.

1) *The loss function $\mathcal{L}_{ICF}$* : The global model is trained through data from all IoT devices. These data contain most samples from the majority classes, while few samples come from the minority classes. In light of this, the training of global model primarily targets the majority class samples and exhibits poor performance on the minority class samples. If we directly incorporate the global model into model aggregation on each IoT device, it leads to the unsatisfactory performance of the customized model on the minority class samples.

To address the issue, ICF is considered in constructing the loss function for training w_i^t . The main reason is that ICF can weight loss and improve the proportion of the total loss contributed by the minority class.

The frequency of each class sample is calculated as follows:

$$F_j = m_{i,j}/m_i, j \in [1, C_i], \quad (3)$$

where $m_{i,j}$ denotes as the sample size in the j th class for the i th IoT device. C_i is the total number of classes in the i th IoT device dataset, and F_j denotes the frequency of the j th class. Based on these frequencies, the weight for the loss of each class sample is derived as follows:

$$\mu_j = 1/F_j, j \in [1, C_i], \quad (4)$$

where μ_j represents as the weight for the loss of the j th class sample. The loss function for training w_i^t is calculated as:

$$\mathcal{L}_{ICF} = \sum_{b=1}^{m_i} u_j \xi(x_{i,b}, y_{i,b}, w_i^t), j \in [1, C_i], \quad (5)$$

where ξ is the cross-entropy loss function, and the sample $(x_{i,b}, y_{i,b})$ belongs to the j th class label, i.e., $y_{i,b} = j$. Finally, the gradient descent is employed to update w_i^t , denoted as:

$$w_i^t = w^{t-1} - \eta \nabla_{w^{t-1}} \mathcal{L}_{ICF}(w^{t-1}), \quad (6)$$

where $\nabla_{w^{t-1}} \mathcal{L}_{ICF}(\cdot)$ denotes the stochastic gradient of w^{t-1} in the loss function $\mathcal{L}_{ICF}$ on the i th IoT device, and η is the learning rate.

To reduce the computational energy consumption on the i th IoT device, the threshold-based sparsification strategy is adopted to reduce the size of the parameter update. Specifically, we first determine the sparse threshold value. Then, we filter out the difference parameters with absolute values $\{|\mathcal{S}(\eta \nabla_{w^{t-1}} \mathcal{L}_{ICF}(w^{t-1}))|, i \in [1, N]\}$ less than the sparse threshold value, while the remaining parameters $\{\Delta w_i^{t-1} = \eta \nabla_{w^{t-1}} \mathcal{L}_{ICF}(w^{t-1}) - \mathcal{S}(\eta \nabla_{w^{t-1}} \mathcal{L}_{ICF}(w^{t-1})), i \in [1, N]\}$ are updated. After sparse operation, we can obtain w_i^t :

$$w_i^t = w_i^{t-1} - \Delta w_i^{t-1}, i \in [1, N]. \quad (7)$$

2) *The loss function $\mathcal{L}_{CE}$* : Since the intermediate model v_i^t does not engage in local model aggregation and has no impact

on other IoT devices, we directly adopt the cross-entropy loss function ξ . Hence, the loss function $\mathcal{L}_{CE}$ of the intermediate model v_i^t is expressed as follows:

$$\mathcal{L}_{CE} = \sum_{b=1}^{m_i} \xi(x_{i,b}, y_{i,b}, v_i^t). \quad (8)$$

We then update the intermediate model v_i^t through the following gradient descent:

$$v_i^t = \bar{v}_i^{t-1} - \eta \nabla_{\bar{v}_i^{t-1}} \mathcal{L}_{CE}(\bar{v}_i^{t-1}), \quad (9)$$

where $\nabla_{\bar{v}_i^{t-1}} \mathcal{L}_{CE}$ stands for the stochastic gradient of $\bar{v}_i^{t-1}$ in the loss function $\mathcal{L}_{CE}$, and $\bar{v}_i^{t-1}$ denotes the customized model from the output of the $t-1^{th}$ round communication.

To further reduce the computational energy consumption on the i^{th} IoT device, we employ the threshold-based sparsification strategy to decrease the size of parameter updates for the intermediate model v_i^t . In detail, the sparse threshold value is first determined. Furthermore, the difference parameters with absolute values $\left\{ \left| \mathcal{S} \left(\eta \nabla_{\bar{v}_i^{t-1}} \mathcal{L}_{CE}(\bar{v}_i^{t-1}) \right) \right|, i \in [1, N] \right\}$ less than the sparse threshold value, while the remaining parameters $\left\{ \Delta \bar{v}_i^{t-1} = \eta \nabla_{\bar{v}_i^{t-1}} \mathcal{L}_{CE}(\bar{v}_i^{t-1}) - \mathcal{S} \left(\eta \nabla_{\bar{v}_i^{t-1}} \mathcal{L}_{CE}(\bar{v}_i^{t-1}) \right), i \in [1, N] \right\}$ are uploaded. After sparse operation, we acquire

$$v_i^t = \bar{v}_i^t - \Delta \bar{v}_i^{t-1}, i \in [1, N]. \quad (10)$$

D. Model Aggregation Phase

After the model training phase of the t^{th} communication round, the local model w_i^t and the intermediate model v_i^t are acquired. To extract dynamic role differences in model layers, we aggregate w_i^t and v_i^t using layer aggregation and an AWC mechanism. Unlike model aggregation as a whole, layer aggregation considers the roles of different layers in the model. Furthermore, AWC uses the chain rule of differentiation to adaptively update aggregation weights, which reflect the dynamic changes of each layer's roles. Next, we describe the details of this phase.

In the i^{th} IoT device, let $w_i^t = (w_{i,1}^t, \dots, w_{i,P}^t)$, $v_i^t = (v_{i,1}^t, \dots, v_{i,P}^t)$, and $\bar{v}_i^t = (\bar{v}_{i,1}^t, \dots, \bar{v}_{i,P}^t)$ denote each layer of the local model w_i^t , the intermediate model v_i^t , and the customized model $\bar{v}_i^t$. Here, P is the total number of layers in these models. $w_{i,h}^t$, $v_{i,h}^t$, and $\bar{v}_{i,h}^t$ denote the h^{th} layer of w_i^t , v_i^t , and $\bar{v}_i^t$, respectively. The customized model $\bar{v}_i^t$ can be formulated as follows:

$$\bar{v}_i^t = v_i^t \odot \alpha_i^t + w_i^t \odot (1 - \alpha_i^t), \quad (11)$$

where $\odot$ is the hadamard product. $\mathbf{1}$ represents the unit vector with the same dimension as α_i^t . $\alpha_i^t = (\alpha_{i,1}^t, \dots, \alpha_{i,P}^t) \in \mathbb{R}^{1 \times P}$ is the aggregation weight vector, in which $\alpha_{i,h}^t \in [0, 1]$ is the h^{th} layer aggregation weight of v_i^t .

According to Eq.(11), it is necessary to balance the aggregation weight vector α_i^t of the local model and the intermediate model when training the customized model. Since we fail to identify the differences between IoT device data distribution and global data distribution, we cannot leverage prior knowledge to

Algorithm 1 : EE-pFL

Input: E total communication rounds, N IoT devices, t the current communication round, P the total layers of the model, w^t the global model, w_i^t the local model, v_i^t the intermediate model, Q periodic calculation interval of aggregation weights for all layers, k retained layers of the model

Output: $(\bar{v}_1, \bar{v}_2, \dots, \bar{v}_N)$ the customized models

```

1: for  $t = 1, 2, \dots, E$  do
2:   IoT device:
3:   Define  $S_t \subseteq \{1, 2, \dots, N\}$ 
4:   Each IoT device  $i \in S_t$  in parallel do
5:     Initialize local model and intermediate model  $w_i^t \leftarrow w^t$ 
       and  $v_i^t \leftarrow \bar{v}_i^{t-1}$ 
6:     Initialize aggregation weights  $\alpha_i^t \leftarrow \alpha_i^{t-1}$ 
7:     Sort  $(\alpha_{i,1}^{t-1}, \dots, \alpha_{i,P}^{t-1})$  and acquire  $v_i^{t,TopK}$ 
8:      $w_i^t = w_i^t - \eta \nabla_{w_i^t} \mathcal{L}_{ICF}(w_i^t)$ 
9:     adopt the threshold-based sparsification strategy to selec-
       tively update  $w_i^t$ 
10:     $v_i^t = v_i^t - \eta \nabla_{v_i^t} \mathcal{L}_{CE}(\bar{v}_i^t)$ 
11:    adopt the threshold-based sparsification strategy to selec-
       tively update  $v_i^t$ 
12:    if  $t \% Q == 0$  then
13:      for  $p = 1, 2, \dots, P$  do
14:         $\alpha_{i,p}^t = \alpha_{i,p}^t - \eta (v_{i,p}^t - w_{i,p}^t) \nabla_{\alpha_{i,p}^t} f_{i,p}(v_{i,p}^{t-1})$ 
15:      end for
16:    end if
17:    if  $t \% Q \neq 0$  then
18:      for  $p = 1, 2, \dots, P$  do
19:        if  $v_{i,p}^t \notin v_i^{t,TopK}$  then
20:           $\alpha_{i,p}^t = \alpha_{i,p}^t - \eta (v_{i,p}^t - w_{i,p}^t) \nabla_{\alpha_{i,p}^t} f_{i,p}(v_{i,p}^{t-1})$ 
21:        end if
22:      end for
23:    end if
24:     $\bar{v}_i^t = v_i^t \odot \alpha_i^t + w_i^t \odot (1 - \alpha_i^t)$ 
25:    Communicate  $\Delta w_i^t$ 
26:    Cloud server:
27:     $w^{t+1} = w^t + \frac{\eta_g}{|S_t|} \sum_{i=1}^{|S_t|} \Delta w_i^t$ 
28:    Communicate  $w^{t+1}$   $N$  IoT devices
29:  end for
30: Return  $(\bar{v}_1, \bar{v}_2, \dots, \bar{v}_N)$ 

```

determine each IoT device's aggregation weight. In light of this, we propose the AWC mechanism. Specifically, Eq.(11) shows that the customized model $\bar{v}_i^t$ is the function of α_i^t . Therefore, we employ the chain rule of differentiation to update and find the optimal value of α_i^t . The updating process of α_i^t is as follows:

$$\alpha_i^t = \alpha_i^{t-1} - \eta (v_i^t - w_i^t) \nabla_{\alpha_i^t} f_i(\bar{v}_i^{t-1}), \quad (12)$$

where $\nabla_{\alpha_i^t} f_i(\cdot)$ denotes the stochastic gradient of α_i^t . Eq.(12) shows that the aggregation weight vector α_i^t is updated relied on the discrepancies between the local model and the intermediate model. If the local model deviates from the intermediate model,

α_i^t changes, thus adjusting the balance between the former and the latter.

Since we need to calculate the aggregation weights of all layers in the layer aggregation, EE-pFL incurs significant computation load compared to traditional FL such as the FedAvg method [9]. In light of this, a TopK strategy is introduced into the AWC. The key idea of the TopK is to heuristically retain the partial layers of the intermediate model based on the top k aggregation weights. These preserved layers are directly included as a part of the customized model without participating in layer aggregation, thus reducing the computational cost of aggregation weights for k layers. Specifically, by adopting the aggregation weight vector $\alpha_i^t = (\alpha_{i,1}^t, \dots, \alpha_{i,P}^t) \in \mathbb{R}^{1 \times P}$ for all P layers, we arrange the elements of α_i^t in descending order and select the corresponding top k weights.

$$v_i^{t,TopK} = TopK(v_{i,1}^t, \dots, v_{i,k}^t \mid \alpha_{i,1}^t, \dots, \alpha_{i,k}^t), \quad (13)$$

where $TopK$ denotes the TopK mechanism mentioned above. k is the hyperparameter, preserving the top k layers in the intermediate model. Algorithm 1 describes the entire workflow of EE-pFL.

III. EXPERIMENTS

A. Experimental Setup

1) *Datasets*: We evaluate EE-pFL on two image classification datasets CIFAR-100 and Tiny-ImageNet, and a text classification dataset AG News, respectively. The CIFAR-100 and Tiny-ImageNet datasets contain 100 and 200 classes, respectively. The AG News dataset covers 4 classes. The three datasets all have the same amount of samples in each class. Two non-IID scenarios are taken into account: 1) a pathological heterogeneous setting, in which each IoT device randomly selects 10 and 20 classes from the CIFAR-100 and Tiny-ImageNet datasets as their private datasets, respectively, and 2) a practical heterogeneous setting. We simulate a real non-IID scenario through the Dirichlet function ($Dir()$) [16].

2) *Benchmarks*: We compare the performance of EE-pFL and advanced benchmarks. These advanced benchmarks contain FedAvg [9], Ditto [10], FedAMP [11], FedFomo [12], PartialFed [13], FedALA [14], and YOGA [15].

3) *Model Selection*: For the CIFAR-100 dataset, we apply the CNN model with two Conv layers and two FC layers to verify the performance of EE-pFL and benchmarks. For the Tiny-ImageNet dataset, a CNN model with two Conv layers and two FC layers is employed to confirm the performance of EE-pFL and the benchmarks. To verify the performance of EE-pFL and the benchmarks on large-scale models, we also introduce the ResNet-18 model in the Tiny-ImageNet dataset. For the AG News dataset, we use the FastText model with one feature embedding layer, one hidden layer formed by one FC, and one output layer formed by one FC.

4) *Parameter Settings*: In the experiments conducted on the CIFAR-100, Tiny-ImageNet, and AG News datasets, the communication round E is 500. CNN, ResNet-18, and FastText are set to retain default layers with $k = 1$, $k = 3$, and $k = 1$, respectively, and the batch size is 10. We set the learning rate

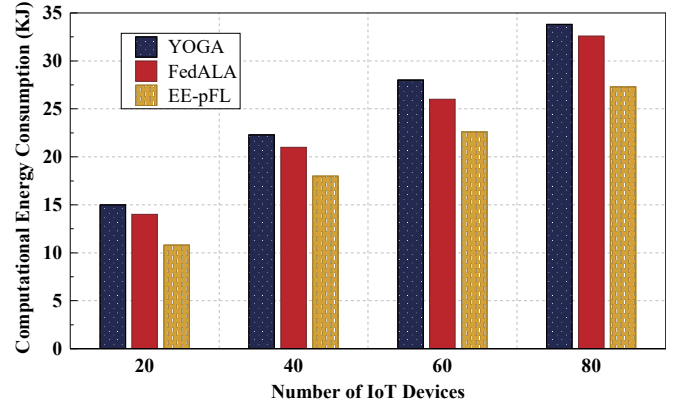


Fig. 3. Computational energy consumption on various number of IoT devices.

η to 0.005, while η in ResNet-18 is 0.1. The number of IoT devices N defaults to 20. The aggregation weights of all layers in these three models are initialized to 0.5 for all IoT devices. For each communication round, the default parameter of the Dirichlet function β is 0.1 in the practical heterogeneous setting. Additionally, we utilize the average classification accuracy of the customized model on each IoT device as a performance evaluation metric.

B. Experimental Results Analysis

1) *Computational Energy Consumption Comparison*: To evaluate the computational energy consumption of EE-pFL, we compare the computational energy consumption of EE-pFL and several benchmarks (e.g., PartialFed, FedALA, YOGA, and EE-pFL) across different numbers of IoT devices. The experimental results are shown in Fig. 3. Some conclusions can be drawn. 1) As the number of IoT devices arises, the computational energy consumption of EE-pFL and these benchmarks increases. The underlying reason is that model training is performed on the IoT devices. If more IoT devices participate in training, the overall computation load increases accordingly. This brings about more computational energy consumption. 2) Compared with these benchmarks, EE-pFL has the lowest computational energy consumption across various numbers of IoT devices. This reveals that the threshold-based sparsification strategy can effectively reduce the computational energy consumption through selectively updating model parameters.

2) *Classification Accuracy Comparison*: On the CIFAR-100, Tiny-ImageNet, and AG News datasets, the average classification accuracy of EE-pFL and the advanced benchmarks in the two non-IID scenarios are contrasted in TABLE I. "Tiny" represents using the four-layer CNN on the Tiny-ImageNet dataset. "Tiny*" denotes utilizing ResNet-18 on the Tiny-ImageNet dataset. In the second non-IID scenario, the default parameter of the Dirichlet function β is set to 0.1. We draw the following conclusions in the two non-IID scenarios. 1) The classification accuracy of layer-level aggregation methods (e.g., PartialFed, FedALA, YOGA, and EE-pFL) outperforms the model-level aggregation (e.g., FedAvg, Ditto, FedAMP, and FedFomo) on various datasets. The potential reason for this difference is

TABLE I
THE CLASSIFICATION ACCURACY (%) IN THE PATHOLOGICAL
HETEROGENEOUS SETTING (PHS) AND PRACTICAL HETEROGENEOUS
SETTING. AG DENOTES AS THE AG NEWS DATASET.

Setting Methods	phs		practical heterogeneous setting			
	CIFAR-100	TINY	CIFAR-100	TINY	TINY*	AG
FedAvg	25.98	14.20	31.98	19.46	19.45	79.57
Ditto	67.23	40.23	52.87	32.15	35.92	95.45
FedAMP	64.34	37.15	42.69	27.99	29.11	92.18
FedFomo	62.49	35.87	45.39	30.33	32.84	95.84
PartialFed	65.35	37.76	51.37	32.78	36.91	94.87
FedALA	67.82	39.48	57.44	36.75	41.89	96.46
YOGA	66.25	38.69	55.23	37.65	38.19	95.32
EE-pFL	68.76	42.16	58.19	39.38	42.42	96.93

that the former provides a fine-grained aggregation strategy to consider the roles of different layers, thus improving the classification accuracy. 2) In layer-level aggregation methods, the classification accuracy of FedALA and YOGA is superior to PartialFed. The underlying reason is that FedALA can extract the needed information in the global model at the element level, which is more accurate than the binary selection and layer-level weight learning in PartialFed. Meanwhile, Based on the data distribution and communication ability of IoT devices, YOGA develops an appropriate layer aggregation scheme for IoT devices through a model named Max-Match. Unfortunately, PartialFed fails to consider data distribution and communication capacity simultaneously when devising a layer aggregation scheme. 3) The classification accuracy of EE-pFL outperforms FedALA and YOGA on different datasets. This reveals that AWC adaptively updates the aggregation weights in accordance with the dynamic role differences in model layers.

4) In contrast to the four-layer CNN, ResNet-18 is regarded as a deep model with more layers. Although most methods exhibit higher classification precision with ResNet-18, EE-pFL still performs best, proving that our proposed method is also adaptable to a deep model.

IV. CONCLUSION

We have presented a novel pFL framework named EE-pFL for establishing Green IoT. In particular, each IoT device has generated a customized model through a model training phase and a model aggregation phase. In the former phase, to reduce the computational energy consumption of IoT devices, we have introduced a threshold-based sparsification strategy. In the latter phase, we have integrated layer aggregation and the AWC. We have proposed layer aggregation to aggregate each layer of the local model and the intermediate model with aggregation weights to produce an individual customized model for each IoT device. We also have designed the AWC to adaptively

update aggregation weights based on dynamic role differences of model layers. The experimental results have confirmed the effectiveness of EE-pFL on computational energy consumption and classification accuracy.

REFERENCES

- [1] Y. Chen, T. Zhao, P. Cheng, M. Ding, and C. W. Chen, "Joint front-edge-cloud iotv analytics: Resource-effective design and scheduling," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 23 941–23 953, 2022.
- [2] M. Du, H. Zheng, X. Feng, Y. Chen, and T. Zhao, "Decentralized federated learning with markov chain based consensus for industrial iot networks," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 4, pp. 6006–6015, 2023.
- [3] Z. Zhang, C.-H. Lung, X. Wei, M. Chen, S. Chatterjee, and Z. Zhang, "In-network caching for icn-based iot (icn-iot): A comprehensive survey," *IEEE Internet of Things Journal*, vol. 10, no. 16, pp. 14 595–14 620, 2023.
- [4] R. Goyat, G. Kumar, M. Conti, T. Devgun, R. Saha, and R. Thomas, "Benigreen: Blockchain-based energy-efficient privacy-preserving scheme for green iot," *IEEE Internet of Things Journal*, vol. 10, no. 18, pp. 16 480–16 493, 2023.
- [5] A. Huang, L. Li, L. Zhang, Y. Niu, T. Zhao, and C.-W. Lin, "Multi-view graph embedding learning for image co-segmentation and co-localization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 6, pp. 4942–4956, 2024.
- [6] H. Liao, N. Jiang, W. Chen, H. Wei, and T. Zhao, "Distillation-based utility assessment for compacted underwater information," *IEEE Signal Processing Letters*, vol. 31, pp. 481–485, 2024.
- [7] J. Chen, Y. Zhu, W. Huang, C. Lan, and T. Zhao, "Scale-adaptive asymmetric sparse variational autoencoder for point cloud compression," *IEEE Transactions on Broadcasting*, vol. 70, no. 3, pp. 884–894, 2024.
- [8] B. Zhang, Y. Mao, X. He, H. Huang, and J. Wu, "Balancing privacy and accuracy using significant gradient protection in federated learning," *IEEE Transactions on Computers*, 2024.
- [9] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.
- [10] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *International conference on machine learning*. PMLR, 2021, pp. 6357–6368.
- [11] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang, "Personalized cross-silo federated learning on non-iid data," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 9, 2021, pp. 7865–7873.
- [12] M. Zhang, K. Sapra, S. Fidler, S. Yeung, and J. M. Alvarez, "Personalized federated learning with first order model optimization," in *International conference on Learning Representations (ICLR)*, 2021.
- [13] B. Sun, H. Huo, Y. Yang, and B. Bai, "Partialfed: Cross-domain personalized federated learning via partial initialization," *Advances in Neural Information Processing Systems*, vol. 34, pp. 23 309–23 320, 2021.
- [14] J. Zhang, Y. Hua, H. Wang, T. Song, Z. Xue, R. Ma, and H. Guan, "Fedala: Adaptive local aggregation for personalized federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 9, 2023, pp. 11 237–11 244.
- [15] J. Liu, J. Liu, H. Xu, Y. Liao, Z. Wang, and Q. Ma, "Yoga: Adaptive layer-wise model aggregation for decentralized federated learning," *IEEE/ACM Transactions on Networking*, 2023.
- [16] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," *Advances in neural information processing systems*, vol. 33, pp. 2351–2363, 2020.



DSSPACE

<https://dspace.org/>

Energy-efficient personalized Federated Learning for establishing Green IoT

Xu, Haowen; Mao, Yingchi; Zheng, Haotian; He, Xiaoming; Rong, Yi ...

2025-06-08

Institute of Electrical and Electronics Engineers (IEEE)

<https://doi.org/10.1109/icc52391.2025.11161693>

Xu H, Mao Y, Zheng H, et al., (2025) Energy-efficient personalized Federated Learning for establishing Green IoT. In: IEEE International Conference on Communications, 8-12 June 2025, Montreal, Canada, pp. 5175-5180