IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 July 2024, accepted 22 July 2024, date of publication 31 July 2024, date of current version 15 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3436009

RESEARCH ARTICLE

Heterogeneous Privacy Level-Based Client Selection for Hybrid Federated and Centralized Learning in Mobile Edge Computing

FARANAKSADAT SOLAT[1], SAKSHI PATNI[1], SUNHWAN LIM[2], AND JOOHYUNG LEE[1], (Senior Member, IEEE)
[1]Department of Computing, Gachon University, Seongnam 13120, Republic of Korea
[2]Electronics and Telecommunications Research Institute (ETRI), Daejeon 34129, Republic of Korea

Corresponding authors: Sunhwan Lim (shlim@etri.re.kr) and Joohyung Lee (j17.lee@gachon.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea Government, Ministry of Science and ICT (MSIT) (RS-2024-00423362, Collaborative Perception and Intelligence Framework for Hyper-connected Interaction among Human and Intelligent Things) and in part by the Gachon University research fund of 2022 (GCU-202208790001).

ABSTRACT To alleviate the substantial local training burden on clients in the federated learning (FL) process, this paper proposes a more efficient approach based on hybrid federated and centralized learning (HFCL), leveraging the Mobile Edge Computing (MEC) environment within wireless communication networks. Considering the existence of heterogeneous data types with different privacy levels -such as 1) sensitive data, which can not be exposed, and 2) less-sensitive data, which can be exposed for centralized learning (CL)-we formulate an optimization problem aimed at achieving a balance between 1) total latency, including computation and communication, and 2) the training burden on the MEC server. This balance is achieved by adjusting the set of participants in FL, taking into account client selection under different privacy levels. A multi-objective optimization problem is designed using mixed-integer nonlinear programming, which is generally recognized as NP-hard. We employ relaxation techniques in combination with the Mutas & Simulated Annealing Heuristic algorithm to develop a near-optimal yet practical algorithm. Our numerical and simulation results reveal that the proposed scheme effectively achieves a global model by striking a balance between the total time required for model convergence and the computational load on the MEC server. Furthermore, experimental results on three well-known real-world datasets demonstrate that the proposed scheme maintains an acceptable level of accuracy and loss.

INDEX TERMS Federated learning, centralized learning, mobile edge computing.

I. INTRODUCTION

The proliferation of training data sourced from diverse Internet of Things (IoT) devices, such as wearable gadgets like Google Glass, Samsung Watch, and Apple Watch, which capture sensitive user activity data [1], has led to an unprecedented surge in data volumes. Projections indicate that by 2023, there will be an estimated 30 billion network devices generating a staggering 100 zettabytes of distributed data [2], [3], [4]. This exponential growth has raised concerns about the scalability of centralized training methods and the associated privacy risks resulting from data exposure. To address these challenges, federated learning (FL) has emerged as an innovative and distributed learning paradigm [5]. FL adopts a more generic approach by ‘‘bringing the code to the data’’ rather than the traditional method of transferring data to Mobile Edge Computing (MEC) servers [6]. However, FL encounters significant

The associate editor coordinating the review of this manuscript and approving it for publication was Tiago Cruz.

challenges due to the inherent heterogeneity among clients, encompassing differences in computing capabilities, network conditions, dataset sizes, and data distributions. In this context, it is challenging to achieve optimal FL performance in terms of convergence speed and accuracy [7].

Addressing this challenge has prompted extensive research efforts in the literature, all aimed at optimizing FL operations. Notably, one prevalent approach involves the integration of MEC to assist FL by harnessing the computing capabilities available at the network edge and reducing the physical distance to MEC servers, which has gained significant attention [8]. Within this context, various optimization techniques have been explored to enhance accuracy and convergence speed. These optimizations encompass the control of hyperparameters related to FL clients such as client selection, clustering as well as the management of radio and computation resources (including CPU/GPU usage) [9], [10], [11], [12], [13], [14], [15]. Furthermore, recognizing the substantial computational resources demanded by FL for local training on clients, there has been a recent surge of interest in hybrid federated and centralized learning (HFCL), which leverages centralized learning (CL) to boost FL performance. More specifically, in HFCL, clients with limited computational power send their raw data to a centralized server for CL, whereas clients with greater computational capability train their models locally, adhering to the FL process.

To date, there have been relatively few works within this HFCL field, mainly because it represents a relatively new and evolving field [16]. Additionally, integrating CL with FL poses challenges, primarily due to privacy concerns that make the coexistence of these two approaches intricate. Nonetheless, by making the assumption that clients possess less-privacy-sensitive data, which can be suitable for either FL or CL, we can harness both the computing power of a MEC server and the distributed computing resources offered by clients participating at FL in the context of HFCL. To the best of our knowledge, the majority of prior research in this field has primarily assumed uniform privacy levels for all data, which represents a limitation of previous approaches. In this context, the recent work presented in [17] proposed an optimization problem based on HFCL for client selection in FL or CL with the goal of minimizing total latency in the presence of both privacy-sensitive and privacy-insensitive clients over wireless networks. This work closely aligns with our own research. However, it does not account for striking a balance between total latency and the computational load imposed on the MEC server. Moreover, the optimization involving integer values can be further enhanced during the transition from continuous to binary representation by leveraging sophisticated heuristics.

To address this limitation and optimize HFCL performance, our study considers the presence of heterogeneous data types characterized by different privacy levels. Correspondingly, we categorize data into two types: i) sensitive data suitable for FL and ii) less-sensitive data suitable for either FL or CL. In this context, we formulate an optimization problem aimed at striking a balance between i) total latency, encompassing both computation and communication, and ii) the training burden placed on the MEC server, all achieved by adjusting the set of participants in FL. The detailed contributions of this article are summarized as follows:

- Under the HFCL by taking into account varying privacy levels of datasets across clients, we introduce a multi-objective optimization problem to strike a balance between two key factors: i) total latency, encompassing both computation and communication times, and ii) the training load imposed on the MEC server. It is achieved through the dynamic adjustment of participants in FL, and taking into account client selection under different privacy levels.
- Specifically, rigorous analytical models for deriving the total latency over HFCL with respect to the FL participation of clients are provided, which include both computation and communication times required in the FL and CL process.
- Utilizing these analytical models, we formulate the optimization problem using mixed-integer nonlinear programming—a widely recognized NP-hard problem. To address this challenge, we employ an optimization technique, transforming the problem into a linear programming form. For additional enhancements in optimizing the continuous-discrete mapping, we adopt the Mutas & Simulated Annealing Heuristic algorithm. By combining these approaches, the proposed scheme, which is named HFCLX, offers a near-optimal yet practical solution.
- Our numerical and simulation findings demonstrate that our HFCLX scheme effectively obtains a global model. It strikes a balance between the total time required for model convergence and the computational latency on the MEC server. Moreover, our experiments conducted on three widely recognized real-world datasets validate the effectiveness of the HFCLX scheme. These experiments considered both Independent and Identically Distributed (IID) and non-IID data scenarios. This is notably noticeable in the MNIST dataset, Fashion-MNIST dataset, and CIFAR-10 dataset for 500 rounds of training. The accuracy reached impressive levels for both IID and non-IID data pairs in each respective dataset. Additionally, we leverage Differential Privacy (DP) to enhance privacy protection in our HFCLX scheme. By adding Gaussian noise to the less sensitive raw data in CL scenario before it is sent to the centralized server to ensure the client privacy protection. Our theoretical analysis and empirical results show that applying DP effectively balances privacy protection and model accuracy.

The remainder of this article is organized as follows: In Section II, we discuss related studies. In Section III, we present the system model for our proposed HFCLX. Section IV outlines the problem formulation that seeks to strike a balance between minimizing total latency and managing the training load imposed on the MEC server. We also discuss the corresponding algorithm. In Section V, we conduct an in-depth performance evaluation of the proposed HFCLX, comparing it with existing works to demonstrate its effectiveness. Finally, in Section VI, we offer concluding remarks on our research. Table 2 provides a compilation of essential symbols defined and employed in this paper.

II. RELATED WORKS

In this section, we present a brief overview of the existing literature concerning CL, FL, and HFCL. Classical CL faces significant privacy threats related to sensitive information and scalability issues. To address these challenges, FL has gained widespread attention in the literature for its emphasis on distributed learning over clients. However, it's worth noting that FL can impose a substantial computational burden on clients during the local training process.

A. THE FUNDAMENTAL BACKGROUND AND CONCEPT OF FL

CL also known as centralized machine learning (ML), is a traditional approach where data from various sources is collected and stored in a centralized location, often a single MEC server or data center. In this setup, a single model is trained on the entire dataset, and the resulting model is then deployed to make predictions or decisions. CL offers advantages in terms of ease of implementation, model quality control, and efficient resource utilization. However, it requires aggregating data into a centralized location. This can raise concerns about data privacy and security. Additionally, it may create potential computing and networking bottlenecks at the MEC server when dealing with large datasets. In this regard, FL is a more recent approach that aims to address some of the privacy and scalability challenges associated with CL. In FL, multiple devices or edge nodes as clients are participating FL process have their own local datasets. Instead of sending all the data to a MEC server [5], the global model is initially sent to the clients. The clients then perform local training using their own local dataset. Then, only the updated local model parameters are sent back to the MEC server. The MEC server aggregates these updates to improve the global model [18]. In FL, the raw data remains on the clients, enhancing privacy and reducing the need to transfer massive amounts of data.

B. EFFICIENT FL MANAGEMENT

There are several existing studies for efficient FL management including local model personalization [19], computing resource allocation [20], [21], [22], [23], [24], radio resource allocation [25], [26], [27], privacy-preserving [2], [19], latency minimization [1], [2], [10], [28], [29], and clustering clients [11], [30], [31]. Specifically, the work in [19] introduced the concept of the Privacy-preserving Federated Adaptation (PFA) as a privacy-preserving personalization technique limited to a single device. The idea behind the PFA method is to leverage the sparsity of neural networks (NNs) to create a privacy-preserving mechanism. This mechanism can be used to replace the raw data for the group of clients during the adaptation process. In [28], a joint communication and computation optimization problem aims to minimize the delay for the FL approach. The authors in [29] proposed a novel mechanism that optimizes communication, computation, and caching configurations in MEC servers. This mechanism aims to minimize the mean latency experienced by mobile devices. In [32], authors proposed a joint mathematical optimization model for client selection and computing resource allocation to perform model training in every iteration. They used 5G slicing services to transform the non-convex optimization problem to convex and then they solved the problem into a convex one. The problem was then solved using the successive convex approximation (SCA) method where 5G slicing services refer to the ability of 5G networks to create customized, virtualized network slices to cater to specific requirements of different applications, users, or services. The work presented in [11] proposed the FedGM, a method that jointly optimizes the creation of groups among MEC servers and the subsequent group associations based on clustering. This approach aims to enhance the convergence time during the FL process, particularly for generating mobile traffic prediction models. The authors employ a genetic algorithm to address this non-convex problem. While recent advancements in privacy-preserving and trust mechanisms for federated learning have made significant strides, there remain distinct gaps that our proposed HFCLX scheme aims to address. The PPRU scheme [33], focusing on vehicular networks, leverages cryptographic techniques to ensure data privacy and manage reputations, emphasizing the importance of privacy in networked systems. However, it does not tackle the challenges of optimizing client selection and computational load balancing in mobile edge computing environments. Similarly, TFL-DT [34] introduces a trust evaluation framework for federated learning within digital twin environments, ensuring the reliability of data and model updates. Despite its focus on trust, it does not explicitly address the varying privacy levels of data and the optimization needed to balance latency and computational load. Furthermore, methods aimed at preventing backdoor attacks in federated learning primarily enhance security but do not inherently solve the latency and load balancing challenges crucial for efficient federated learning [35].

C. THE COMPARISON WITH STATE OF THE ART

An in-depth analysis of CL, FL, and HFCL is provided below:

- CL: In CL, the clients transfer the raw data to the centralized server for non-parallel centralized training. Then the centralized server aggregates the raw data to update the global model. In this scenario, transferring the raw data to the centralized server causes privacy issues in sensitive data.

TABLE 1. Comparison of CL, FL, HFCL, and Proposed HFCLX.

	CL	FL	HFCL	Proposed HFCLX
Privacy issues in sensitive data	X	O	O	O
Parallel training	X	O	O	O
Centralized training	O	X	O	O
Heterogeneous clients	X	O	O	O
Client selections	X	O	O	O
Balance between Total Latency and Workload on the MEC server	X	X	X	O

- FL: Unlike CL, FL is the distributed training that should handle the heterogeneous clients' issue who train their datasets locally in parallel, and transmit their updated local models to the centralized server for model aggregation and updating the global model. However, there is room to discuss related to heterogeneous clients as well as client selection issues.
- HFCL: To address the above issues, despite CL and FL, in HFCL, the heterogeneous clients can train centralized as well as distributed training by considering the idea of both CL and FL where some clients with low computational capability transmit the raw data to the centralized server for server training. While the other clients with high computational capability, train based on FL scenario. Finally, the centralized server trains the raw data and updates the global model by using the aggregated local models that are received from clients and the trained model. However, there is room to discuss about balancing the total latency and workload on the centralized server.

Consequently, in an effort to harness the advantages of both FL and CL while mitigating privacy concerns, HFCL has emerged as a promising approach, which utilizes both CL and FL by considering the clients - in this scenario based on data sensitivity- as active or inactive, depends on their computational capability [36]. Table 1 presents a comparison of CL, FL, HFCL, and proposed HFCLX. Notably, the proposed HFCLX demonstrates a superior ability to balance total latency and workload on the MEC server.

HFCL is a machine learning and data analysis approach that combines elements of both FL and CL to optimize model training and data processing. It recognizes that while FL addresses data privacy and distribution challenges, there might still be benefits to centralizing some aspects of the learning process. In [10], the edge-assisted FL (EAFL) framework was introduced. The framework demonstrated its effectiveness through integrated design and latency minimization by offloading data, and it showed adaptability across diverse scenarios. The hyperparameter-based offloading strategies were formulated to mitigate acceptable latency within the EAFL framework. Nonetheless, when data is offloaded across all clients without considering data sensitivity, it can lead to data leakage and increased latency due to limited bandwidth. Furthermore, in [37] the HFCL framework was presented for collaborative edge device-based model training. To address latency challenges, the Sequential Dataset Transmission (SDT) approach was proposed, enhancing performance over FL while maintaining lower communication overhead than CL. Moreover, the study presented in [38] designed a novel HFCL framework over all MEC systems. This framework effectively balances computation efficiency and communication cost and enhances model accuracy. Additionally, the recent study presented in [17] introduced an optimization problem based on HFCL for client selection in FL or CL with the goal of minimizing total latency in scenarios involving both privacy-sensitive and privacy-insensitive clients across wireless networks. The work in [39] investigated the cost of centralized versus distributed learning in terms of transmission and processing delay. Based on this analysis, they also proposed a hybrid approach tailored to satellite networks equipped with cloud-centralized servers. This approach leverages both centralized and distributed methods, adapting to device scenarios to optimize learning strategies with a focus on minimizing transmission delay, achieved through the utilization of Deep Q-Networks (DQN). Our HFCLX scheme fills these gaps by integrating hybrid federated and centralized learning, optimizing client selection based on heterogeneous privacy levels, and employing advanced optimization techniques to reduce latency and balance the computational burden on MEC servers, thus providing a comprehensive solution that enhances both efficiency and security in MEC system.

III. SYSTEM MODEL

As depicted in Fig. 1, there is a set of clients denoted as $\mathcal{K}$, where $|\mathcal{K}| = K$ denotes the total number of clients. Each client is directly linked to the base station (BSs) associated with a MEC server through a wireless network. For each $k \in \mathcal{K}$, each client k has its own local dataset denoted as $\mathcal{D}_k$ where the size of local dataset $|\mathcal{D}_k|$ is D_k. In the dataset, a data sample i usually consists of the input vector x_i (e.g. the pixels of an image) and the output scalar y_i (e.g. the label of the image). In the proposed framework, clients with sensitive data perform FL, while the remaining clients with less sensitive data may participate in either FL or CL. This decision is determined by the HFCLX scheme module, aiming to strike a balance between the total latency required for model convergence and the computational load imposed on the MEC server. To deal with latency minimization and alleviate the burden on the MEC server problems, MEC-based FL has been proposed for data training with low latency by exploiting resources at the network edge near the data sources. In the HFCLX scheme, the reduction of convergence total time is addressed by defining the FL client selection problem, which takes into account various groups of clients characterized by distinct privacy concerns. The specific steps of the proposed HFCLX are outlined as follows:

- Step 1: **Group Selection**- The initial step involves dividing clients into two distinct groups based on the

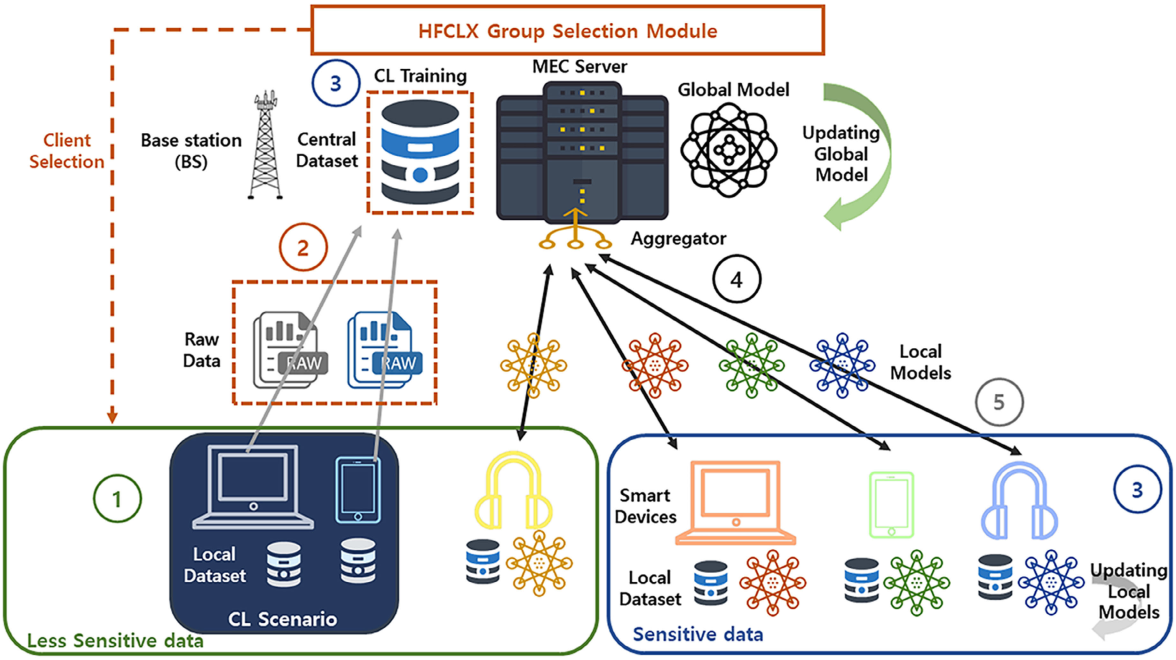

FIGURE 1. HFCLX scheme.

sensitivity of their data: i) clients with sensitive data suitable for FL and ii) clients with less sensitive data suitable for either FL or CL. Within the less sensitive group, we formulate a multi-objective optimization problem that seeks to strike a balance between i) total latency, including computation and communication, and ii) the training burden on the MEC server, achieved by adjusting the set of participants in FL, taking into account client selection under different privacy levels. In this context, we introduce the variable α_k as a binary indicator to represent data sensitivity. Specifically, if $\alpha_k = 1$, it denotes clients with sensitive data; conversely, $\alpha_k = 0$ represents clients with less-sensitive data. Notably, we assume that all clients reveal their data sensitivity by providing their α_k values to the MEC server. Consequently, with given values of α_k for all clients, we can determine the participation of clients with less-sensitive data in the FL approach using the binary variable β_k, which requires optimization. This optimization is conducted by our proposed HFCLX scheme. The HFCLX group selection module within the MEC server manages this role.

- Step 2: **CL Approach**- In this step, clients categorized under the CL group participate by transmitting their raw datasets to the MEC server as their data falls within the category of less sensitive data.
- Step 3: **FL Approach and MEC Server**- Following the data transfer from the CL clients to the MEC server, FL clients execute local training. Here, the total loss function at client k participating in FL process is as follows

$$F_k(w_k) = \frac{1}{D_k} \sum_{i \in \mathcal{D}_k} f_i(w_k). \tag{1}$$

where w_k denotes the current local model parameter for client k, and $f_i(w_k)$ is the local loss function for client k at data sample i. Simultaneously, the MEC server conducts training using the raw data it has received from the CL clients. Here, the total loss function at MEC server is as follows

$$F_s(w_s) = \frac{1}{|\mathcal{D}_s|} \sum_{i \in \mathcal{D}_s} f_i(w_s), \tag{2}$$

where $\mathcal{D}_s$ denotes the aggregated dataset received by CL clients, and w_s is the local model parameter for MEC server. $f_i(w_s)$ is the local loss function for MEC server at data sample i. In this context, $|\mathcal{D}_s| = \sum_{k=1}^{K}(1-\alpha_k)(1-\beta_k)D_k$. Here, the term $(1-\alpha_k)(1-\beta_k)D_k$ represents the amount of dataset of client k for transferring the raw data to the MEC server for training, and it depends on the binary variables α_k and β_k. To clarify, when a client k participates in the CL approach, this term simplifies to D_k when both α_k and β_k are set to 0.

- Step 4: **Parameter Update**- Upon completing their local training, FL clients transmit their updated parameters in the form of local models back to the MEC server during this phase.
- Step 5: **Aggregated Model Update**- The MEC server plays an important role in this step by actively participating in model computation. It collects the model trained using raw data received from CL clients and all local

TABLE 2. Variables and functions table.

Name	Instruction
$\mathcal{K}$	The set of clients
$\lvert\mathcal{K}\rvert = K$	The total number of clients
$\mathcal{D}_k$	The local dataset for each client k
$\lvert\mathcal{D}_k\rvert = D_k$	The size of local dataset
i	The data sample
x_i	The input vector
y_i	The output scalar
$f(w)$	The local loss function
$F_s(w_s)$	The global loss function at the MEC server
w	The global model
w^*	The optimal global model
w_k	The current local model parameter for client k
$f_i(w_k)$	The local loss function for client k
$f_i(w_s)$	The local loss function for MEC server
$F_k(w_k)$	The total local loss function for client k
α_k	The binary variable signifies client participation based on sensitive data
β_k	The binary variable signifies client participation possessing lower data sensitivity
$t^{FL}_{comp,k}$	The computation time for one round per each k client
W_k	The computation of client k per round
τ	The epoch number per round
B	The total bandwidth
H_k	The iteration number (mini-batch number) in one epoch for client k
H_s	The mini-batch size (training data size of one iteration)
H_{ser}	The iteration number (mini-batch number) in one epoch at MEC server for CL clients' aggregated dataset
G	The number of CPU cycles required for training 1-bit data
e_k	The CPU's frequency of client k
t^{FL}_{comm}	The communication time for one round per each k client
t_w	A slot as communication time for local model upload and download per round
$t^{CL}_{comm,k}$	The communication time for each k client
p_k	The transmission power of client k
g_k	The client k's channel gain
t^S_{comp}	The computation time for centralized training over selected CL clients
W_s	The total computation of the MEC server
e_s	The CPU frequency of the MEC server
T_{FL}	The total time for the FL process for one round
T_{CL}	The total time for the CL process for one round
$r(\epsilon)$	The total rounds in FL
e_k	The CPU's frequency of client k
T	The total time
C	The cost function

models generated by FL clients. Subsequently, it updates the aggregated model, integrating insights from both centralized and distributed data sources. According to the FedAvg algorithm defined in [5], the global model parameter w is as follows

$$w = \frac{1}{\sum_{k\in\mathcal{K}} D_k}\Big(\sum_{k\in\mathcal{K}}(\alpha_k + (1-\alpha_k)\beta_k)D_k w_k + |\mathcal{D}_s|w_s\Big). \tag{3}$$

The term $(\alpha_k + (1-\alpha_k)\beta_k)D_k$ represents the amount of dataset of client k for local training, and it depends on the binary variables α_k and β_k. To clarify, when a client k participates in the FL approach, this term simplifies to D_k when either α_k is set to 1, or when α_k is set to 0 and β_k is set to 1.

A. ANALYTICAL MODELS

1) COMPUTATION AND COMMUNICATION TIME FOR FL

Let W_k denote the computational workload on the client k for local training in FL. According to the mini-batch gradient descent algorithm in [5] and [10], W_k can be defined with respect to the size of the dataset. To model the W_k, we define H_k as the iteration number (mini-batch number) in one epoch for client k, which is given by

$$H_k = \frac{(\alpha_k + (1-\alpha_k)\beta_k)D_k}{H_s}, \tag{4}$$

where, H_s is the mini-batch size (training data size of one iteration). Then, W_k is given by

$$\begin{aligned} W_k = \tau H_k G H_s &= \tau\Big(\frac{(\alpha_k + (1-\alpha_k)\beta_k)D_k}{H_s}\Big)GH_s \\ &= \tau G(\alpha_k + (1-\alpha_k)\beta_k)D_k, \end{aligned} \tag{5}$$

where τ is the epoch number per round and the constant G denotes the number of CPU cycles required for training 1-bit data according to [10]. Finally, the computation time for one round per each k client has defined as $t^{FL}_{comp,k}$, which is given by

$$t^{FL}_{comp,k} = \frac{W_k}{e_k}, \tag{6}$$

where e_k is the CPU's frequency of client k.

To simplify the communication time during FL, we assume that the fixed amount of time slot t_w is required for the local upload of the model and the global download of the model to / from the MEC server in each round as in [2]. Then, the communication time for one round at each k client t^{FL}_{comm} is

$$t^{FL}_{comm} = t_w. \tag{7}$$

2) COMPUTATION AND COMMUNICATION TIME FOR CL

We define W_s as the total computational workload, measured in CPU cycles, for the MEC server during the model training phase, where the MEC server operates at a CPU frequency denoted e_s for training purposes. Then, H_{ser} is the iteration number (mini-batch number) in one epoch at MEC server for CL clients' aggregated dataset, which is given by.

$$H_{ser} = \frac{\sum_{k=1}^{K}(1-\alpha_k)(1-\beta_k)D_k}{H_s}. \tag{8}$$

Finally, W_s is formulated as

$$\begin{aligned} W_s = \tau H_{ser} G H_s &= \tau\Big(\frac{\sum_{k=1}^{K}(1-\alpha_k)(1-\beta_k)D_k}{H_s}\Big)GH_s \\ &= \tau G\sum_{k=1}^{K}(1-\alpha_k)(1-\beta_k)D_k \end{aligned}$$

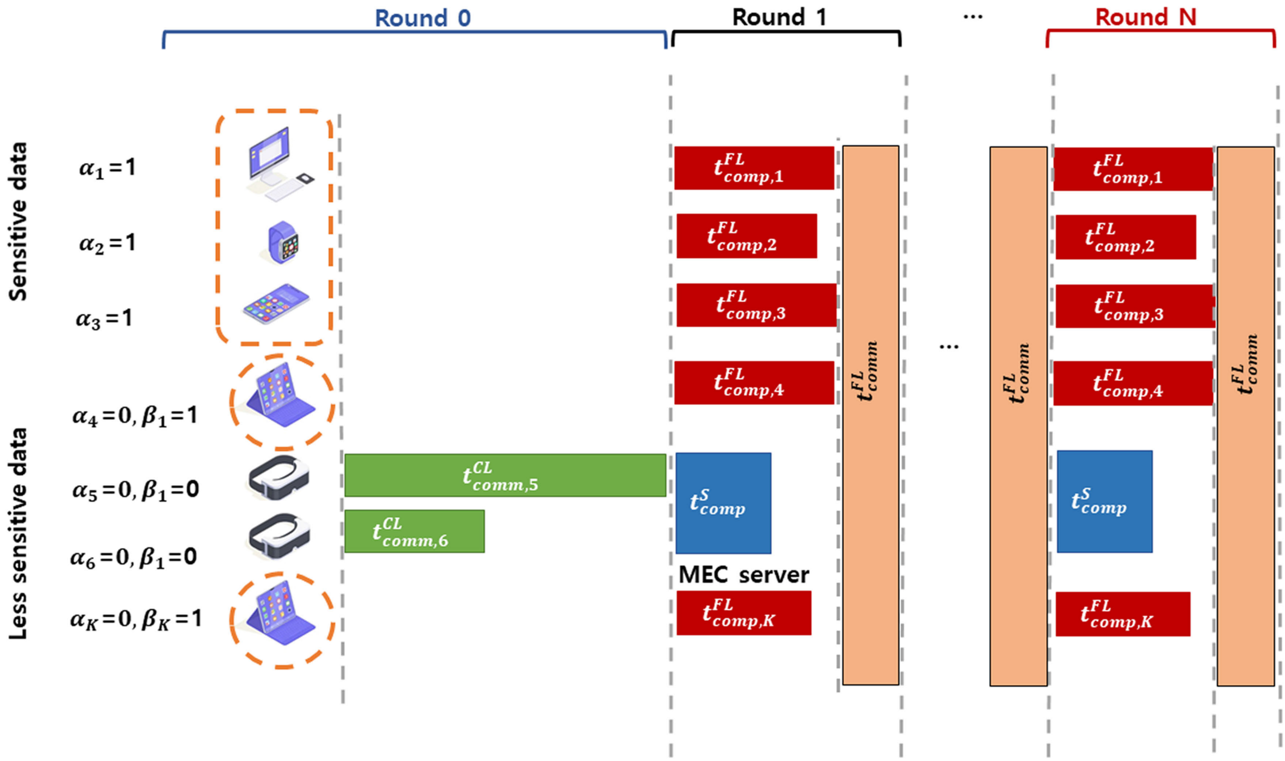

FIGURE 2. **HFCLX time diagram.**

$$= \tau G \sum_{k=1}^{K} (1-\alpha_k)(1-\beta_k)D_k. \tag{9}$$

Then, the computation time for centralized training over selected clients for the CL has been denoted as t^S_{comp} in (10), which is given by

$$t^S_{comp} = \frac{W_s}{e_s}. \tag{10}$$

The communication time for each k client to upload the D_k amount of its local dataset to the MEC server via the wireless link, denoted as $t^{CL}_{comm,k}$, is given by

$$t^{CL}_{comm,k} = \frac{(1-\alpha_k)(1-\beta_k)D_k}{B\log_2(1+p_k g_k)}, \tag{11}$$

where B denotes the total bandwidth and p_k is the transmission power of client k, and g_k denotes client k's channel gain according to [10]. In this context, if α_k is set to 1 (indicating clients with sensitive data) or if α_k is set to 0 and β_k is set to 1 (indicating clients with less sensitive data participating in FL), clients are engaged in the FL approach, which does not require $t^{CL}_{comm,k}$. Conversely, when α_k is 0 and β_k is 1 (indicating clients with less sensitive data participating in CL), they are required to transmit their raw data D_k to the server for the CL approach.

3) TOTAL LATENCY FOR HFCLX

Based on analytical models defined in previous subsections, the total latency for the proposed HFCLX can be formulated. As shown in Fig. 2, at Round 0, to send all local data from selected clients for CL (β_k=0) with less-sensitive data (α_k=0), the required time during Round 0 is given by

$$T^{CL}_{comm} = \max_{k\in\mathcal{K}}[t^{CL}_{comm,k}] \tag{12}$$

From Round 1 to model convergence, FL process is conducted including i) parallel training process over clients and MEC server and ii) model upload and download as depicted in Fig. 2. To formulate the total latency over entire FL process, we firstly define T_{FL} as the total latency for one round of FL process, which is given by

$$T_{FL} = \max_{k\in\mathcal{K}}[t^{FL}_{comp,k}, t^S_{comp}] + t^{FL}_{comm} \tag{13}$$

Finally, as defined in [10], $r(\epsilon)$ is the total required rounds in FL for obtaining the specific training loss ϵ. Then, based on (10) and (11), the total latency T required for achieving a specific training loss ϵ is obtained by

$$T = T^{CL}_{comm} + r(\epsilon)T_{FL} \tag{14}$$

IV. PROPOSED HFCLX SCHEME

In this section, we introduce a novel HFCLX scheme designed to optimize client selection. The primary goal is to strike a balance between two key factors i) total latency (T), encompassing both computation and communication time, and ii) the training load on the MEC server, quantified by the number of CL participants. Subsequently, we outline the algorithm used to address this optimization problem, which is formulated as a non-convex problem, which leverages a combination of relaxation techniques and heuristics to provide a solution.

A. PROBLEM FORMULATION

As previously discussed, the design of the cost function C takes into account two crucial factors. The first factor represents the total latency T as a penalty component, while the second factor represents the number of clients with less sensitive data participating in FL as a positive contribution. The rationale behind the second term is that as the number of clients participating in FL increases, there is a reduction in the computational burden on the MEC server due to the benefits of distributed training. Therefore, the design of the cost function C is intended to strike a balance in the trade-off between total latency and MEC server overload by managing the client selection variable, β_k. As in [40] and [41], we adopt a weighted linear sum method to define the cost function C, which is formulated by

$$C = T - \gamma \sum_{k=1}^{K} \beta_k, \qquad (15)$$

where $\gamma \geq 0$ is the weighting factor of computational load on the MEC server, representing a preference for computational load reduction on the MEC server. A higher value of γ reflects a stronger preference for reducing the computational load on the MEC server. Conversely, when γ is set to a smaller value, the emphasis shifts towards prioritizing the reduction of total latency over concerns about overloading the MEC server. Using the intertwined cost function defined in (15), we can formulate the multi-objective optimization problem P_1 as follows:

$$\begin{aligned} P_1: \quad & \min_{\beta_k} C = T - \gamma \sum_{k=1}^{K} \beta_k \\ & = [\max_{k\in\mathcal{K}}[\frac{(1-\alpha_k)(1-\beta_k)D_k}{B\log_2(1+p_k g_k)}] \\ & + r(\epsilon)\max_{k\in\mathcal{K}}[(\tau G(\alpha_k \\ & + (1-\alpha_k)\beta_k)\frac{D_k}{e_k}, \tau G \sum_{k=1}^{K}(1-\alpha_k)(1-\beta_k)\frac{D_k}{e_s}] \\ & + r(\epsilon)t_w] - \gamma \sum_{k=1}^{K} \beta_k, \qquad (16a) \\ \text{s.t.} \quad & C_1 \ : \beta_K \in \{0,1\}, \quad \forall k \in \mathcal{K}. \qquad (16b) \end{aligned}$$

The P_1 is classified as a non-convex mixed-integer nonlinear programming (MINLP) problem since it combines the complexity of optimizing integer variables (specifically, β_k in our case) with the challenge of dealing with nonlinear functions (specifically, including the non-differentiable max(.) function in the objective function in our case). Typically, for such MINLP problems, relaxation techniques are employed to find solutions that are near-optimal. Therefore, in the subsequent subsections, we will apply various relaxation techniques and heuristics to efficiently address and solve this MINLP problem.

Algorithm 1 Proposed HCFLX scheme

```
Input: $k$, $\beta_k$, $\tilde{\beta}_k$, $p_k$, $B$, $D_k$, $t_1$, $t_2$, $T$, $\theta_c$, $e_k$, $e_s$, $g_k$
Dividing problem $P_2$ to sub-problems $P^{(sub)}_{2-1}$, $P^{(sub)}_{2-2}$, and $P^{(sub)}_{2-3}$
Initialize: $t_1$, $t_2$, and $\tilde{\beta}_k$ are normally initialized within the constraints in $P^{(sub)}_{2-1}$
Output: $\tilde{\beta}_k$, $t_1$, $t_2$
while True do
    updating $t_1 \leftarrow \max_{k\in\mathcal{K}}[\frac{(1-\alpha_k)(1-\tilde{\beta}_k)D_k}{B\log_2(1+p_k g_k)}]$ via lower bound from $P^{(sub)}_{2-2}$
    updating $t_2 \leftarrow \max_{k\in\mathcal{K}}[(\tau G(\alpha_k + (1-\alpha_k)\tilde{\beta}_k)\frac{D_k}{e_k}, \tau G \sum_{k=1}^{K}(1-\alpha_k)(1-\tilde{\beta}_k)\frac{D_k}{e_s}]$ through the lower bound of $P^{(sub)}_{2-3}$
    updating $\tilde{\beta}_k \leftarrow$ sub-problem $P^{(sub)}_{2-1}$ via Simplex Algorithm
    $C_i \leftarrow C(\tilde{\beta}_k, t_1, t_2)$
    if $|C_i - C_{i-1}| < \theta_c$ then
        break
    end if
    i = i + 1
end while
return $\tilde{\beta}_k$, $t_1$, $t_2$
```

B. PROPOSED SCHEME

To make the problem P_1 into the convex problem, the binary variable of β_k is relaxed into a continuous value, denoted as $\tilde{\beta}_k$. Moreover, this problem P_1 still has the form max(.) in the objective function, which is not differentiable. Hence, we convert max(.) into an affine function by introducing an additional variables t_1 and t_2, and letting

$$t_1 = \max_{k\in\mathcal{K}}[\frac{(1-\alpha_k)(1-\beta_k)D_k}{B\log_2(1+p_k g_k)}], \qquad (17a)$$

$$t_2 = \max_{k\in\mathcal{K}}[\tau G(\alpha_k + (1-\alpha_k)\beta_k)\frac{D_k}{e_k}, \tau G \sum_{k=1}^{K}(1-\alpha_k)(1-\beta_k)\frac{D_k}{e_s}]. \qquad (17b)$$

The new variables t_1 and t_2 include some additional constraints

$$\frac{(1-\alpha_k)(1-\tilde{\beta}_k)D_k}{B\log_2(1+p_k g_k)} \leq t_1, \qquad (18a)$$

$$\tau G(\alpha_k + (1-\alpha_k)\tilde{\beta}_k)\frac{D_k}{e_k} \leq t_2, \qquad (18b)$$

$$\tau G \sum_{k=1}^{K}(1-\alpha_k)(1-\tilde{\beta}_k)\frac{D_k}{e_s} \leq t_2. \qquad (18c)$$

Using the additional constraints with the new variables t_1 and t_2, we can rewrite the problem P_2 as given by

$$P_2: \quad \min_{\tilde{\beta}_k, t_1, t_2} \; C = [t_1 + r(\epsilon)t_2 + r(\epsilon)t_w] - \gamma \sum_{k=1}^{K} \tilde{\beta}_k, \tag{19a}$$

$$\text{s.t.} \quad C_1 : 0 \leq \tilde{\beta}_k \leq 1, \quad \forall k \in \mathcal{K}, \tag{19b}$$

$$C_2 : \frac{(1-\alpha_k)(1-\tilde{\beta}_k)D_k}{B\log_2(1+p_k g_k)} \leq t_1, \tag{19c}$$

$$C_3 : \tau G(\alpha_k + (1-\alpha_k)\tilde{\beta}_k)\frac{D_k}{e_k} \leq t_2, \tag{19d}$$

$$C_4 : \tau G \sum_{k=1}^{K} (1-\alpha_k)(1-\tilde{\beta}_k)\frac{D_k}{e_s} \leq t_2. \tag{19e}$$

Then, the relaxed version of problem P_2 can be solved by decomposing it into three sub-problems: $P_{2-1}^{(sub)}$, $P_{2-2}^{(sub)}$, and $P_{2-3}^{(sub)}$, with each sub-problem associated with specific optimization variables—$\tilde{\beta}_k$, t_1, and t_2, respectively. Then, the sub-problem $P_{2-1}^{(sub)}$ is given by

$$P_{2-1}^{(sub)}: \quad \min_{\tilde{\beta}_k} -\gamma \sum_{k=1}^{K} \tilde{\beta}_k, \tag{20a}$$

$$\text{s.t.} \quad C_1 : 0 \leq \tilde{\beta}_k \leq 1, \quad \forall k \in \mathcal{K}, \tag{20b}$$

$$C_2 : \frac{(1-\alpha_k)(1-\tilde{\beta}_k)D_k}{B\log_2(1+p_k g_k)} \leq t_1, \tag{20c}$$

$$C_3 : (\tau G(\alpha_k + (1-\alpha_k)\tilde{\beta}_k)\frac{D_k}{e_k} \leq t_2, \tag{20d}$$

$$C_4 : \tau G \sum_{k=1}^{K} (1-\alpha_k)(1-\tilde{\beta}_k)\frac{D_k}{e_s} \leq t_2. \tag{20e}$$

Lemma 1: $P_{2-1}^{(sub)}$ is a linear program (LP) with respect to optimization variables ($\tilde{\beta}_k$).

First, the objective function is linear with respect to $\tilde{\beta}_k$. In addition, the inequality constraints are affine with respect to the optimization variables ($\tilde{\beta}_k$). Correspondingly, since the objective and inequality constraint functions are affine, the problem a LP with respect to the optimization variables ($\tilde{\beta}_k$).

Then, sub-problem $P_{2-2}^{(sub)}$ and sub-problem $P_{2-3}^{(sub)}$ are defined as

$$P_{2-2}^{(sub)}: \quad \min_{t_2} r(\epsilon)t_2, \tag{21a}$$

$$\text{s.t.} \quad C_2 : \frac{(1-\alpha_k)(1-\tilde{\beta}_k)D_k}{B\log_2(1+p_k g_k)} \leq t_1, \quad \forall k \in \mathcal{K}, \tag{21b}$$

$$C_4 : \tau G \sum_{k=1}^{K} (1-\alpha_k)(1-\tilde{\beta}_k)\frac{D_k}{e_s} \leq t_2. \tag{21c}$$

and,

$$P_{2-3}^{(sub)}: \quad \min_{t_2} r(\epsilon)t_2, \tag{22a}$$

$$\text{s.t.} \quad C_3 : \tau G(\alpha_k + (1-\alpha_k)\tilde{\beta}_k)\frac{D_k}{e_k} \leq t_2, \tag{22b}$$

Lemma 2: The subproblems of $P_{2-2}^{(sub)}$ and $P_{2-3}^{(sub)}$ are a strictly increasing function with respect to optimization variables t_1 and t_2, respectively. Thus, $t_1^* = \max_{k\in\mathcal{K}}[\frac{(1-\alpha_k)(1-\tilde{\beta}_k^*)D_k}{B\log_2(1+p_k g_k)}]$ and $t_2^* = \max_{k\in\mathcal{K}}[(\tau G(\alpha_k + (1-\alpha_k)\tilde{\beta}_k^*)\frac{D_k}{e_k}, \tau G\sum_{k=1}^{K}(1-\alpha_k)(1-\tilde{\beta}_k^*)\frac{D_k}{e_s}]$.

As the sub-problems of $P_{2-2}^{(sub)}$ and $P_{2-3}^{(sub)}$ have the form of t_1 and t_2 in the objective function, respectively, it is clear that the objection functions of $P_{2-2}^{(sub)}$ and $P_{2-3}^{(sub)}$ are strictly increasing functions with respect to t_1 and t_2, respectively. Correspondingly, the optimal point of t_1 and t_2 (t^*) are at the lower bound of constraints. Thus, we can conclude that $t_1^* = \max_{k\in\mathcal{K}}[\frac{(1-\alpha_k)(1-\tilde{\beta}_k^*)D_k}{B\log_2(1+p_k g_k)}]$ and $t_2^* = \max_{k\in\mathcal{K}}[\tau G(\alpha_k + (1-\alpha_k)\tilde{\beta}_k^*)\frac{D_k}{e_k}, \tau G\sum_{k=1}^{K}(1-\alpha_k)(1-\tilde{\beta}_k^*)\frac{D_k}{e_s}]$.

Lemmas 1 and 2 lay the foundation for solving P_2 through the block coordinate descent method (BCD) [42]. Specifically, given fixed values of t_1 and t_2, we can efficiently determine the optimal values of $\tilde{\beta}_k$ using the Simplex algorithm (SA). This process is executed based on the block coordinate descent method. The algorithmic steps are summarized in **Algorithm 1** is given by

- **Initialization:** The algorithm begins by initializing β_k, t_1, and t_2 normally within the constraints in the Lemma 1.
- **Updating Lower Bounds Loop:** Within the client loop, in each iteration, the algorithm updating the lower bound for t_1, t_2, and $\tilde{\beta}_k$ using the Simplex algorithm (SA) according to Lemma 2 in lines 1-10.
- **Acceptance or Rejection:** Subsequently, t_1 and t_2 are iteratively refined, with each one being fixed while the other varies until the cost function converges (i.e., the difference between the current and previous cost values falls below the convergence threshold θ_c).
- **Return Optimal Values:** After iterating over all clients, the algorithm returns the values of $\tilde{\beta}_k$, t_1, and t_2 in line 11 to use in Algorithm 2.

Once we obtain the optimal value $\tilde{\beta}_k{}^*$, which is a relaxed variable of β_k, from Algorithm 1, it needs to be converted into binary values. The simplest approach for conversion involves using a threshold value, typically 0.5. However, in this paper, instead of relying on such a straightforward mechanism, which can sometimes yield suboptimal results, we employ the Mutas & Simulated Annealing heuristic algorithm for continuous-to-discrete mapping, as outlined in **Algorithm 2**. This approach aims to achieve a near-optimal conversion of β_k, as demonstrated in [43]. The details of Mutas & Simulated Annealing heuristic algorithm in **Algorithm 2** are as follows:

Algorithm 2 Proposed HCFLX Scheme With Mutas & Simulated Annealing Heuristic Akgorithm

```
Input: $k$, $\beta_k$, $\tilde{\beta_k}$, $p_k$, $B$, $D_k$, $t_1$, $t_2$, $T$, $\theta_c$, $e_k$, $e_s$, $g_k$
Dividing problem $P_2$ to sub-problems $P_{2-1}^{(sub)}$, $P_{2-2}^{(sub)}$, and $P_{2-3}^{(sub)}$
Initialize: $t_1$, $t_2$, and $\tilde{\beta_k}$ are normally initialized within the constraints in $P_{2-1}^{(sub)}$
Output: Optimal $\beta_k^*$, $T^*$
1: $\beta_k^{old}, t_1^{old}, t_2^{old} \leftarrow Mutas(\tilde{\beta}_k^{old})$
2: $f \leftarrow D(t_1^{old}, t_2^{old})$ & $T \leftarrow T_{max}$
3: for $k \leftarrow 1$ to $K$ do
4:     while $T > T_{min}$ do
5:         randomly generates a new assignment $\tilde{\beta}_k^{new}$ in the
               neighbourhood of $\tilde{\beta}_k^{old}$
6:         $\beta_k^{new}, t_1^{new}, t_2^{new} \leftarrow Mutas(\tilde{\beta}_k^{new})$
7:         $f' \leftarrow D(t_1^{old}, t_2^{old})$ & $\Delta d = f - f'$
8:         if $\Delta d \leq 0$ then
9:             $\beta_k^{old} \leftarrow \beta_k^{new}$ & $f \leftarrow f'$
10:        else if $Rand(0, 1) < \exp(-\frac{\Delta d}{T})$ then
11:            $\beta_k^{old} \leftarrow \beta_k^{new}$ & $f \leftarrow f'$
12:        end if
13:    end while
14: end for
15: $\beta_k^* \leftarrow \tilde{\beta_k}$ return $\beta_k^*$, $T^*$
```

- **Initialization:** The algorithm begins by initializing β_k, t_1, and t_2 considered as β_k^{old}, t_1^{old}, and t_2^{old} using the **Mutas** function applied to $\tilde{\beta}_k^{old}$. It also calculates the function value f based on the current values of t_1^{old} and t_2^{old} and sets an initial temperature T to its maximum value (T_{max}) in lines 1-2.
- **Loop Over Clients** (k)**:** The algorithm iterates over all clients, denoted by the variable k in lines 3-14.
- **Simulated Annealing Loop:** Within the client loop, there is a Simulated Annealing loop. This loop continues until the T reaches a minimum threshold T_{min} in lines 4-13.
- **Neighborhood Exploration:** In each iteration of the Simulated Annealing loop, a new assignment $\tilde{\beta}_k^{new}$ is randomly generated in the neighborhood of the current $\tilde{\beta}_k^{old}$ in line 5.
- **Mutation (Mutas):** The algorithm applies the Mutas function to $\tilde{\beta}_k^{new}$, resulting in updated values for β_k^{new}, t_1^{new}, and t_2^{new} in line 6.
- **Objective Function Evaluation:** It calculates f' based on the current values of t_1^{old} and t_2^{old} and computes Δd as the difference between the previous function value f and the new function value f' in lines 7-9.
- **Acceptance or Rejection:** If $\Delta d \leq 0$, the new assignment β_k^{new} is accepted, and f is updated to f'. Otherwise, there is a probability-based acceptance of the new assignment, based on the temperature T and the difference Δd in lines 10-12.
- **Client Loop End:** Once the temperature T reaches the minimum threshold, the Simulated Annealing loop for client k ends in line 14.
- **Return Optimal Values:** After iterating over all clients, the algorithm returns the optimal values of β_k^* and T^* in line 15.

C. THEORETICAL ANALYSIS OF PRIVACY PROTECTION USING DIFFERENTIAL PRIVACY (DP) IN PROPOSED HFCLX

To ensure privacy in our proposed HFCLX, we leverage DP as described in [44], a robust framework that introduces random noise to the data to obscure individual entries. Here, we detail the theoretical analysis of how DP can be applied to CL within a HFCL environment. In the proposed HFCLX, clients are categorized based on the sensitivity of their data:

- **FL Clients**: Clients with sensitive data that should not be exposed. These clients participate in FL, where only model updates (not raw data) are shared.
- **CL Clients**: Clients with less sensitive data. These clients send raw data to a centralized server for training.

Even though CL clients have less sensitive data that can be expected, additional measures can be taken to further protect their privacy. That is, we can also apply DP to the data before it is sent to the centralized server. To this aim, we add noise to less sensitive raw data as follows:

- **Data Perturbation**: Each client's raw data is perturbed by adding Gaussian noise. This noise is calibrated to ensure a balance between privacy protection and model accuracy.
- **Mathematical Representation**: For a given dataset $\mathcal{D}_k$, each data point x_i is transformed to $x_i + N(0, \sigma^2)$, where $N(0, \sigma^2)$ represents Gaussian noise with mean 0 and variance σ^2.

Furthermore, we assume the privacy and noise analysis parameters. According to [45], the standard deviation σ of the Gaussian noise is given $\sigma = \frac{\Delta f \sqrt{2 \log(1.25/\delta)}}{\epsilon}$, and then, we rearrange the formula to solve for ϵ is given $\epsilon = \frac{\Delta f \sqrt{2 \log(1.25/\delta)}}{\sigma}$, where ϵ represents a measure of privacy loss. Smaller values of ϵ indicate stronger privacy ($1/\epsilon$). δ is the probability of the privacy guarantee being violated. Smaller values of δ indicate stronger privacy and Δf is the sensitivity of the function.

V. PERFORMANCE EVALUATION

This section presents the numerical and simulation results that validate the effectiveness of the proposed HCFLX scheme under various parameter settings.

A. EXPERIMENTAL SETUP

In order to analyze the performance of the proposed HFCLX scheme, various benchmarks are analyzed as follows. It should be noted that the key distinction between these benchmarks and our proposed scheme lies in their treatment of clients with less sensitive data, as we made the assumption that clients with sensitive data should actively participate in the FL process.

- Benchmark 1: This benchmark employs the CL approach for all clients with less sensitive data.

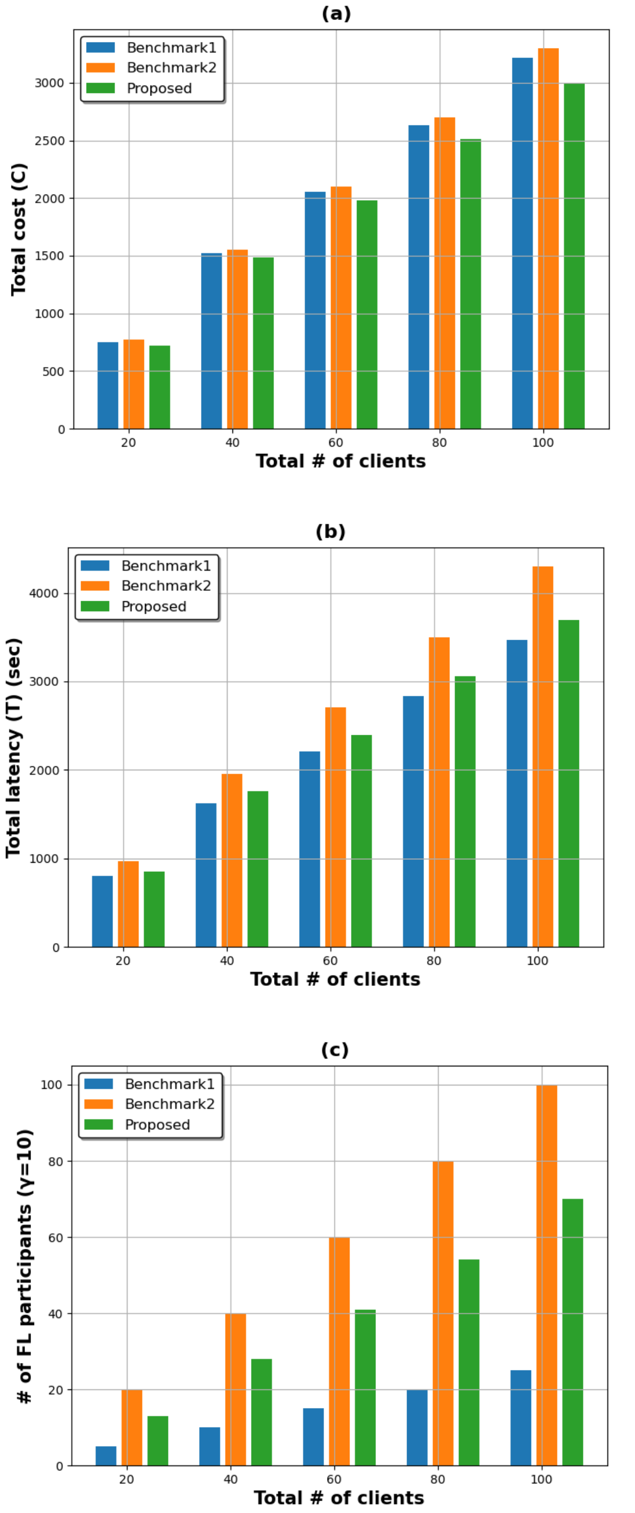

FIGURE 3. (a) Total cost (C) with respect to the sum of clients with less sensitive data and sensitive data. (b) Total latency (T) with respect to the sum of clients with less sensitive data and sensitive data. (c) The number of FL participants with respect to the sum of clients with less sensitive data and sensitive data for 20 clients.

- Benchmark 2: In contrast, Benchmark 2 adopts the FL approach for all clients with less sensitive data according to the FedAvg algorithm [5].

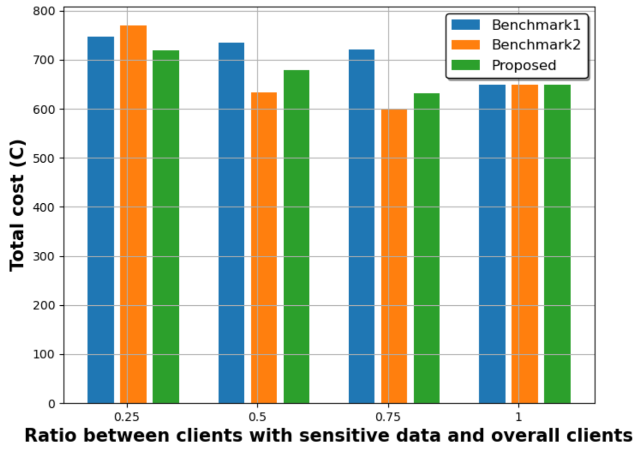

FIGURE 4. Total cost (C) with respect to the ratio between clients with sensitive data and overall clients.

TABLE 3. Complete set of information about the model architectures and hyperparameters used in our experiments.

Dataset	MNIST	Fashion-MNIST	CIFAR-10
Model	CNN	CNN	CNN
Network dense	200	200	200
Activation	Relu	Relu	Relu
Last layer Activation	Softmax	Softmax	Softmax
Optimizer	Stochastic Gradient Descent (SGD)	SGD	SGD
Clients [$\mathcal{K}$]	100	100	100
Classes	10	10	10
Learning rate	0.01	0.01	0.01
Batch size [H_k]	10	10	10
Build shape	784	784	1024
Local updates/rounds	500	500	500

- Proposed HFCLX scheme: Our HFCLX scheme focuses on optimizing client selection for FL. It aims to strike a balance between reducing total latency and managing the computational load on the MEC server.

B. EVALUATION OF PROPOSED COST FUNCTION

We consider a network environment comprising a total of K number of clients, where some of the clients possess sensitive data while others retain the less sensitive data. Specifically, clients obtaining sensitive data adhere to the FL approach, while for clients with less sensitive data, we introduce and employ our proposed HFCLX scheme. This dual approach results in the formation of two distinct client groups, each adhering to either the FL or CL approach to aim to make a balance between reducing total latency and managing the computational load on the MEC server. Table 3 provides the complete set of information about the model architectures and hyperparameters used in our experiments.

In Fig. 3 (a), we illustrate the total cost (C) with respect to the number of clients, including those with less sensitive and sensitive data. It is evident that the total cost (C) increases as the number of clients increases. This increase

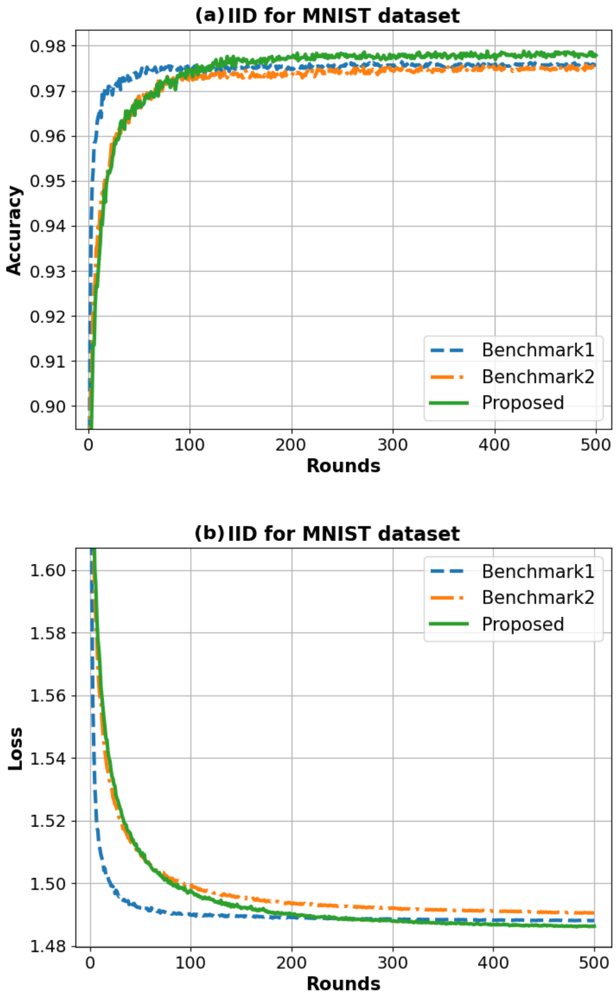

FIGURE 5. IID data on MNIST dataset for 100 clients: (a) Accuracy (b) Loss.

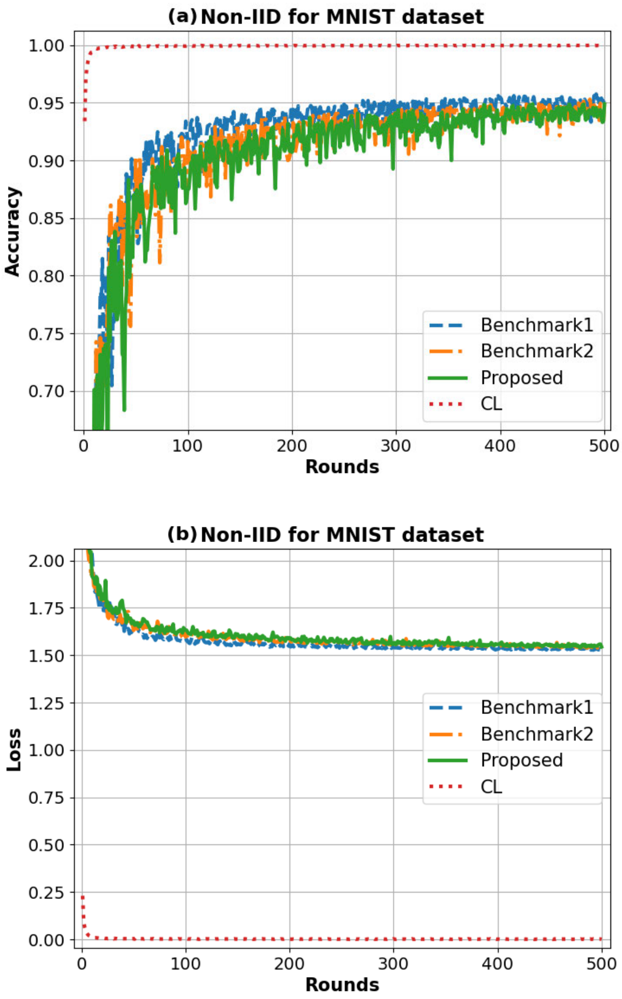

FIGURE 6. Non-IID data on MNIST dataset for 100 clients: (a) Accuracy (b) Loss.

is a direct consequence of the findings depicted in Fig. 3 (b), which show that the total latency (T) also increases as the number of clients increases. In contrast, Fig. 3 (c) demonstrates that the number of FL clients also increases with the increasing number of clients, thereby reducing the burden on the MEC server. Specifically, in Benchmark 1, only sensitive clients participate in the FL approach, while Benchmark 2 incorporates all clients, regardless of data sensitivity, in the FL approach. Consequently, Fig. 3 (a) represents a weighted sum of the total latency (T) and the negative count of FL clients. In our scenario with γ set to 10, this results in an overall increasing trend in the total cost (C). Notably, our proposed scheme outperforms all benchmarks by striking a balance between total latency (T) and the computational burden on the MEC server, thereby achieving the minimum cost (C) across a wide range of total client counts.

Furthermore, in Fig. 4, we present an analysis of the total cost (C) in relation to the proportion of clients with sensitive data in comparison to the total number of clients. As illustrated in Fig. 4, the proposed scheme demonstrates superior performance when striking a balance between two key factors: i) the overall latency, encompassing both computation and communication, and ii) the training load on the MEC server. As the ratio equals to 1, all clients should participate in FL process showing that the total cost becomes same across all benchmarks and the proposed scheme.

C. EVALUATION OF ACCURACY

In this subsection, the evaluation of accuracy is conducted, encompassing the relationship between communication rounds and the accuracy of the proposed HFCLX scheme. The proposed HFCLX scheme was implemented based on a part EAFL scheme [10], which is the latest implementation adopted for the edge computing field. The main difference between these two works is that the offloading method was used in EAFL, while the HFCL approach is used in the proposed HFCLX scheme. To evaluate the performance of the proposed HFCLX scheme, we conducted experiments on three well-known datasets: MNIST, Fashion MNIST, and

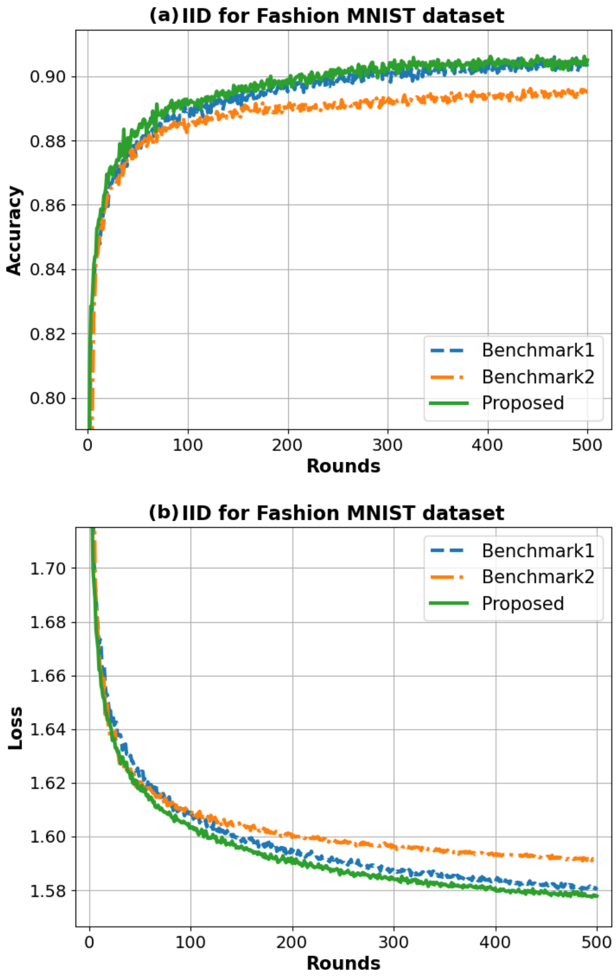

FIGURE 7. IID data on Fashion MNIST dataset for 100 clients: (a) Accuracy (b) Loss.

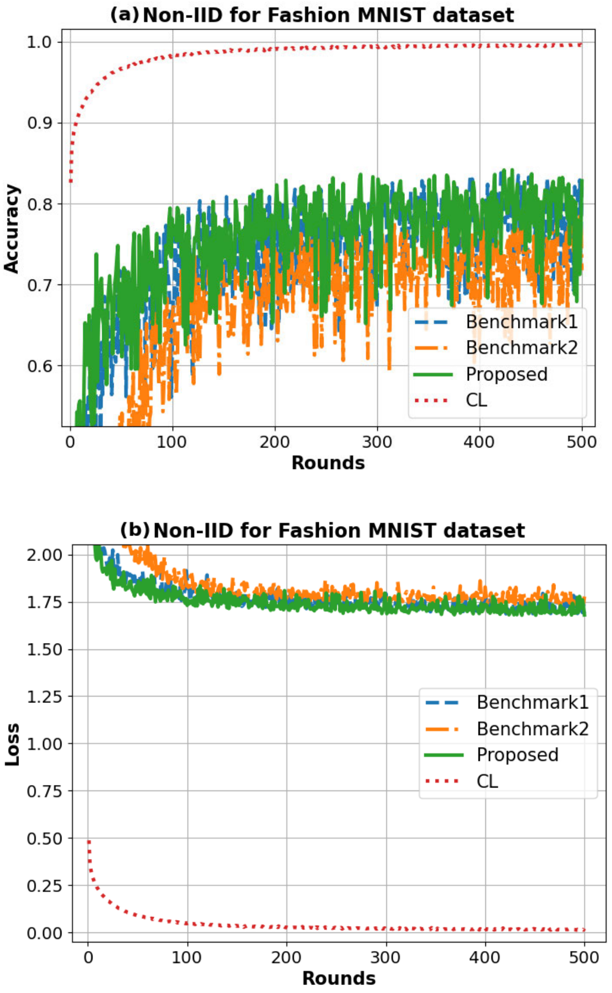

FIGURE 8. Non-IID data on Fashion MNIST dataset for 100 clients: (a) Accuracy (b) Loss.

CIFAR-10, with a focus on both Independent and IID and non-IID data settings, which are widely used in FL and ML evaluations [46], [47], [48]. It is noticeable that to make the non-IID data setting in each mentioned dataset, we consider class imbalance by assigning different proportions of classes to different clients.

In Fig. 5 (a) the accuracy curve, demonstrates an impressive accuracy rate of nearly 98 percent for IID data. This signifies the effectiveness of the proposed HFCLX scheme. In Fig. 5 (b), the loss curve illustrates the convergence of the model during training, showing that the proposed scheme is 0.08 percent lower than the Benchmarks. Furthermore, Fig. 6 (a) shows the accuracy reaches nearly 95 percent for the non-IID data setting, and in Fig. 6 (b) displays the convergence of the loss, that the proposed scheme is 0.96 and near to 0.005 percent lower than the Benchmark 1 and Benchmark 2, respectively. Here, we present the CL graph as the upper bound, illustrating the disparity between the CL and the proposed scheme, as well as other benchmarks, within the non-IID setting. Thus, from this empirical study, for both IID and non-IID settings in the MNIST dataset, the test accuracy gap between the proposed scheme and benchmarks is negligible while it achieving the balance between the total latency and computational load on the MEC server.

Fig. 7 (a) shows the accuracy reaches approximately 91 percent for the IID data setting and in Fig. 5 (b) the convergence of the loss, that the proposed scheme is 0.14 and 0.87 percent lower than Benchmark 1 and Benchmark 2, respectively. In Fig. 8 (a), the accuracy is approximately 84 percent for the non-IID data setting, and in Fig. 8 (b) the convergence of the loss, that the proposed scheme is 0.29 and 0.57 percent lower than Benchmark 1 and Benchmark 2, respectively. Furthermore, Fig. 9 (a) shows the accuracy reaches approximately 85 percent for the IID data setting, and in Fig. 9 (b) the convergence of the loss, that the proposed scheme is 0.07 and 0.14 percent lower than Benchmark 1 and Benchmark 2, respectively. Fig. 10 (a) shows the accuracy is approximately 62 percent for the non-IID data setting, and Fig. 10 (b) shows the convergence of the loss, that the proposed scheme is 2.04 and 2.16 percent lower than Benchmark 1 and Benchmark 2, respectively. Moreover, in Fig. 10. the lower performance of the CIFAR-10 dataset

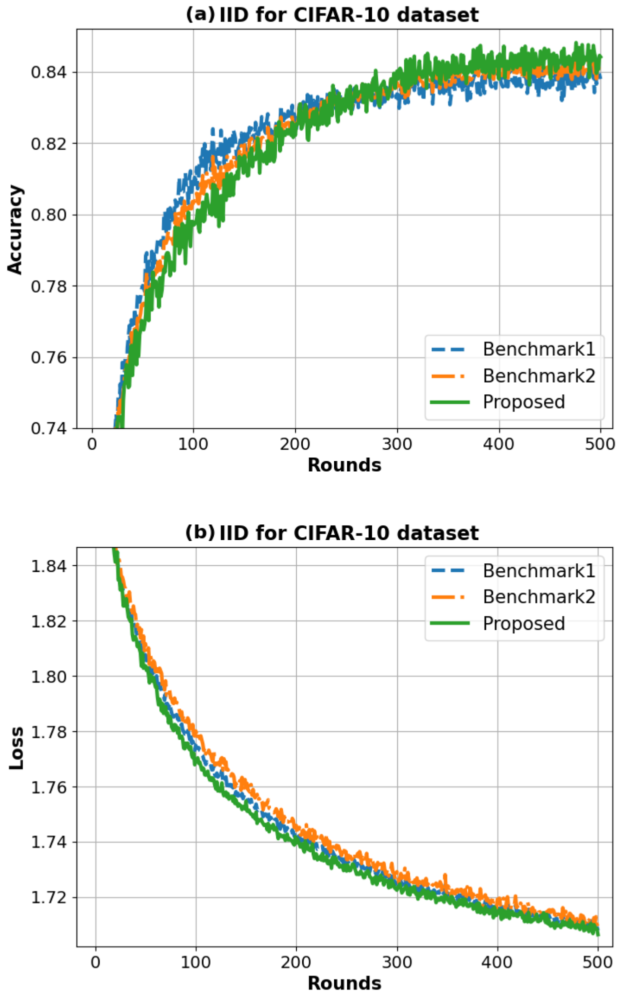

FIGURE 9. IID data on CIFAR-10 dataset for 100 clients: (a) Accuracy (b) Loss.

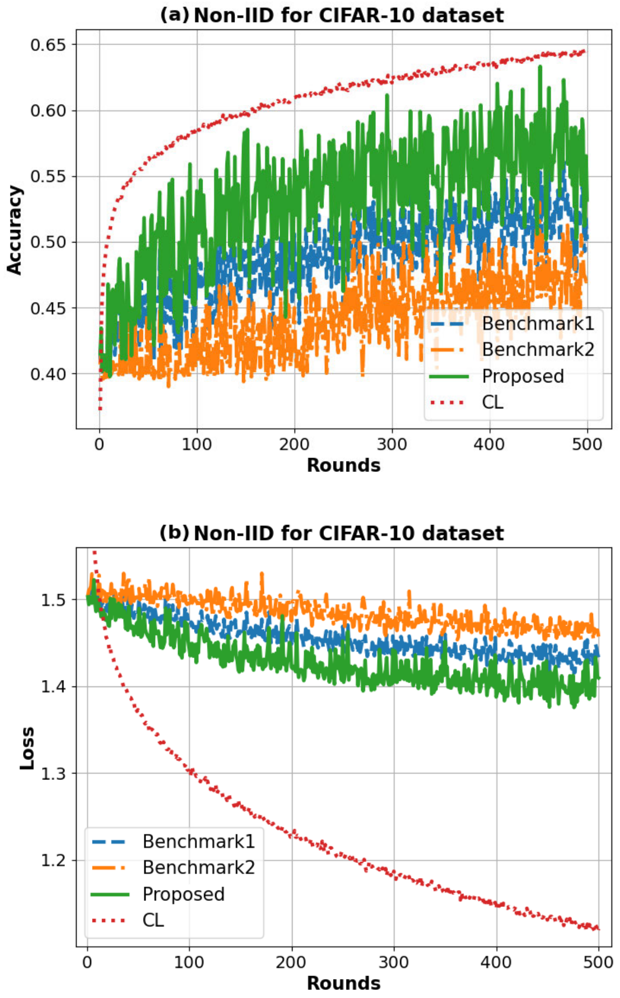

FIGURE 10. Non-IID data on CIFAR-10 dataset for 100 clients: (a) Accuracy (b) Loss.

compared to MNIST and Fashion MNIST can be attributed to several factors:

- Dataset Heterogeneity: CIFAR-10 is more complex, with color images and ten distinct classes, while MNIST and Fashion MNIST have grayscale images and fewer classes. The increased complexity of CIFAR-10 can pose challenges for the HFCLX scheme.
- Image Content: CIFAR-10 contains diverse objects, backgrounds, and orientations, making it harder for the model to generalize effectively, compared to the more uniform content in MNIST and Fashion MNIST.
- Non-IID Data: If CIFAR-10 is used in a non-IID setting, where data distribution among clients is uneven, biased training can hinder convergence and affect performance.
- Model Complexity: The effectiveness of the HFCLX scheme depends on its complexity and architecture. It might struggle to handle the complexity of the CIFAR-10 dataset optimally.

Fig. 11 (a) and (b) demonstrate applying the DP method on the clients who participate in CL scenario to protect privacy in the proposed HFCLX scheme and the impact of DP on the accuracy of datasets, respectively. We assume $\delta = 10^{-5}$, σ ranging from 0.1 to 0.4, and $\Delta f = 1$. In Fig. 11 (a), the 4 noise levels of σ from 0.1 to 0.4 show that by increasing the amount of σ while keeping δ fixed, the Privacy protection level $(1/\epsilon)$ increases. Furthermore, Fig. 11 (b) represents that by adding DP, the accuracy of different datasets is lower compared to the case without DP. Therefore, careful DP management should be needed to balance the tradeoff between accuracy and privacy, which is beyond scope of this paper.

D. DISCUSSION

While our HFCLX scheme demonstrates promising results, particularly with the MNIST, Fashion MNIST, and CIFAR-10 datasets, the complexity of more intricate datasets like CIFAR-100 presents a scalability challenge. CIFAR-100, with its larger number of classes and increased data complexity, demands more robust computational resources and optimized algorithms. To handle the increased complexity of datasets such as CIFAR-100, we propose several solutions:

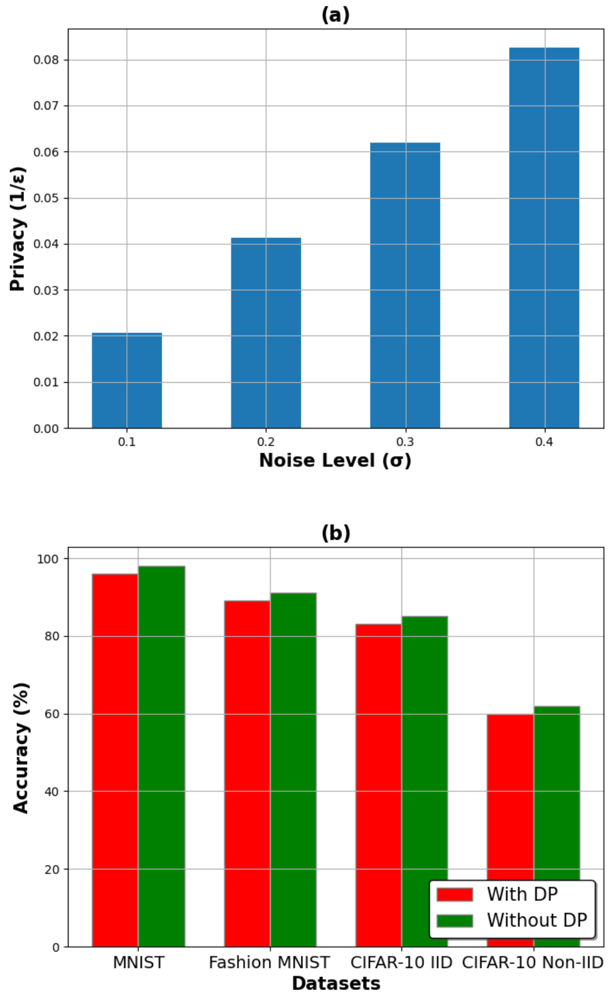

FIGURE 11. Relationship between noise level (σ) and privacy ($1/\epsilon$); (b) Comparison of model accuracy with and without Differential Privacy (DP) across different datasets.

joint optimization of computing and scaling, dynamic resource allocation, distributed computing, and the use of deeper neural network architectures tailored for complex datasets. Additionally, we suggest hybrid training approaches that combine the benefits of federated and centralized learning, allowing for scalable and privacy-preserving training.

Future work will focus on enhancing scalability and addressing current limitations. This includes implementing adaptive learning rates, developing cluster-based client selection methods, and investigating the application of differential privacy directly to raw data in the centralized learning process to further alleviate privacy concerns. Enhancing privacy-aware federated learning techniques and exploring reinforcement learning for client selection and meta-learning approaches will also be prioritized. Applying the HFCLX scheme to real-world scenarios in IoT, edge computing, healthcare, and finance will demonstrate its effectiveness in diverse and practical applications. These improvements will ensure that our approach remains effective and relevant in handling complex datasets and real-world challenges, ultimately advancing the field of HFCL.

VI. CONCLUSION

In this paper, we have proposed the HFCLX scheme by introducing a multi-objective optimization problem to seek to strike a balance between i) total latency, including computation and communication, and ii) the training burden on the MEC server, achieved by adjusting the set of participants in FL, taking into account client selection under different privacy levels. Despite the mixed-integer nonlinear programming problem, we have employed relaxation techniques in combination with the Mutas & Simulated Annealing Heuristic algorithm to develop a near-optimal yet practical algorithm. Numerical and simulation results have been provided to validate the efficiency and accuracy of our proposed scheme and demonstrate the advantages of the HFCLX scheme. The results demonstrate that our proposed HFCLX scheme can mitigate the total cost (C) by an average of 7 percent and nearly 5 percent compared to Benchmarks 1 and 2, respectively. Moreover, experimental results on three well-known real-world datasets demonstrate that the proposed HFCLX scheme maintains an acceptable level of accuracy and loss. This is particularly evident in the MNIST dataset, Fashion-MNIST dataset, and CIFAR-10 dataset for 500 rounds of training for IID and non-IId data, respectively. As part of future work, we can explore enhanced algorithms, such as Reinforcement Learning (RL), to improve energy efficiency in our system as online learning methods.

REFERENCES

[1] D. Wu, R. Ullah, P. Harvey, P. Kilpatrick, I. Spence, and B. Varghese, "FedAdapt: Adaptive offloading for IoT devices in federated learning," *IEEE Internet Things J.*, vol. 9, no. 21, pp. 20889–20901, Nov. 2022.

[2] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2021.

[3] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3072–3108, 4th Quart., 2019.

[4] Y. Liu, S. Bi, Z. Shi, and L. Hanzo, "When machine learning meets big data: A wireless communication perspective," *IEEE Veh. Technol. Mag.*, vol. 15, no. 1, pp. 63–72, Mar. 2020.

[5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, Apr. 2017, pp. 1273–1282.

[6] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecný, S. Mazzocchi, B. McMahan, and J. Roselander, "Towards federated learning at scale: System design," in *Proc. Int. Conf. Mach. Learn. Syst.*, vol. 1, 2019, pp. 374–388.

[7] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, "Toward an intelligent edge: Wireless communication meets machine learning," *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 19–25, Jan. 2020.

[8] C. Feng, Z. Zhao, Y. Wang, T. Q. S. Quek, and M. Peng, "On the design of federated learning in the mobile edge computing systems," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 5902–5916, Sep. 2021.

[9] G. Zhu, Y. Wang, and K. Huang, "Low-latency broadband analog aggregation for federated edge learning," 2018, *arXiv:1812.11494*.

[10] Z. Ji, L. Chen, N. Zhao, Y. Chen, G. Wei, and F. R. Yu, "Computation offloading for edge-assisted federated learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9330–9344, Sep. 2021.

[11] F. Solat, T. Y. Kim, and J. Lee, ''A novel group management scheme of clustered federated learning for mobile traffic prediction in mobile edge computing systems,'' *J. Commun. Netw.*, vol. 25, no. 4, pp. 480–490, Aug. 2023.

[12] R. Saha, S. Misra, A. Chakraborty, C. Chatterjee, and P. K. Deb, ''Data-centric client selection for federated learning over distributed edge networks,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 2, pp. 675–686, Feb. 2023.

[13] P. Tian, W. Liao, W. Yu, and E. Blasch, ''WSCC: A weight-similarity-based client clustering approach for non-IID federated learning,'' *IEEE Internet Things J.*, vol. 9, no. 20, pp. 20243–20256, Oct. 2022.

[14] J. Wu, S. Drew, and J. Zhou, ''FedLE: Federated learning client selection with lifespan extension for edge IoT networks,'' 2023, *arXiv:2302.07305*.

[15] M. Jiang and R. Tang, ''Study on hyperparameter adaptive federated learning,'' in *Proc. IEEE 3rd Int. Conf. Electron. Technol., Commun. Inf. (ICETCI)*, May 2023, pp. 712–717.

[16] A. M. Elbir, S. Coleri, A. K. Papazafeiropoulos, P. Kourtessis, and S. Chatzinotas, ''A hybrid architecture for federated and centralized learning,'' *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 3, pp. 1529–1542, Sep. 2022.

[17] N. Huang, M. Dai, Y. Wu, T. Q. S. Quek, and X. Shen, ''Wireless federated learning with hybrid local and centralized training: A latency minimization design,'' *IEEE J. Sel. Topics Signal Process.*, vol. 17, no. 1, pp. 248–263, Jan. 2023.

[18] Q. Yang, Y. Liu, T. Chen, and Y. Tong, ''Federated machine learning: Concept and applications,'' *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, Jan. 2019.

[19] B. Liu, Y. Guo, and X. Chen, ''PFA: Privacy-preserving federated adaptation for effective model personalization,'' in *Proc. Web Conf.*, Apr. 2021, pp. 923–934.

[20] Y. He, M. Yang, Z. He, and M. Guizani, ''Resource allocation based on digital twin-enabled federated learning framework in heterogeneous cellular network,'' *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 1149–1158, Jan. 2023.

[21] Z. Tianqing, W. Zhou, D. Ye, Z. Cheng, and J. Li, ''Resource allocation in IoT edge computing via concurrent federated reinforcement learning,'' *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1414–1426, Jan. 2022.

[22] Y. Zhan, P. Li, and S. Guo, ''Experience-driven computational resource allocation of federated learning by deep reinforcement learning,'' in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, May 2020, pp. 234–243.

[23] W. Y. B. Lim, J. S. Ng, Z. Xiong, J. Jin, Y. Zhang, D. Niyato, C. Leung, and C. Miao, ''Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 3, pp. 536–550, Mar. 2022.

[24] M. M. Wadu, S. Samarakoon, and M. Bennis, ''Federated learning under channel uncertainty: Joint client scheduling and resource allocation,'' in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, May 2020, pp. 1–6.

[25] Z. Lin, H. Liu, and Y. J. A. Zhang, ''CFLIT: Coexisting federated learning and information transfer,'' *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 8436–8453, Apr. 2023.

[26] J. Zhang, S. Chen, X. Zhou, X. Wang, and Y.-B. Lin, ''Joint scheduling of participants, local iterations, and radio resources for fair federated learning over mobile edge networks,'' *IEEE Trans. Mobile Comput.*, vol. 22, no. 7, pp. 3985–3999, Feb. 2022.

[27] H. Ko, J. Lee, S. Seo, S. Pack, and V. C. M. Leung, ''Joint client selection and bandwidth allocation algorithm for federated learning,'' *IEEE Trans. Mobile Comput.*, vol. 22, no. 6, pp. 3380–3390, Jun. 2023.

[28] Z. Yang, M. Chen, W. Saad, C. S. Hong, M. Shikh-Bahaei, H. V. Poor, and S. Cui, ''Delay minimization for federated learning over wireless communication networks,'' 2020, *arXiv:2007.03462*.

[29] C.-L. Chen, C. G. Brinton, and V. Aggarwal, ''Latency minimization for mobile edge computing networks,'' *IEEE Trans. Mobile Comput.*, vol. 22, no. 4, pp. 2233–2247, Apr. 2023.

[30] C. Zhang, S. Dang, B. Shihada, and M.-S. Alouini, ''Dual attention-based federated learning for wireless traffic prediction,'' in *Proc. IEEE Conf. Comput. Commun.*, May 2021, pp. 1–10.

[31] S. Dash, A. U. Khan, S. K. Swain, and B. Kar, ''Clustering based efficient MEC server placement and association in 5G networks,'' in *Proc. 19th OITS Int. Conf. Inf. Technol. (OCIT)*, Dec. 2021, pp. 167–172.

[32] A. K. Singh and K. K. Nguyen, ''Joint selection of local trainers and resource allocation for federated learning in open RAN intelligent controllers,'' in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2022, pp. 1874–1879.

[33] Z. Liu, L. Wan, J. Guo, F. Huang, X. Feng, L. Wang, and J. Ma, ''PPRU: A privacy-preserving reputation updating scheme for cloud-assisted vehicular networks,'' *IEEE Trans. Veh. Technol.*, early access, Dec. 8, 2023, doi: 10.1109/TVT.2023.3340723.

[34] J. Guo, Z. Liu, S. Tian, F. Huang, J. Li, X. Li, K. K. Igorevich, and J. Ma, ''TFL-DT: A trust evaluation scheme for federated learning in digital twin for mobile networks,'' *IEEE J. Sel. Areas Commun.*, vol. 41, no. 11, pp. 3548–3560, Nov. 2023.

[35] Y. Miao, R. Xie, X. Li, Z. Liu, K.-K.-R. Choo, and R. H. Deng, ''Efficient and secure federated learning against backdoor attacks,'' *IEEE Trans. Dependable Secure Comput.*, early access, Jan. 16, 2024, doi: 10.1109/TDSC.2024.3354736.

[36] J. Lee, F. Solat, T. Y. Kim, and H. V. Poor, ''Federated learning-empowered mobile network management for 5G and beyond networks: From access to core,'' *IEEE Commun. Surveys Tuts.*, early access, Jan. 16, 2024, doi: 10.1109/COMST.2024.3352910.

[37] A. M. Elbir, S. Coleri, and K. V. Mishra, ''Hybrid federated and centralized learning,'' in *Proc. 29th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2021, pp. 1541–1545.

[38] W. Hong, X. Luo, Z. Zhao, M. Peng, and T. Q. S. Quek, ''Optimal design of hybrid federated and centralized learning in the mobile edge computing systems,'' in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2021, pp. 1–6.

[39] T. K. Rodrigues and N. Kato, ''Hybrid centralized and distributed learning for MEC-equipped satellite 6G networks,'' *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 1201–1211, Apr. 2023.

[40] J. Kim, D. Kim, J. Lee, and J. Hwang, ''A novel joint dataset and computation management scheme for energy-efficient federated learning in mobile edge computing,'' *IEEE Wireless Commun. Lett.*, vol. 11, no. 5, pp. 898–902, May 2022.

[41] S. Luo, X. Chen, Q. Wu, Z. Zhou, and S. Yu, ''HFEL: Joint edge association and resource allocation for cost-efficient hierarchical federated edge learning,'' *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6535–6548, Oct. 2020.

[42] D. P. Bertsekas, ''Nonlinear programming,'' *J. Oper. Res. Soc.*, vol. 48, no. 3, p. 334, Dec. 1997.

[43] C. Wang, S. Zhang, Z. Qian, M. Xiao, J. Wu, B. Ye, and S. Lu, ''Joint server assignment and resource management for edge-based MAR system,'' *IEEE/ACM Trans. Netw.*, vol. 28, no. 5, pp. 2378–2391, Oct. 2020.

[44] S. Kalra, J. Wen, J. C. Cresswell, M. Volkovs, and H. R. Tizhoosh, ''Decentralized federated learning through proxy model sharing,'' *Nature Commun.*, vol. 14, no. 1, p. 2899, 2023.

[45] C. Dwork and A. Roth, ''The algorithmic foundations of differential privacy,'' *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.

[46] L. Deng, ''The MNIST database of handwritten digit images for machine learning research,'' *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.

[47] H. Xiao, K. Rasul, and R. Vollgraf, ''Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms,'' 2017, *arXiv:1708.07747*.

[48] A. Krizhevsky and G. Hinton, ''Learning multiple layers of features from tiny images,'' Dept. Comput. Sci., Univ. Toronto, Tech. Rep. UTML TR 2009-003, Apr. 2009.

FARANAKSADAT SOLAT received the B.S. degree in industrial engineering from Alzahra University, in 2017, the M.S. degree in information technology engineering from Iran University of Science and Technology (IUST), Tehran, Iran, in 2019, and the Ph.D. degree from the School of Computing, Gachon University, Seongnam, South Korea, in 2022. Her research interests include system optimization, machine learning, FL, 5G/6G network management, and reinforcement learning.

SAKSHI PATNI received the bachelor's degree in computer applications from Punjab University, Chandigarh, in 2012, the M.C.A. degree in computer applications from Banasthali University, India, in 2015, and the Ph.D. degree from the National Institute of Technology, Kurukshetra, in 2020. From 2021 to March 2023, she was an Assistant Professor with the Engineering Institutes. Currently, she is a Research Professor with the Department of Computing, Gachon University, Seongnam, South Korea. She has published various research papers in SCI, Scopus journals, and international conferences. Her main research interests include cloud computing, load balancing, resource management, FL, and information security. She received the Best Paper Award at the IEEE International Conference ICECCS organized in Malaysia.

SUNHWAN LIM received the Ph.D. degree from Chungnam National University, Daejeon, South Korea, in 2011. Since April 1999, he has been the Principal Researcher (Chief Technical Staff) of the Electronics and Telecommunications Research Institute (ETRI). He is currently the Head of research on "Development of Collective Collaboration Intelligence Framework for Internet of Autonomous Things" and "Development of 5G-IoT Trustworthy AI-Data Commons Framework." His recent topics of interest are AI, AI-data commons, and digital twin. He serves as a member of the Institute of Information and Communications Technology Planning and Evaluation (IITP)'s assessment committee. Also, he is the Chairman of the IoT Platform Working Group (WG) for the IoT Convergence Forum.

JOOHYUNG LEE (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2008, 2010, and 2014, respectively. From 2012 to 2013, he was a Visiting Researcher with the Information Engineering Group, Department of Electronic Engineering, City University of Hong Kong, Hong Kong. From 2014 to 2017, he was a Senior Engineer with Samsung Electronics. He is currently an Associate Professor with the School of Computing, Gachon University, South Korea, and a Visiting Fellow with the Department of Electrical and Computer Engineering, Princeton University. He has contributed several articles to the International Telecommunication Union Telecommunication (ITU-T) and the 3rd Generation Partnership Project (3GPP). His current research interests include resource allocation, optimization, and protocol design, with a focus on resource management for machine learning, including FL, 6G networks, cloud/edge computing, smart grids, augmented reality, virtual reality, and network economics. He received the Best Paper Award at the Integrated Communications, Navigation, and Surveillance Conference, in 2011, and the Award for Outstanding Contribution in Reviewing at *Computer Communications* (Elsevier). He has been a technical reviewer for several conferences and journals.

• • •