Optimizing Federated Semantic Learning in Distributed AIGC-Enabled Human Digital Twins: A Multi-Criteria and Multi-Shard User Selection Framework

Samuel D. Okegbile, *Member, IEEE*, Haoran Gao, Oluwasegun Talabi, Jun Cai, *Senior Member, IEEE*, Dusit Niyato, *Fellow, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—Artificial intelligence-generated content (AIGC) has been proposed as a solution to meet the requirements of ultra-reliable, secure, and privacy-preserving connectivity in human digital twin (HDT) networks. In such an AIGC-enhanced HDT, contents representing the true statuses of physical twins are generated in the virtual environment for the immediate update and evolution of the corresponding virtual twins (VTs). However, adopting a distributed AIGC in HDT presents several challenges, including the need for personalized VTs, data privacy concerns, and insufficient contextual understanding. This paper introduces a multi-layer federated semantic learning framework to address these challenges, incorporating batch learning to meet the training requirements for semantic-channel encoders and decoders. Furthermore, we introduce a novel user association framework to maximize the overall system performance under shard formation constraints. We then formulate a long-term joint optimization problem for user selection over finite learning periods. A novel Lyapunov-based online optimization strategy was proposed to mitigate the impact of time-varying and unpredictable training conditions. Additionally, we introduce a multi-arm bandit-based method and a context-centric user selection approach to solve the optimization problem. The results demonstrate that the proposed user association framework addresses the limitations of existing approaches, thereby improving the overall performance of the multi-shard AIGC-enhanced HDT.**

Received 4 December 2024; revised 26 January 2025; accepted 3 February 2025. Date of publication 11 February 2025; date of current version 5 June 2025. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant, in part by the Concordia University PERFORM Research Chair Program, in part by the Singapore Ministry of Education (MOE) Tier 1 (RG87/22 and RG24/24), in part by the NTU Centre for Computational Technologies in Finance (NTU-CCTF), in part by Seitee Pte Ltd, and in part by the RIE2025 Industry Alignment Fund - Industry Collaboration Projects (IAF-ICP) under Grant I2301E0026, administered by A*STAR. Recommended for acceptance by S. Wang. *(Corresponding author: Jun Cai.)*

Samuel D. Okegbile was with the Department of Electrical and Computer Engineering, Concordia University, Montreal, QC H3G 1M8, Canada. He is now with the School of Computing, University of the Fraser Valley, Abbotsford, BC V2S 7M8, Canada (e-mail: samuel.okegbile@ufv.ca).

Haoran Gao, Oluwasegun Talabi, and Jun Cai are with the Department of Electrical and Computer Engineering, Concordia University, Montreal, QC H3G 1M8, Canada (e-mail: haoran.gao@mail.concordia.ca; oluwasegun.talabi@mail.concordia.ca; jun.cai@concordia.ca).

Dusit Niyato is with the College of Computing and Data Science, Nanyang Technological University,, Singapore 639798 (e-mail: dniyato@ntu.edu.sg).

Xuemin Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

Digital Object Identifier 10.1109/TMC.2025.3541191

***Index Terms*—Artificial intelligence-generated content (AIGC), data batching, federated semantic learning, human digital twin (HDT), multi-arm bandit, user association.**

I. Introduction

THE concept of human digital twins (HDT) continues to receive much attention due to its ability to revolutionize every human-centric system [1]. When adopted in healthcare, HDT can improve the existing personalized healthcare services. It can enhance health monitoring, predict diseases, personalize treatments, and optimize well-being while transforming healthcare services through data-driven insights and interventions [2]. Generally, HDT aims to create a true replica of each individual or human organ (for application-specific HDT systems), referred to as the virtual twin (VT). This VT is maintained in a virtual environment and relies on ultra-reliable connectivity between the physical and virtual environments. The physical counterpart, called the physical twin (PT), interacts with the VT in real-time to enable this seamless connection and representation. However, achieving timely PT-VT synchronization is challenging due to the requirement for ultra-reliable, secure, and privacy-preserving data sharing for VT evolution. Therefore, an innovative approach is essential for extracting data and information in the virtual environment (without the need for frequent PT-VT data transmissions) to support the evolution and updating processes of VTs.

One useful approach is the incorporation of artificial intelligence-generated content (AIGC) solutions. AIGC utilizes advanced artificial intelligence (AI) algorithms for efficient content generation and may be adopted in HDT to generate contents that represent the true statuses of PTs. Subsequently, the generated contents for each PT could be used to update its corresponding VT. Generally, AIGC can streamline reliable content generation in virtual environments, minimizing the need for timely synchronization between each PT-VT pair. However, adopting AIGC in HDT presents several challenges, including concerns about realizing the personalization requirement of each VT [3], [4], [5], data privacy, the imperative for robust cybersecurity measures, insufficient contextual understanding, low precision in content generation, a lack of suitable user engagement, and

high ambiguity in user inputs [6]. These challenges may result in the generation of content that fails to convey the intended meaning in the physical environment thereby contributing to communication ineffectiveness.

To address these challenges, federated semantic learning (FSL) [6], [7] has been identified as a practical solution. FSL integrates a multi-user semantic communication system and federated learning (FL) to enhance the learning experience among multiple semantic-channel encoders and decoders following a collaborative and efficient multi-layer learning framework. This innovative paradigm has the potential to revolutionize traditional collaborative learning by prioritizing semantic understanding and knowledge sharing among decentralized devices. FSL can facilitate the training of multiple customized AIGC models (and thus the evolution of the associating VT models) while preserving data privacy, enabling various users or devices to contribute valuable insights and knowledge without compromising individual information. The training of multiple customized AIGC models can be achieved through the formation of multiple shards (i.e., clusters or groups) where each shard is composed of multiple semantic-channel encoders and a decoder while multiple decoders in different shards share their learning experiences among each other to improve data reconstruction and content generation. Such an approach can improve model personalization and accuracy in each shard by capturing rich contextual nuances. By promoting cross-device semantic coherence, a multi-layer FSL empowers a diverse network of semantic-channel encoders and decoders to collaboratively learn and improve without centralizing sensitive data. This pioneering technique paves the way for efficient, privacy-preserving, and contextually aware learning across a decentralized HDT environment.

Consider a typical HDT-driven personalized healthcare monitoring application scenario, where each individual has a VT that simulates their health status, behaviors, vital metrics, etc. We can put each individual into different clusters based on specific characteristics such as chronic conditions, history of cardiovascular diseases, service required, etc. Thus, each customized AIGC model within each cluster aims to generate contents specific to its cluster, tailoring the model's learning and predictions based on the unique health context of its group. By using multi-shard learning, the system can efficiently allocate resources. Each shard will process data relevant to its specific group, ensuring that the AIGC model does not overburden itself with irrelevant data. This leads to faster, more accurate personalized insights and recommendations, such as suggesting lifestyle changes, medication adjustments, or preventative measures.

Despite this, the integration of a multi-layer FSL into AIGC-enhanced HDT requires periodic or on-demand training (depending on the application scenarios) of semantic-channel encoders and decoders to maintain updated customized AIGC models and the counterpart VT models. This training is contingent upon the availability of sufficient new data in the physical environment to ensure the reliable and efficient encoding and decoding processes executed by semantic transmitters and receivers, respectively. However, this periodic and/or on-demand training can potentially disrupt content generation and thus timely PT-VT synchronizations across multiple shards, as full-fledged semantic transmission during training in a shard may not accurately represent real-world scenarios. To mitigate disruptions while adapting to the dynamic environment, we introduce the concept of batch learning. In batch learning, training occurs in batches of shards to minimize the overall system disruptions, reducing the maximum number of shards in the training phase at any given time.

A. Motivation and Contributions

While the introduction of sharding and multi-layer FSL techniques can facilitate the creation and maintenance of multiple customized AIGC models, essential for maintaining similar but diverse VTs, such a solution heavily relies on an optimal user selection process. In HDT, any typical PT-VT pair requires the support of other related PT-VT pairs to improve its VT model accuracy and efficiency. Achieving an optimized user selection, however, is not straightforward. In a multi-shard AIGC-enhanced HDT (MAH), an effective user selection process must consider factors such as semantic understanding, data context, user profiles, and preferences when assigning users to each shard. Nonetheless, selecting suitable users to optimize content generation in each shard proves challenging due to differences in criteria or features required to maintain each customized AIGC model and the inherent heterogeneity among users. One possible solution is to implement a random selection strategy [8]. Adopting a random selection strategy however poses significant issues given the diversity in data quality, as well as computational and communication resources among users [9]. To address such issues, a unilateral selection approach [8] may be adopted. This approach focuses solely on the server's perspective in designing the selection strategy, often neglecting the considerations and interests of the clients involved in the process.

In a typical MAH scenario, clients (equipped with semantic transmitters) exhibit significant heterogeneity in terms of data distribution and hardware configurations. Similarly, servers (equipped with semantic receivers) located in edge servers have diverse requirements based on their respective customized AIGC models. Randomly associating clients to each server in each shard during every learning round may not fully exploit the local updates from heterogeneous clients, resulting in lower model accuracy, a slower convergence rate, and inefficient content generation for VT evolution. To address this client heterogeneity problem, a more suitable user association algorithm[1] should be developed. We hence propose a new multi-criteria and bilateral user selection framework for MAH. To further enhance the learning and content generation processes in each shard, we introduce a multi-layer learning approach following the concept of client-edge-cloud collaboration. For each shard, one semantic receiver is grouped with multiple semantic transmitters based on the proposed multi-criterion approach.

However, integrating a multi-criteria solution for MAH is complicated since different temporal user association patterns

[1]For convenience, we use the terms user association and user selection interchangeably in this paper. We also used the terms user and client interchangeably.

lead to considerably different learning performances [10], and user association is critical in determining overall training and system efficiency [11]. Increasing the number of clients in each shard can enhance training accuracy, convergence stability, and a more expansive customized AIGC model but comes at the expense of increased training time and energy costs. This introduces training-operations complexity, as prolonged training time in each shard increases its hibernation period. Therefore, we aim to select an optimal number of clients contributing to the training process subject to the various user selection constraints knowing that a larger number of users can potentially improve accuracy and convergence stability while increasing the rate of shard hibernation. This paper presents a multi-criteria solution for MAH that intends to enhance performance while minimizing disruptions to shard operations. The main contributions of this paper are summarized as follows.

- We present a novel user association framework for MAH systems, aiming to maximize the overall system performance under shard formation constraints. Our approach involves the development of a multi-criteria solution, considering key selected constraints: data context, intra-shard diversity, time and energy cost, and client distributions. These factors significantly impact the overall performance of the proposed MAH system.
- We formulate a long-term joint optimization problem for user selection and shard hibernation over a finite number of learning periods, considering stochastic learning rates and finite shard formation constraints. To address the resulting problem, we leverage training-operation specifications to provide a long-term performance guarantee.
- To limit the impact of the time-varying and unpredictable training-operation conditions, we propose a novel Lyapunov-based online optimization. This approach aims to maximize the number of users in each shard to enhance training performance while minimizing the average shard hibernation period to improve the content generation capacity and reliability of each customized AIGC model. We introduce a multi-arm bandit-based method and a context-centric user selection approach to provide solutions for the optimization problem.
- We implement a use case for the FSL-enhanced MAH and subsequently present some results to demonstrate the effectiveness of our proposed solutions over existing ones.

We strongly believe that the presented solution can be adopted to improve content generation processes in various AIGC applications and systems with minimal or no modifications.

B. Organization

The remainder of this article is organized as follows. Section II provides a summary of related work. In Section III, we present the system model, followed by the performance analysis and problem formulation in Section IV. Section V delves into the proposed algorithm designed to address the formulated problem. Details of the considered use case and the outcomes of the performance evaluation are discussed in Section VI. Lastly, Section VII concludes this article. The main notations used in this article are summarized in Table I.

TABLE I
SUMMARY OF NOTATION

Notation	Definition
K	Number of semantic transmitters (i.e., clients)
E	Number of edge servers
$\mathcal{D}_k$; $\mathcal{D}_e$	Local dataset of client k; dataset of edge e
h_k	Fading coefficient
n_k	Independent & identically distributed Gaussian noise
$\theta_t^{(k)}$	Parameter vector of client k
α_k	Semantic transmitter network parameter
β_k	Semantic receiver network parameter
η	Learning rate
D_k; D_e	Number of samples in $\mathcal{D}_k$; number of samples in $\mathcal{D}_e$
$N_{\mathcal{S}_e}$	Total number of clients in shard $\mathcal{S}_e$
tr_e	Shard training rate
f_r^k	CPU frequency required to execute one sample of $\mathcal{D}_k$
f_k	CPU frequency of client k
$x^{(k)}$	Matrix representing source input data of $k-$th user
b_k	Proportion of communication resources allocated k

II. Related Work

User selection has received significant attention, particularly in FL, given the crucial task of identifying appropriate users or devices for participation in collaborative model training processes. The adoption of FL in AIGC systems has been considered to address the centralized training limitations of traditional AIGC service [12], to enhance privacy [13], and to facilitate edge-driven personalized customization models [14]. However, finding a balance between the imperative for model improvement and considerations such as privacy and resource constraints is challenging [6]. In FL, the detection of critical learning periods has been mentioned as an important task to achieve adaptive device selection where identifying critical learning periods is usually integrated to enhance federated optimization methods and detect critical learning periods. A joint client selection and resource management solution in wireless FL for the Internet of Things was proposed in [11] where the relationship between training efficiency, number of selected clients and total energy consumption were discussed.

In a related study [10], the authors formulated a stochastic optimization problem for joint client selection and bandwidth allocation. A bandit approach in [15] carried out client selection based on a time-varying reward influenced by the history of previous selections. The objective of this client selection problem is to schedule a subset of clients for training and transmission at each given time to optimize learning performance. Clients were managed based on their resource conditions in [16] while a multicriteria-based optimization model that allows maximization of the number of FL clients with sufficient resources and time to complete the training tasks without dropout was presented in [17].

In addition, the authors in [18] investigated the client selection problem, considering effective participation and fairness by selecting clients with the lowest failure probability. An automated and quality-aware client selection framework for efficient FL was proposed in [19] to evaluate the learning quality of clients within a limited budget. Similar works, such as [20], modeled fairness-guaranteed client selection as a Lyapunov optimization problem, and [21] explored adaptive client selection. The presence of a single aggregation server in these FL-based selection approaches means these solutions are not suitable in MAH where user association between multiple clients and multiple servers is necessary to meet the system-specific requirement.

The study presented in [22] introduced an FSL framework designed to collaboratively train the semantic-channel encoders across multiple devices. This collaborative training is coordinated via a base station-enabled semantic-channel decoder. Additionally, the authors in [23] delved into the exploration of a multi-user semantic communication system specifically tailored for executing object-identification tasks. Furthermore, the work in [24], [25] discussed a task-oriented multi-user semantic communication system for multimodal data transmission. The presented solution involved the utilization of a deep neural network-enabled semantic communication system to efficiently carry out various tasks. Such solutions [24], [25] also extended to the examination of a deep learning-based multi-user semantic communication system, evaluating its effectiveness in transmitting both single-modal and multimodal data.

Nevertheless, these works are also constrained by a single receiver equipped with a semantic-channel decoder while none have explored the benefits of multi-layer training to enhance the learning experience of multiple semantic transmitters and receivers. Furthermore, there is still a lack of identified user association or selection mechanisms suitable for adoption in distributed AIGC systems while existing works on AIGC have mainly focused on the deployment of AIGC models without addressing the necessity of regularly retraining these models to prevent performance degradation over time.

III. System Model

We consider the deployment of an FSL-enabled MAH with a multi-layer framework following the concept of the client-edge-cloud collaboration [26], [27], [28], [29] as shown in Fig. 1, where multiple PTs, through their respective source users (i.e., clients equipped with their respective semantic transmitters), collaborate to form a shard with the corresponding destination user (i.e., edge server containing the semantic receiver, the associated customized AIGC model and VT models). This facilitates the maintenance of an updated customized AIGC model, essential for generating timely content needed for the evolution of their individual VT models. Unlike traditional communication methods which transmit raw data, leading to high bandwidth demand, and insufficient privacy and security, semantic communication focuses on conveying the intended meaning, prioritizing understanding and relevance over exact data fidelity. This makes such a method suitable for MAH. The AIGC services include the model initialization, configuration and inference [12]. As

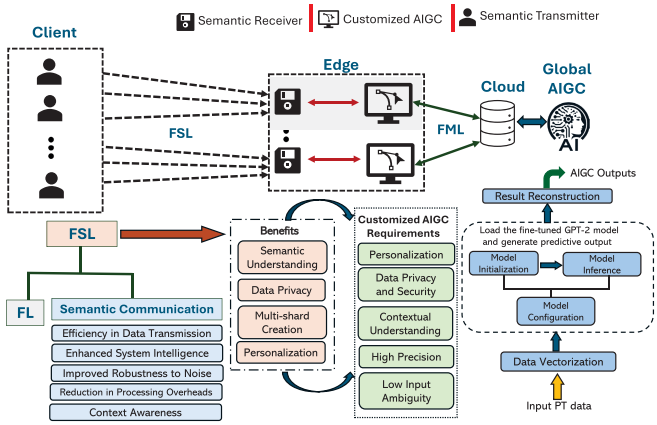


Fig. 1. General framework of FSL-enabled MAH.

shown in Fig. 2, the client layer consists of K semantic transmitters, given as $\mathcal{K} = \{1, \ldots, K\}$, each containing its associated semantic-channel encoder. Similarly, the edge layer includes E edge servers, also given as $\mathcal{E} = \{1, \ldots, E\}$, each consisting of a semantic receiver as well as its corresponding semantic-channel decoder and semantic knowledge base (KB), which facilitates the training and operations of the associated customized AIGC model. At the same time, the cloud layer is made up of the cloud server and contains the global AIGC model and KB that are also useful in improving the learning experience at each edge server $e \in \mathcal{E}$ via the federated multi-task learning (FML) [30]. Each participating client $k \in \mathcal{K}$ has a local dataset $\mathcal{D}_k$. In the supervised learning case, $\mathcal{D}_k$ defines the collection of datasets given as a set of input-output pairs $\{x_i, y_i\}_{i=1}^{D_k}$, where $x_i \in \mathbb{R}^d$ is a $d-$dimensional input feature vector and $y_i \in \mathbb{R}$ is the output label. We know that the source input data of the $k-$th user can be denoted as $x^{(k)}$ such that the extracted semantic symbol is given as

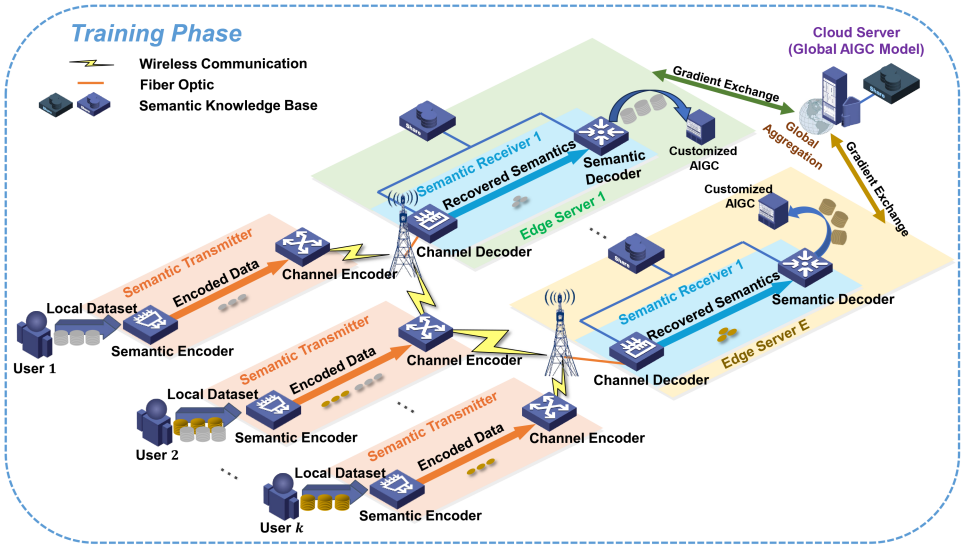
$$s_k = \mathcal{T}_{\alpha^{(k)}}(x^{(k)}), \forall k \in \mathcal{K}, \tag{1}$$

where $\mathcal{T}_{\alpha^{(k)}}(\cdot)$ represents the semantic transmitter network with the parameter set $\alpha^{(k)}$ for the $k-$th source user. If the $k-$th transmitter sends s_k to its paired semantic receiver network over a wireless channel with additive noise following the traditional task-oriented semantic communication framework, then the received semantic feature is

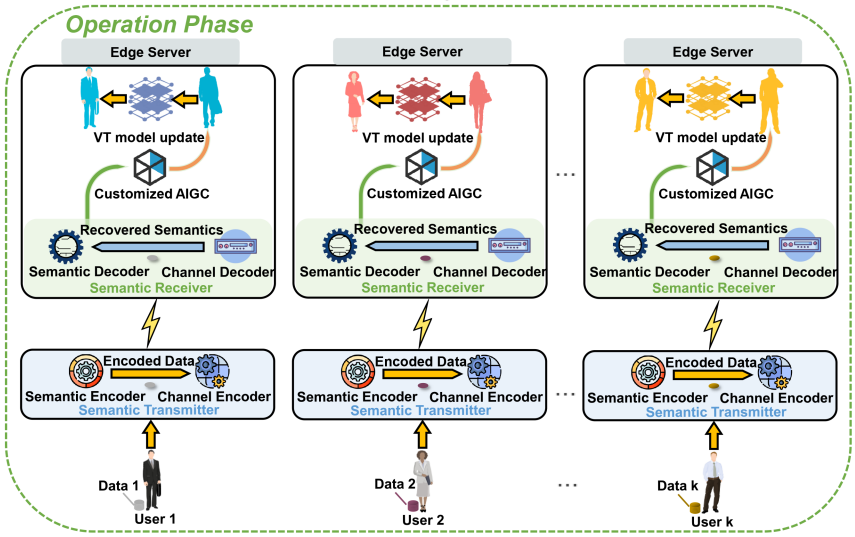
$$\bar{s_k} = \mathcal{R}_{\beta^{(k)}}(h_k s_k + n_k), \forall k \in \mathcal{K}, \tag{2}$$

where $\mathcal{R}_{\beta^{(k)}}(\cdot)$ is the semantic receiver network with the parameter set $\beta^{(k)}$, h_k denotes the fading coefficient, and $n_k \sim \mathcal{N}(0, \sigma_k^2)$ is independent and identically distributed Gaussian noise of variance σ_k^2.

The proposed FSL-enabled MAH framework depicted in Fig. 1 comprises two distinct phases: training and operation. The training phase, depicted in Fig. 2(a), utilizes a multi-layer framework, where the first layer learning occurs between client-edge nodes using the FSL, while the second layer learning between edge-cloud nodes employs the FML.



(a) Training Phase



(b) Operation Phase

Fig. 2. FSL-enabled MAH framework.

A. Training Phase Modeling

While semantic and channel encoders/decoders have distinct roles within semantic communications, they are often jointly trained in practice. In Fig. 2, they are shown separately to highlight that semantic encoder outputs feed channel encoders, and channel decoder outputs feed semantic decoders. Hence, in the training phase, multiple semantic transmitters (i.e., clients), each containing a semantic-channel encoder, and a semantic receiver participate in collaborative training to improve semantics encoding and decoding processes while updating the associated AIGC model, as depicted in Fig. 2(a). Each receiver, situated in its respective edge server, also contains a semantic-channel decoder engineered to extract features that optimize the evolution and content generation of its customized AIGC model, simultaneously enhancing its decoding efficiency and accuracy. At each communication round $t \in \{1, 2, \ldots, T\}$, we may update the model parameters iteratively by taking small steps in the direction that minimizes the loss, following the stochastic gradient descent update rule

$$\theta_{t+1}^{(k)} = \theta_t^{(k)} - \eta \nabla_{\theta^{(k)}} \mathcal{L}_t^{(k)}(\theta_t^{(k)}), \tag{3}$$

where $\theta_t^{(k)} = (\alpha_t^{(k)}, \beta_t^{(k,e)})$ is the parameter vector of user k at time step t, $\nabla_{\theta^{(k)}} \mathcal{L}_t^{(k)}(\theta_t^{(k)})$ is the gradient of the loss function with respect to the parameter vector $\theta_t^{(k)}$, and η is the learning rate, controlling the size of the update steps [31], [32].

However, it is important to note that each receiver is constrained by the context of its associated AIGC model and benefits more from users with datasets containing similar contexts (e.g., similar distributions, features, or domain-specific characteristics) since mixing highly heterogeneous data may confuse the model and hinder its performance. As a result, we incorporate a sharding technique, where all clients are divided into smaller,

more manageable groups, called shards, based on multiple criteria. Each shard is an independent group of multiple clients and a server. At any round t, the set of shards is defined as $\{\mathcal{S}_1(t), \ldots, \mathcal{S}_E(t)\}$. The difference between the original data $x^{(k)}$ and the one generated by the receiving customized AIGC model $\bar{x^{(k)}}$ is measured using the mean squared error (MSE) loss, as

$$\mathcal{L}_t^{(k,e)}\left(\theta_t^{(k)}, \mathcal{S}_e(t)\right) = \frac{1}{\sum_{k\in\mathcal{K}} \tau_{\mathcal{S}_e}^k D_k} \sum_{k=1}^{N_{\mathcal{S}_e}} \tau_{\mathcal{S}_e}^k \sum_{i=0}^{D_k-1} \left(x_i^{(k)} - \bar{x_i^{(k)}}\right)^2, \quad (4)$$

where the parameter $N_{\mathcal{S}_e} \leq K$ is the total number of clients in shard $\mathcal{S}_e$ and $\tau_{\mathcal{S}_e}^k \in \{0, 1\}$ represents a client participating indicator, obtained as

$$\tau_{\mathcal{S}_e}^k = \begin{cases} 1, & \text{if client } k \text{ is active in shard } \mathcal{S}_e \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

In addition, each edge server collaborates with other available edge servers and the cloud server, following the FML, to synchronize their experiences, enhancing learning across each shard. Since FML aims to minimize the overall loss across all servers while considering their individual tasks, we define a global loss function $\mathcal{L}^{(c)}$ that aggregates the losses from each server as

$$\mathcal{L}_t^{(\text{global})}(\theta_t^{(c)}) = \sum_{e\in\mathcal{E}} \tau_e \omega_e \mathcal{L}_t^{(k,e)}\left(\theta_t^{(k)}, \mathcal{S}_e(t)\right) + \omega_c \mathcal{L}_t^{(c)}\left(\theta_t^{(c)}\right), \quad (6)$$

where ω_e and ω_c are weights assigned to each edge server and the cloud server, respectively, $\tau_e \in \{0, 1\}$ indicates whether an edge server participates in the edge-cloud training, and $\theta_t^{(c)}$ represent the parameters of the model in the cloud server. The loss function $\mathcal{L}_t^{(c)}(\theta_t^{(c)})$ for the cloud server can be expressed as

$$\mathcal{L}_t^{(c)}\left(\theta_t^{(c)}\right) = \frac{1}{|\mathcal{D}_{\text{cloud}}|} \sum_{(x,y)\in\mathcal{D}_{\text{cloud}}} \mathcal{L}_t\left(f_{\text{cloud}}(x; \theta_t^{(c)}), y\right), \quad (7)$$

where $\mathcal{D}_{\text{cloud}} \triangleq \bigcup_{e\in\mathcal{E}} \mathcal{D}_e$ represents the dataset at the cloud server, which is the aggregation of the recovered semantic data of each edge server $\mathcal{D}_e$, and $f_{\text{cloud}}(x; \theta_t^{(c)})$ is the output of the model on sample x of label y at the cloud server.

The gradient loss parameter $\nabla_{\theta^{(k)}} \mathcal{L}_t^{(k,e)}(\theta_t^{(k)}, \mathcal{S}_e(t))$ can be expressed, from (4), as

$$\nabla_{\theta^{(k)}} \mathcal{L}_t^{(k,e)}\left(\theta_t^{(k)}, \mathcal{S}_e(t)\right) = \frac{2}{\sum_{k\in\mathcal{K}} \tau_{\mathcal{S}_e}^k D_k} \times \sum_{k=1}^{N_{\mathcal{S}_e}} \tau_{\mathcal{S}_e}^k \sum_{i=0}^{D_k-1} \left(x_i^{(k)} - \bar{x_i^{(k)}}\right) \nabla_{\bar{x_i^{(k)}}}, \quad (8)$$

where $\nabla_{\bar{x_i^{(k)}}}$ is the partial derivative of $\bar{x_i^{(k)}}$. Similarly, the gradient for the cloud server loss can be computed from (7) as

$$\nabla_{\theta^{(c)}} \mathcal{L}_t^{(c)}\left(\theta_t^{(c)}\right) = \frac{1}{|\mathcal{D}_{\text{cloud}}|} \sum_{(x,y)\in\mathcal{D}_{\text{cloud}}} \nabla_{\theta^{(c)}} \mathcal{L}_t\left(f_{\text{cloud}}(x; \theta_t^{(c)}), y\right), \quad (9)$$

with $\nabla_{\theta^{(c)}} \mathcal{L}_t(f_{\text{cloud}}(x; \theta_t^{(c)}), y)$ representing the gradient of the loss function concerning the output of the model at the cloud server. The learning process of each semantic transmitter $k \in \mathcal{K}$ and semantic receiver $e \in \mathcal{E}$ in any typical shard $\mathcal{S}_e$ can thus be captured respectively as

$$\alpha_{t+1}^{(k)} = \alpha_t^{(k)} - \eta\Big(\omega_e \nabla_{\theta^{(k)}} \mathcal{L}_t^{(k,e)}\left(\theta_t^{(k)}, \mathcal{S}_e(t)\right) + \omega_c \nabla_{\theta^{(c)}} \mathcal{L}_t^{(c)}\left(\theta_t^{(c)}\right)\Big), \quad (10)$$

$$\beta_{t+1}^{(k,e)} = \beta_t^{(k,e)} - \eta\Big(\omega_e \nabla_{\theta^{(k)}} \mathcal{L}_t^{(k,e)}\left(\theta_t^{(k)}, \mathcal{S}_e(t)\right) + \omega_c \nabla_{\theta^{(c)}} \mathcal{L}_t^{(c)}\left(\theta_t^{(c)}\right)\Big). \quad (11)$$

It is found that it may not always be necessary to select the maximum number of clients for each shard during training [10]. Thus, the shard formation should account for both statistical and system heterogeneity. Numerous clients may contribute similar and redundant gradient information during aggregation, failing to accurately represent the true data distribution globally [21]. Choosing such clients can result in resource wastage and bias the global model at the corresponding receiver towards specific clients, resulting in global model degradation. Therefore, a user selection approach must prioritize intra-shard diversity to choose representative clients for each shard while adhering to resource constraints. For each shard, the objective is to identify a diverse subset of clients whose aggregated model updates approximate the aggregated updates of all clients. Let the model update of kth client be given as $\Delta\theta^{(k)}$, the diversity between the model updates of any two clients, k and j, can be computed following the cosine similarity scoring [33] as

$$CS(\Delta\theta^{(k)}, \Delta\theta^{(j)}) = \frac{\Delta\theta^{(k)} \cdot \Delta\theta^{(j)}}{\|\Delta\theta^{(k)}\| \|\Delta\theta^{(j)}\|}, \quad (12)$$

where $\Delta\theta^{(k)} \cdot \Delta\theta^{(j)}$ represents the dot product between any two vectors $\Delta\theta^{(k)}$ and $\Delta\theta^{(j)}$, and $\|\cdot\|$ signifies the euclidean norm. From (3), the diversity score for each client can be obtained as

$$DS(k) = \sum_{k\neq j} \left(1 - CS(\Delta\theta^{(k)}, \Delta\theta^{(j)})\right). \quad (13)$$

The diverse subset of clients may then be obtained by selecting the clients with the highest diversity scores such that the diverse subset of clients is defined as

$$D_{\text{sub}} = \{k | DS(k) \geq div_{th}\}, \quad (14)$$

where div_{th} is a predetermined threshold value that determines the level of diversity required for a client to be included in the subset. By ensuring intra-shard diversity during client selection, we can minimize redundant communication and increase the impact of under-represented clients that contribute distinct information, thus enhancing overall learning efficiency.

Although diversity enhances learning accuracy, client selection plays a crucial role in determining the overall training time, given variations among clients in terms of dataset size,

available resources, channel state, and energy availability [15]. Selecting an higher parameter div_{th} may lead to the straggler problem, significantly increasing the training latency and the overall energy consumption during training. Hence, a trade-off exists between the number of selected clients (through div_{th}) and total energy consumption.

B. Operation Phase Modeling

The objective of the training phase is to optimize features encoding and decoding during operations: semantic transmission and content generation for the evolution of VT models. Subsequently, in the operational phase, each transmitter extracts semantics from its data and transmits this semantic information to its paired receiver for feature reconstruction and content generation, through the corresponding customized AIGC model. More specifically, each semantic encoder, associated with a client $k \in \mathcal{K}$, takes as input the local dataset $\mathcal{D}_k$ and outputs a set of important features extracted from $\mathcal{D}_k$. These extracted feature vectors are then fed into the counterpart channel encoder to manage the effects of channel noise and interference in the wireless environment over the transmitted symbol. The semantic-channel encoder operation is summarized in (1).

At the receiver side, when considering the additive white Gaussian noise and Rayleigh fading channel, the received signal from the client k can be represented as

$$\bar{X_k} = h_k s_k + n_k. \tag{15}$$

The estimated semantic feature $\bar{s_k}$ is then recovered from $\bar{X_k}$ using (2) before employing the customized AIGC model to generate appropriate contents that capture the true states in the physical environment.

C. Hibernation Period Modeling

Considering a typical FSL-enabled MAH system, the system remains in the operation phase at all times, excluding training intervals, called the hibernation period. Thus, the hibernation period is influenced by the training rate of each shard, given as tr_e. To mitigate the risk of service disruption caused by training activities, we propose implementing a partial hibernation method, allowing only selected shards to enter the training phase at any given time. The objective is to minimize the impact of the time each shard spends on training on the overall availability of the system. To achieve this, we introduce batch learning, where training is performed in batches of shards. Batch learning reduces the maximum number of shards actively training at any given time by grouping shards based on their idle periods. Only selected groups train concurrently, limiting system-wide dependency. A scheduler coordinates group training using predicted idle periods, ensuring efficient operation. This approach minimizes disruptions by isolating training to specific groups of shards, allowing other shards to remain operational. Decentralized scheduling and periodic updates prevent system-wide shutdowns, maintaining overall functionality.

Generally, a higher shard training rate $tr_e \in [0, 1]$ may lead to more frequent updates to each node in the network, improving the system's overall responsiveness. However, it also increases the likelihood of system disruptions due to the higher training load. This creates a tradeoff between extending hibernation periods and maintaining timely updates to ensure node accuracy.

IV. Multi-Criteria Scheme Analysis and Problem Formulation

This section analyzes the proposed multi-criteria solution for the multi-shard FSL-enabled MAH system. Subsequently, we will delve into the optimization problem.

A. Analysis of Multi-Criteria Scheme

The proposed multi-criteria scheme relies on four key features: context, intra-shard diversity, cost, and location. Context measures the relevance of client data to a problem, assessed through the domain of the associated customized AIGC model. For any typical shard, the goal of the context criteria is to reduce the distribution difference of each client and the target-domain instances. Given that $\{x_i^k\}_{i=1}^{D_k}$ and $\{x_j^e\}_{j=1}^{D_e}$ are the samples of client $k \in \mathcal{K}$ and its corresponding edge server $e \in \mathcal{E}$, respectively, we can characterize the context using the maximum mean discrepancy (MMD) as

$$M_{MD}(k,e) = \left\| \frac{1}{D_k} \sum_{i=1}^{D_k} \Phi(x_i^k) - \frac{1}{D_e} \sum_{j=1}^{D_e} \Phi(x_j^e) \right\|_{\mathcal{H}}, \tag{16}$$

where $\Phi(\cdot)$ is the feature mapping function that maps data points into the kernel Hilbert space and $\|\cdot\|_{\mathcal{H}}$ denotes the norm in the kernel Hilbert space. Thus, for any typical shard $\mathcal{S}_e$, we need

$$\sum_{k=1}^{K} \tau_{\mathcal{S}_e}^k \left\| \frac{1}{D_k} \sum_{i=1}^{D_k} \Phi(x_i^k) - \frac{1}{D_e} \sum_{j=1}^{D_e} \Phi(x_j^e) \right\|_{\mathcal{H}} \leq c_{th}, \tag{17}$$

where c_{th} represents the maximum degree of discrepancy required among participating clients in each shard.

Next, it is important to measure the degree of intra-shard diversity as selecting the maximum number of clients that satisfies the context requirement may not always be necessary owing to the cost constraint. At every learning round t, it is necessary to identify a subset $\mathcal{M}^{\mathcal{S}_e}$ of clients whose combined gradients can serve as an approximation to the full gradients across all $N_{\mathcal{S}_e}$ clients subject to (14). We consider the existence of a mapping $\pi^{\mathcal{S}_e} : \mathcal{N} \to \mathcal{M}^{\mathcal{S}_e}$ such that the gradients from client $k \in \mathcal{N}$ can be approximated by the gradients from a chosen client $\pi^{\mathcal{S}_e}(k) \in \mathcal{M}^{\mathcal{S}_e}$. Let $\mathcal{A}_k^{\mathcal{S}_e} = \{k \in \mathcal{N}^{\sim} | {}^{\sim}\pi^{\mathcal{S}_e}(n) = k\}$ denote the set of clients approximated by clients $k \in \mathcal{M}^{\mathcal{S}_e}$ and $\gamma_k^{\mathcal{S}_e} = |\mathcal{A}_k^{\mathcal{S}_e}|$. We define the approximation error at learning period l (which consists of T communication rounds) as

$$a_{err}(t) = \left\| \frac{1}{N_{\mathcal{S}_e}} \left(\sum_{k \in \mathcal{M}^{\mathcal{S}_e}} \gamma_k^{\mathcal{S}_e} \nabla \mathcal{L}_t^{(k)} \left(\theta_t^{(k)}, \mathcal{S}_e(t) \right) - \sum_{k \in \mathcal{N}} \nabla \mathcal{L}_t^{(k)} \left(\theta_t^{(k)}, \mathcal{S}_e(t) \right) \right) \right\|. \tag{18}$$

Generally, a_{err} assesses the degree to which the selected subset $\mathcal{M}^{\mathcal{S}_e}$ approximate $N_{\mathcal{S}_e}$ and its upper-bound [21] can be obtained

as

$$a_{err}^{up} \leq \frac{1}{N_{\mathcal{S}_e}} \sum_{k \in \mathcal{N}} \Big\| \nabla \mathcal{L}_t^{(k)} \left(\theta_t^{(k)}, \mathcal{S}_e(t) \right) - \nabla \mathcal{L}_t^{(\pi^{\mathcal{S}_e}(n))} \left(\theta_t^{(\pi^{\mathcal{S}_e}(n))}, \mathcal{S}_e(t) \right) \Big\|. \tag{19}$$

Another parameter that can affect the performance of the multi-criteria scheme is cost. Cost captures both the time and energy consumption in the multi-layer training. This includes the cost associated with local training (i.e., the semantic-channel encoding) at each source user $k \in \mathcal{K}$ of CPU frequency f_k, the computation costs incurred at the edge and cloud servers, as well as the cost of exchanging gradients between client and edge layers, and between edge and cloud layers. Given that f_r^k is the number of CPU frequencies required by client $k \in \mathcal{K}$ to execute one sample of its D_k, the local training time cost over T iterations is

$$T_{lo}(k) = \sum_{t=1}^{T} \tau_{\mathcal{S}_e}^k(t) \frac{f_r^k D_k(t)}{a_e^{(k)}(t) f_k}, \tag{20}$$

where $a_e^{(k)}(t) \in (0, 1]$ is the ratio of the computation resource allocated to the local training of $\mathcal{S}_e$ at round t. The corresponding energy cost is obtained following the energy model of a complementary metal-oxide-semiconductor (CMOS) circuit [34] as

$$E_{lo}(k) = T_{lo}(k) \rho_k a_e^{(k)} f_k^3, \tag{21}$$

where ρ_k represents the effective switched capacitance coefficient of client k depending on the chip architecture. After local training, the gradients are sent to the edge server. The total time and energy costs over T iterations are respectively given as

$$\begin{aligned} T_{tr}(k) &= \sum_{t=1}^{T} \frac{|s_k(t)|}{r_{k,e}(t)}, \forall k \in \mathcal{K}, e \in \mathcal{E}, \\ E_{tr}(k) &= T_{tr}(k) p_k, \forall k \in \mathcal{K}, e \in \mathcal{E}, \end{aligned} \tag{22}$$

where p_k is the transmission power of client k and $r_{k,e}(t)$ represents the client-edge transmission rate. Similarly, the time cost and energy cost for the learning process at the edge $e \in \mathcal{E}$, given that $a_j^{(e)}(t) \in (0, 1]$, is the ratio of its computation resource allocated to the training related to client k in $\mathcal{S}_e$ at iteration t, which can be expressed as

$$\begin{aligned} T_{es}(\mathcal{S}_e) &= \sum_{t=1}^{T} \frac{\tau_e(t) f_r^e \sum_{k \in \mathcal{K}} |s_k(t)|}{a_j^{(e)}(t) f_e}, \\ E_{es}(\mathcal{S}_e) &= T_{es}(\mathcal{S}_e) \rho_e a_j^{(e)} f_e^3, \end{aligned} \tag{23}$$

where f_r^e is the number of CPU frequencies required to process one unit of the sample, $f_e > f_k$ is the CPU frequency of edge $e \in \mathcal{E}$ and ρ_e is its effective switched capacitance coefficient. At the cloud layer, the costs are given as

$$\begin{aligned} T_{cs} &= \sum_{t=1}^{T} \frac{f_r^c \sum_{e \in \mathcal{E}} D_e(t)}{f_c}, \\ E_{cs} &= T_{cs} \rho_c f_c^3, \end{aligned} \tag{24}$$

where f_r^c is the number of CPU frequencies required to process one unit of the sample $\sum_{e \in \mathcal{E}} D_e$ by the cloud server, $f_c \gg f_k$ is the CPU frequency of the cloud server and ρ_c is its effective switched capacitance coefficient. Owing to the higher computational capacity at the edge and cloud layers, we can assume the cost of downlink transmission from the edge to the client layer as well as the edge-cloud communication cost to be negligible. Thus, the overall time cost and energy cost over L learning periods can be respectively obtained as

$$\begin{aligned} T_{ove}(L) = \sum_{l=1}^{L} \sum_{t=1}^{T} \Bigg(\max_{k \in \mathcal{K}} \left(\tau_{\mathcal{S}_e}^k(t,l) \frac{f_r^k D_k(t,l)}{a_e^{(k)}(t,l) f_k} + \frac{|s_k(t,l)|}{r_{k,e}(t,l)} \right) \\ + \max_{e \in \mathcal{E}} \left(\frac{\tau_e(t,l) f_r^e \sum_{k \in \mathcal{K}} |s_k(\bar{t},l)|}{a_j^{(e)}(t,l) f_e} \right) + \frac{f_r^c \sum_{e \in \mathcal{E}} D_e(t,l)}{f_c} \Bigg), \end{aligned} \tag{25}$$

$$\begin{aligned} E_{ove}(L) = \sum_{l=1}^{L} \Bigg[\sum_{k=1}^{K} \tau_{\mathcal{S}_e}^k(l) \Big(T_{lo}(k,l) \rho_k a_e^{(k)}(l) f_k^3 + \\ T_{tr}(k,l) p_k(l) \Big) + \sum_{e \in \mathcal{E}} \tau_e(l) \Big(T_{es}(\mathcal{S}_e, l) \rho_e a_j^{(e)}(l) f_e^3 \Big) \\ + T_{cs}(l) \rho_c f_c^3 \Bigg]. \end{aligned} \tag{26}$$

The last criterion is the client distribution, which is measured as the distance between any client and its associated edge server during training. This criterion is closely related to the cost, as the distance between a client and its edge server within a shard can impact latency and energy consumption, potentially increasing both as the client-edge distance increases. Let $(|x_t^{(k)}|, |y_t^{(k)}|)$ represent the spatial location of any client $k \in \mathcal{K}$ at any iteration t, while $(|x_e|, |y_e|)$ also captures the location of the associated edge, the distance between client $k \in \mathcal{K}$ and edge $e \in \mathcal{E}$ can be expressed as

$$\varnothing_{k,e}(t) = \left((|x_t^{(k)}| - |x_e|)^2 + (|y_t^{(k)}| - |y_e|)^2 \right)^{\frac{1}{2}}. \tag{27}$$

From this, we know that the client-edge transmission rate in (22) can be obtained using the Shannon-Hartley formula as

$$r_{k,e}(t) = b_k(t) B \log \left(1 + \frac{(\varnothing_{k,e}(t))^{\alpha} p_k(t) |h_{k,e}(t)|^2}{b_k(t) B \sigma^2} \right), \tag{28}$$

where B is the available communication bandwidth, σ^2 is the channel noise power, $b_k(t)$ is the proportion of communication resources allocated to client $k \in \mathcal{K}$ and $h_{k,e}$ is the channel coefficient between client k and server e.

Using this multi-criteria scheme, user association can be regularly realized at the beginning of each communication round to improve the learning efficiency during multi-layer training. The

details of this multi-layer training are presented in Algorithm 1 where each user transmits its locally obtained semantic symbols to the tagged edge server for forward propagation. Subsequently, utilizing the gradients computed from the loss values at the edge, the semantic-channel decoder undergoes updates through backward propagation. These updates are followed by the exchange of aggregated gradients with other active edge servers via the cloud server, aiming to enhance the learning process at multiple edges using the FML technique. The updated gradients are then transmitted from each edge to its corresponding clients for updating their respective semantic-channel encoder. The overall time complexity of Algorithm 1 can be approximated as $O(T(|\mathcal{M}^{\mathcal{S}_e}| + |\mathcal{E}^{\mathcal{S}}|))$, where $|\mathcal{M}^{\mathcal{S}_e}|$ is the number of selected clients and $|\mathcal{E}^{\mathcal{S}}|$ is the number of participating edge servers per each round t.

B. Problem Formulation

To perform shard formation using the presented multi-criteria scheme, we describe the overall training performance in any typical learning phase l as

$$U^{(l)}(\tau_l^k) = \sum_{t=1}^{T} \frac{1}{\aleph_t} \sum_{k=1}^{K} D_k(t) \tau_{\mathcal{S}_e}^k(t), \tag{29}$$

where $\tau_{\mathcal{S}_e}^k(t) \in \{\tau_{\mathcal{S}_e}^k(1), \ldots, \tau_{\mathcal{S}_e}^k(T)\}$ captures the overall client selection decision in round $t \in \{1, \ldots, T\}$ with $\tau_{\mathcal{S}_e}^k(t) \in \{0, 1\}$. The parameter $\aleph_t$ is a temporal factor that captures the varying closeness of clients in each cluster following the multi-criteria approach and is obtained as

$$\aleph_t = \omega_1(a_{err}(t)) + \omega_2 \Bigg(\sum_{k=1}^{K} [\tau_{\mathcal{S}_e}^k(t) M_{MD}^{(t)}(k,e) + \varnothing_{k,e}(t)] + \frac{T_{ove}(l)}{T} + \frac{E_{ove}(l)}{T} \Bigg), \forall \omega_1 + \omega_2 = 1. \tag{30}$$

The weights ω_1 and ω_2 are designed to ensure that all criteria have a balanced influence on the overall performance. The smaller the parameter $\aleph_t$, the better the training performance.

In consideration of long-term performance and the quality of training, the goal is to maximize the weighted sum of any selected clients through the training performance in (29) over L learning periods and optimize the shard hibernation period, given as $H_{\mathcal{S}_e}^{(l)} \approx T_{ove}(l)$, and the overall learning accuracy, while meeting the long-term cluster formation constraints through context, intra-shard diversity, cost, and location. Formally, the problem is presented as follows:

$$\text{P1:} \max_{b_k, T, \mathcal{M}^{\mathcal{S}_e}, \tau_{\mathcal{S}_e}^k, \tau_e, tr_e, f_k} \sum_{l=1}^{L} \Bigg(\vartheta U^{(l)}(\tau_l^k) - (1-\vartheta) \Bigg(\mathcal{L}_T^{(k,e)}(\theta_T^{(k)}, \mathcal{S}_e, l) + \sum_{e=1}^{E} tr_e H_{\mathcal{S}_e}^{(l)} \Bigg) \Bigg) \tag{31}$$

$$\text{s.t. } \tau_e(t), \tau_{\mathcal{S}_e}^k(t) \in \{0,1\}, \forall k \in \mathcal{K}, \forall e \in \mathcal{E}, \tag{31a}$$

$$0 \leq tr_e \leq 1, \tag{31b}$$

$$0 \leq \vartheta \leq 1, \tag{31c}$$

$$a_{err}(l) \leq a_{err}^{th}, \forall l, \tag{31d}$$

$$\sum_{k=1}^{K} \tau_{\mathcal{S}_e}^k \left\| \frac{1}{D_k} \sum_{i=1}^{D_k} \Phi(x_i^k, l) - \frac{1}{D_e} \sum_{j=1}^{D_e} \Phi(x_j^e, l) \right\|_{\mathcal{H}} \leq c_{th}, \tag{31e}$$

$$\lim_{L \to \infty} \frac{1}{L} \sum_{l=1}^{L} T_{ove}(l) \leq T_{\max}^{\mathcal{S}_e}, \tag{31f}$$

$$\lim_{L \to \infty} \frac{1}{L} \sum_{l=1}^{L} E_{ove}(l) \leq E_{\max}^{\mathcal{S}_e}, \tag{31g}$$

$$f_k^{(l)} \leq f_{\max}^{(k)}, \tag{31h}$$

$$b_{\min} \leq b_k(t) \leq 1, \forall t, \sum_{k=1}^{K} b_k(t) = 1, \tag{31i}$$

$$T_{\min} \leq T(l) \leq T_{\max}, \tag{31j}$$

where a_{err}^{th} is the maximum threshold for the approximation error, $T_{\max}^{\mathcal{S}_e}$ represents the threshold for time cost and $E_{\max}^{\mathcal{S}_e}$ defines the threshold for energy cost in any typical shard $\mathcal{S}_e$. In addition, $f_{\max}^{(k)}$ gives the maximum bound for client computation capacity (i.e., the CPU frequency) while $T_{\min}$ and $T_{\max}$ depict the minimum and maximum bounds of communication round in any learning period, respectively. We assume the presence of a scheduler within the entire system, tasked with ensuring that all constraints are satisfied to guarantee efficient user association.

Constraint (31a) ensures that the client selection decision is binary, either 0 or 1, while Constraint (31b) guarantees that the shard training rate remains within an acceptable range. Constraint (31c) balances the trade-off between overall training performance and associated costs (i.e., the loss and hibernation period). Constraints (31d) and (31e) ensure that the approximation error and the maximum degree of discrepancy, respectively, stay within acceptable thresholds. Constraints (31f) and (31g) ensure that the long-term constraints for both time and energy costs are satisfied, while Constraint (31h) ensures that the computation capacity of any client remains within the acceptable range. Constraint (31i) guarantees that the proportion of communication resources allocated to client k adheres to the pre-defined limit, and Constraint (31j) ensures that the total number of communication rounds in each learning period does not exceed the acceptable threshold to maintain balance in $P1$.

A major challenge of directly solving $P1$ involves the existence of trade-offs between the long-term energy constraints, bandwidth allocation decisions, and learning accuracy across various learning rounds. While including more clients in the present learning period (subject to intra-shard diversity and context constraints) may enhance overall learning accuracy, it may limit the bandwidth allocated to each client, consequently increasing energy consumption for these clients and the hibernation time. Furthermore, an increased energy consumption in the current period has the potential to reduce the available energy

Algorithm 1: Multi-Layer Learning.

Input: Set of clients $\mathcal{K}$, set of edge servers $\mathcal{E}$, client participating indicator $\tau_{\mathcal{S}_e}^k$, edge server participating indicator τ_e, number of rounds T, learning rate η.
Output: Parameter vector $\theta_T^{(k)}$, global model parameter $\theta_T^{(c)}, \forall k \in \mathcal{K}, e \in \mathcal{E}, t \in \{1, \ldots, T\}$.
Initialize: $\theta_0^{(k)} = (\alpha_0^{(k)}, \beta_0^{(k,e)})$, $\mathcal{M}^{\mathcal{S}_e}$
For each round $t = 0$ to T **do**
For each $k \in \mathcal{M}^{\mathcal{S}_e}$ **do**
$s^{(k)} = \mathcal{T}_{\alpha^{(k)}}(x^{(k)}), \forall k \in \mathcal{K}$
Offload $s^{(k)}$ over wireless channel
$\bar{s^{(k)}} = \mathcal{R}_{\beta^{(k)}}(h_k s^{(k)} + n_k)$,
Recover $\bar{x^{(k)}}$ using the customized AIGC model
Compute and perturb gradients $\nabla \mathcal{L}_t^{(k,e)}(\theta_t^{(k)})$
Update parameters $\theta_{t+1}^{(k)} = \theta_t^{(k)} -$
$\eta(\omega_e \nabla_{\theta^{(k)}} \mathcal{L}_t^{(k,e)}(\theta_t^{(k)}) + \omega_c \nabla_{\theta^{(c)}} \mathcal{L}_t^{(c)}(\theta_t^{(c)}))$
End For
Edge layer aggregation: $\theta_{t+1}^{(e)} = \frac{1}{\sum_k D_k} \sum_k \tau_{\mathcal{S}_e}^k D_k \theta_{t+1}^{(k)}$
Edge $e \in \mathcal{E}$ with $\tau_e = 1$ offloads $\theta_{t+1}^{(e)}$ to cloud
Cloud layer aggregation: $\theta_{t+1}^{(c)} = \frac{\omega_e}{\sum_e D_e} \sum_e \tau_e D_e \theta_{t+1}^{(e)} + \omega_c \theta_t^{(c)}$
Broadcasts $\theta_{t+1}^{(c)}$ to all participating edge servers
Edge server synchronization: $\theta_{t+1}^{(e)} \leftarrow \theta_{t+1}^{(c)}$
Broadcasts $\theta_{t+1}^{(e)}$ to all clients $k \in \mathcal{M}^{\mathcal{S}_e}$
Synchronization at client k: $\alpha_{t+1}^{(k)} \leftarrow \theta_{t+1}^{(e)}$
End For

for meeting the future learning accuracy requirement. The need to find an effective approach to solve the long-term performance problem becomes important.

In the next section, we detail the methods adopted to solve the long-term performance problem, presented in (31).

V. Performance Optimization of the Multi-Criteria Scheme

To increase the possibility of finding a solution for the long-term performance problem, we reformulate $P1$. We note that incorporating the multi-layer FSL loss into P1 further increases its complexity. Since the multi-layer FSL loss depends on the number of communication rounds T, we may simplify $P1$ by converting the FSL loss into a constraint subject to the overall communication round $T \leq T_{\max}$. We can then reformulate the problem as

$$\text{P2:} \max_{b_k, T, \mathcal{M}^{\mathcal{S}_e}, \tau_{\mathcal{S}_e}^k, \tau_e, tr_e, f_k} \sum_{l=1}^{L} \left(\vartheta U^{(l)}(\tau_l^k) - (1-\vartheta) \left(\sum_{e=1}^{E} tr_e H_{\mathcal{S}_e}^{(l)} \right) \right)$$

$$\text{s.t. } (31a) - (31i), \tag{32}$$

$$\mathcal{L}_T^{(k,e)}(\theta_T^{(k)}, \mathcal{S}_e, l) \leq \mathcal{L}_{th}^{(e)}, \forall T(l) \leq T_{\max}, \tag{32a}$$

where $\mathcal{L}_{th}^{(e)}$ represents the maximum allowable learning loss at each learning period l. Problem $P2$ involves client selection and cluster hibernation, and also presents a significant challenge due to the absence of future information over the long-term period L. Achieving an optimal solution necessitates comprehensive offline data, encompassing hibernation periods and selected clients across every training round of each learning period l throughout the extended timeframe L. Indeed, $P2$ presents a challenging scheduling problem characterized by long-term objectives and constraints related to shard formation. Addressing this optimization issue offline poses significant challenges due to several factors including:

- The scheduling process is impacted by random events, such as client availability, which are only known at the onset of each round. Consequently, devising an offline strategy devoid of this real-time information poses a challenge in ensuring compliance with associated constraints.
- The time-coupling constraints outlined in (31f) and (31g) present considerable difficulty when attempting to solve them using offline methodologies.
- The scheduler must consider multiple criteria such as cost, diversity, and location, which can only be known after or during the actual learning process. However, decisions must be made before the commencement of each learning period, making it challenging to obtain the actual values of these criteria during decision-making.

We hence adopt a Lyapunov optimization framework to transform $P2$ into an online problem. To achieve this, we first decompose $P2$ by defining two virtual queues: a hibernation period violation queue Q_H and an energy deficit queue Q_E to describe how training hibernation period $H_{\mathcal{S}_e}^{(l)}$ and overall energy cost $E_{ove}(l)$ at each learning period l may deviate from the long-term budget $T_{\max}^{\mathcal{S}_e}$ and $E_{\max}^{\mathcal{S}_e}$, respectively. The dynamic evolutions of these queues are obtained as

$$\begin{aligned} Q_H(l+1) &= \max[Q_H(l) + T_{ove}(l) - \tfrac{1}{L} T_{\max}^{\mathcal{S}_e}, 0], \\ Q_E(l+1) &= \max[Q_E(l) + E_{ove}(l) - \tfrac{1}{L} E_{\max}^{\mathcal{S}_e}, 0]. \end{aligned} \tag{33}$$

Let $V \geq 0$ denote a penalty factor set for the purpose of balancing the tradeoff between maximizing the objective and satisfying the shard formation constraints. In every learning period l, the aim is to solve the per-period problem given as

$$\text{P3:} \max_{b_k, T, \mathcal{M}^{\mathcal{S}_e}, \tau_{\mathcal{S}_e}^k, \tau_e, tr_e, f_k} V. \left(\vartheta U^{(l)}(\tau_l^k) - (1-\vartheta) \left(\sum_{e=1}^{E} tr_e H_{\mathcal{S}_e}^{(l)} \right) \right) - (Q_H(l) T_{ove}(l) + Q_E(l) E_{ove}(l))$$

$$\text{s.t. } (31a) - (31e), (31h), (31i), (32a). \tag{34}$$

However, $P3$ is still difficult to solve since the criteria for user association are unknown before the commencement of the actual learning process at each learning period. Hence, following the concept of the contextual combinational multi-arm bandit-based

model [35], we develop a context-enabled multi-criteria scheme (CeM) with volatile arms and submodular rewards [36] for estimating the key shard formation criteria based on the historical information and performance of each client $k \in \mathcal{K}$ and edge server $e \in \mathcal{E}$, assumed to be available to the scheduler.

A. Context-Enabled Multi-Criteria Scheme

CeM relies on two variants of classical multi-arm bandit (MAB) strategies: combinatorial and contextual MABs. Within CeM, the action (arm) space operates on a combinatorial level. This means that instead of choosing a single action, the agent must select a combination of actions (otherwise called the composite action). Consequently, each combination of actions may result in a different reward. The primary objective for such an agent is to learn the optimal combination of actions that will maximize cumulative rewards. Additionally, these rewards tied to specific action combinations may depend on additional contextual information available to the agent at each decision point. Given that each action in CeM corresponds to a client, the algorithm aims to find the best combination of clients that maximizes cumulative rewards.

We consider a sequential decision-making process over T communication rounds. For each learning period l, let the tuple $(\mathcal{K}, \mathcal{S}_t, \{\vartheta_k^{(*)}\}_{k\in\mathcal{K}}, \{z_t^k\}_{k\in\mathcal{K}}, \{\epsilon_t^k\}_{k\in\mathcal{K}})$ represent the CeM problem where $\mathcal{K}$ is the set of arms available in any communication round t, $S_t \subseteq 2^{[K]}$ represents all possible subsets or combination of arms in round t, $\vartheta_k^{(*)}$ is the unknown but stationary coefficient vector, z_t^k represents the known but dynamic historical feature vectors and ϵ_t^k is the zero-mean random variable that presents the noise. Note that the sets of arms $\mathcal{K}$ are time-varying, meaning $\mathcal{K}(t)$, $0 < t \leq T$, may vary in different rounds. For each arm $k \in \mathcal{K}$, the resulting performance can be captured as a function of the contextual vector $Z_t = \{z_t^k\}_{k\in\mathcal{K}}$ while the quality (i.e., the performance or reward) of arm k, drawn from an unknown distribution, can be expressed as $r(z_t^k)$ with expected value $\mu(z_t^k) = \mathbb{E}[r(z_t^k)]$. Given that $r^t = \{r(z_t^k)\}_{k\in\mathcal{K}}$ and $\mu^t = \{\mu(z_t^k)\}_{k\in\mathcal{K}}$ are the qualities of available arms in round t and their expected value, respectively, the objective of the developed CeM is to select a super arm $\mathcal{M}_t^{\mathcal{S}_e} \subseteq \mathcal{K}$ that maximize the total reward.

Generally, the number of arms in any super arm $\mathcal{M}_t^{\mathcal{S}_e}$ is bounded by the maximum number $\mathcal{M}_{\max} \leq K$ such that $\mathcal{M}_t^{\mathcal{S}_e} \leq \mathcal{M}_{\max}, \forall t$, while $\mathcal{M}_{\max}$ is a constant across rounds $0 < t \leq T$. The submodular reward function $u : \mathcal{S}_t \to \mathbb{R}^+$ can be defined as the observed reward of any super arm. Let $u(r^t, \mathcal{M}_t^{\mathcal{S}_e})$ represent the reward of selecting super arm $\mathcal{M}_t^{\mathcal{S}_e}$. Its value is jointly determined through the qualities of each arm $\{r(z_t^k)\}_{k\in\mathcal{K}}$ as well as the relationships between arms that create sub-modularity. If we let χ represent the problem $P3$, then the reward function can be modeled as $\chi(r^t, \mathcal{M}_t^{\mathcal{S}_e})$ such that for any $\mathcal{M}_t^{\mathcal{S}_e} \subseteq \mathcal{K}$,

$$\max_{\mathcal{M}_1^{\mathcal{S}_e},\ldots,\mathcal{M}_T^{\mathcal{S}_e}} \sum_{t=1}^{T} \mathbb{E}[\chi(r^t, \mathcal{M}_t^{\mathcal{S}_e})],$$
$$\text{s.t.} \quad |\mathcal{M}_t^{\mathcal{S}_e}| \leq \mathcal{M}_{\max}, \mathcal{M}_t^{\mathcal{S}_e} \subseteq \mathcal{K}, \forall t. \tag{35}$$

The problem in (35) can also be reformulated by decoupling it into T subproblems, given as

$$\max_{\mathcal{M}_t^{\mathcal{S}_e}\in\mathcal{K}} \mathbb{E}[\chi(r^t, \mathcal{M}_t^{\mathcal{S}_e})],$$
$$\text{s.t.} \quad |\mathcal{M}_t^{\mathcal{S}_e}| \leq \mathcal{M}_{\max}, \mathcal{M}_t^{\mathcal{S}_e} \subseteq \mathcal{K}. \tag{36}$$

By relying on historical performance information, the scheduler is aware of the historical qualities of previously selected arms and can use this to estimate the expected quality of its present selection. Thus, we can slightly modify $\mathbb{E}[\chi(r^t, \mathcal{M}_t^{\mathcal{S}_e})]$ as $\chi(\mathbb{E}[r^t], \mathcal{M}_t^{\mathcal{S}_e}) = \chi(r^t, \mathcal{M}_t^{\mathcal{S}_e})$. With this, the scheduler aims to select a super arm $\mathcal{M}_t^{\mathcal{S}_e*}(Z_t)$ that satisfies

$$\mathcal{M}_t^{\mathcal{S}_e*}(Z_t) = \arg \max_{\mathcal{M}^{\mathcal{S}_e}\subseteq\mathcal{K}; |\mathcal{M}^{\mathcal{S}_e}|\leq\mathcal{M}_{\max}} \chi(\mu^t, \mathcal{M}_t^{\mathcal{S}_e}). \tag{37}$$

Nevertheless, maximizing a submodular function subject to a cardinality constraint in (37) poses an NP-hard problem. Fortunately, the greedy algorithm provides a polynomial-time solution and guarantees achieving at least $(1-q)$ of the optimum value.

Definition 1: Define $\mathcal{M}^{\mathcal{S}_e}$ and $\mathcal{M}_t^{\mathcal{S}_e*}$ as the super arm selected by the greedy algorithm [37], [38] and the optimal super arm, respectively. At least $(1-q)$ optimum means

$$\chi(\mu^t, \mathcal{M}_t^{\mathcal{S}_e}) \geq (1-q)\chi(\mu^t, \mathcal{M}_t^{\mathcal{S}_e*}). \tag{38}$$

Definition 2: The α-regret of any algorithm up to round T is defined as

$$\alpha(T) = (1-q)\sum_{t=1}^{T} \mathbb{E}[\chi(r^t, \mathcal{M}_t^{\mathcal{S}_e*})] - \sum_{t=1}^{T} \mathbb{E}[\chi(r^t, \mathcal{M}_t^{\mathcal{S}_e})]. \tag{39}$$

The estimation of the expected rewards follows the contextual bandit with similarity information method [36], and is presented in Algorithm 2, where arms are categorized into distinct groups based on their contextual information. Subsequently, the algorithm learns the expected quality for each group of arms under the assumption that arms sharing similar contextual information exhibit similar qualities. For every context z_t^k, the developed CeM identifies a potential arm group $p_k^t \in \mathcal{P}_T$ requiring estimation of its expected quality, satisfying the condition $z_t^k \in p_k^t$. The algorithm then examines whether there are any $p \in p^t = \{p_k^t\}_{k\in\mathcal{K}}$ that have not been explored adequately, while also collecting the arms falling within the under-explored groups $\mathcal{K}^u$. Depending on $\mathcal{K}^u$, the CeM transitions into either the exploration or exploitation phase. After selecting the set of arms, CeM evaluates the qualities exhibited by the chosen arms, subsequently updating both the estimated quality and the count for each group in p_t. The overall complexity of Algorithm 2 per iteration relies on the complexity of exploration and exploitation, which depends on the number of observed contexts in the current iteration $|\mathcal{K}|$, $|\mathcal{K}^u|$ and $\mathcal{M}_{\max}$. This can be expressed as $O(|\mathcal{K}| \cdot |\mathcal{P}_T| + \mathcal{M}_{\max} \cdot (|\mathcal{K}| + |\mathcal{K}^u|))$. For $T \ll \infty$ iterations, the total time complexity is $O(T(\cdot|\mathcal{K}| \cdot |\mathcal{P}_T| + \mathcal{M}_{\max} \cdot (|\mathcal{K}| + |\mathcal{K}^u|)))$. As $T \to \infty$, the complexity becomes increasingly unmanageable. However, in practical systems, T is expected to remain finite.

Algorithm 2: Context-Enabled Multi-Criteria Scheme.

Input: T, contextual information space, exploration parameter $E_p^{(t)}$.
Initialization: Context information space partition $\mathcal{P}_T$; counter $c_0^p = 0$; estimated context-specific qualities $\hat{r}(p) = 0, \forall p \in \mathcal{P}_T$.
For $t = 1, \ldots, T$ **do**
  Observe $\mathcal{K}$ with its corresponding contexts $Z_t = \{z_t^k\}_{k\in\mathcal{K}}$
  Find group $p_t = \{p_k^t\}_{k\in\mathcal{K}} \forall z_t^k \in p_k^t; p_k^t \in \mathcal{P}_T; k \in \mathcal{K}$
  Check for under-explored groups
  $\mathcal{P}_t^u \triangleq \{p \in \mathcal{P}_T | \exists\, k \in \mathcal{K}, z_t^k \in p, c_t^p \leq E_p^{(t)}\}$
  and the associated arm set $\mathcal{K}^u \triangleq \{k \in \mathcal{K} | p_k^t \in \mathcal{P}_T^u\}$.
  Exploration:
  If $\mathcal{P}_t^u \neq \{\}$:
    If $|\mathcal{K}^u| \geq \mathcal{M}_{\max}$:
      Create $\mathcal{M}_t^{\mathcal{S}_e}$ by randomly selecting $\mathcal{M}_{\max}$ arms from $\mathcal{K}^u$
    Else: Create $\mathcal{M}_t^{\mathcal{S}_e}$ by selecting $|\mathcal{K}^u|$ arms in $\mathcal{K}^u$ and other $\mathcal{M}_{\max} - |\mathcal{K}^u|$ arms: k_i
      $\forall k_i = \arg\max_{k_i \in \mathcal{K}\setminus\{\mathcal{K}^u \bigcup \mathcal{M}_{i-1}^{\mathcal{S}_e}\}}$
      $\Delta(\hat{r^t}, \{k_i\} | \mathcal{K}^u \bigcup \mathcal{M}_{i-1}^{\mathcal{S}_e}), i = 1, \ldots, \mathcal{M}_{\max} - |\mathcal{K}^u|$
  Exploration:
  Else: Create $\mathcal{M}_t^{\mathcal{S}_e}$ by selecting $\mathcal{M}_{\max}$ arms: k_i
    $\forall k_i = \arg\max_{k_i \in \mathcal{K}\setminus\mathcal{M}_{i-1}^{\mathcal{S}_e}} \Delta(\hat{r^t}, \{k_i\} | \mathcal{M}_{i-1}^{\mathcal{S}_e})$,
    $i = 1, \ldots, \mathcal{M}_{\max}$
  For each $k \in \mathcal{M}_t^{\mathcal{S}_e}$ **do**:
  Observe r_k
  Update $\hat{r}(p_k^t) \leftarrow \frac{\hat{r}(p_k^t) c_t^{p_k} + r_k}{c_t^{p_k}+1}$
  Update $c_t^{p_k} \leftarrow c_t^{p_k} + 1$

For each communication round $t \in \{1, \ldots, T\}$, we can transform $P3$ using the presented MAB-based solution. To do this, we first observe that the performance of the user association framework as well as its associated hibernation time is associated with the computation parameter of each client f_k, the running state parameters $b_k, \tau_{\mathcal{S}_e}^k$ and τ_e, and the multi-criteria features: diversity, location and context. Given $Z_t \triangleq \{z_t^1, \ldots, z_t^K\} \subseteq \mathbb{R}^d$, we thus define the feature parameter $z_t^k = [f_k(t), D_k(t), b_k(t), \tau_{\mathcal{S}_e}^k(t), \tau_e(t), a_{err}(t), \varnothing_{k,e}(t), M_{MD}^t(k,e)]$ and $\vartheta_k^{(*)} = [\hat{\mathcal{M}^{\mathcal{S}_e}}, \hat{T_{ove}}(l), \hat{E_{ove}}(l)]$. If the parameters defined in z_t^k are estimated based on the dynamic historical data, we can obtain predicted values for the unknown parameters. Clearly, as the available historical information accumulates, the scheduling approach is expected to achieve improved accuracy. With this, $P3$ can be transformed to

$$\text{P4:} \max_{b_k, T, \mathcal{M}^{\mathcal{S}_e}, \tau_{\mathcal{S}_e}^k, \tau_e, tr_e, f_k} V.\Bigg(\vartheta U^{(l)}(\tau_l^k) - (1-\vartheta)\left(\sum_{e=1}^{E} tr_e H_{\mathcal{S}_e}^{(l)}\right)\Bigg) - (Q_H(l)\hat{T_{ove}}(l) + Q_E(l)\hat{E_{ove}}(l))$$
$$\text{s.t.} \quad (31b) - (31e), (31h), (31i), (32a). \tag{40}$$

Algorithm 3: Context-Centric User Selection Algorithm (CUSA).

Input: $Q_H(0) = 0, Q_E(0) = 0$
Initialize: $M_{MD}^k = M_{MD}(k,e) \in [0,1]$
Selection priority: Sort $k \in \mathcal{K}$ in ascending order based on M_{MD}^k
$\mathcal{M}_0^{\mathcal{S}_e} = \{k : M_{MD} = 0\}, \mathcal{M}^{\mathcal{S}_e} = \mathcal{M}_0^{\mathcal{S}_e}, \mathcal{M} = \{\mathcal{M}_0^{\mathcal{S}_e}\}$
For $k = |\mathcal{M}_0^{\mathcal{S}_e}| + 1$ to K **do**
  Update $\mathcal{M}^{\mathcal{S}_e} \bigcup \{k\}$
  Solve $P5$ and obtain
  $b_k^*(\mathcal{M}^{\mathcal{S}_e}), T^*(\mathcal{M}^{\mathcal{S}_e}), tr_e^*(\mathcal{M}^{\mathcal{S}_e}), f_k^*(\mathcal{M}^{\mathcal{S}_e})$
  If $(\vartheta U^{(l)}(\tau_l^k) - (1-\vartheta)(\sum_{e=1}^{E} tr_e H_{\mathcal{S}_e}^{(l)})) < 0$ **then**
    Stop Iteration
  Else
    Update $\mathcal{M} = \mathcal{M} \cup \{\mathcal{M}^{\mathcal{S}_e}\}$
  End If
End For
Obtain $\mathcal{M}^{\mathcal{S}_e *} = \arg\max_{\mathcal{M}^{\mathcal{S}_e} \in \mathcal{M}} W^*(\mathcal{M})$
Return $b_k^*(\mathcal{M}^{\mathcal{S}_e *}), T^*(\mathcal{M}^{\mathcal{S}_e *}), tr_e^*(\mathcal{M}^{\mathcal{S}_e *}), f_k^*(\mathcal{M}^{\mathcal{S}_e *})$

Next, we provide a solution to the problem $P4$.

B. Context-Centric User Selection Algorithm

The problem $P4$ can be solved as a mixed-integer problem although the solution is still very complicated. To solve $P4$, we present CUSA, a context-centric user selection algorithm as shown in Algorithm 3, where clients $k \in \mathcal{K}$ are incrementally associated with an edge $e \in \mathcal{E}$, based on the context parameter $M_{MD}(k,e)$, to form the corresponding shard. For simplicity, we consider that the shard members remain fixed during every learning phase. We refer to $M_{MD}(k,e)$ as the selection priority since $M_{MD}(k,e)$ represents the level of significance of client data k to edge e. Let the initial super arm be $\mathcal{M}_0^{\mathcal{S}_e}$ containing clients with $M_{MD}(k,e) = 0$. Clients with $M_{MD}(k,e) > 0$ are sorted in ascending order and are added to $\mathcal{M}^{\mathcal{S}_e}$ following the sorted order. Given that $A_k^*(B)$ is the optimal value of A for a given super arm B, for each selected super arm, $P4$ is formulated as

$$\text{P5:} \max_{b_k, T, tr_e, f_k} V.\Bigg(\vartheta \frac{T}{\aleph_t} \sum_{k \in \mathcal{M}^{\mathcal{S}_e}} D_k - (1-\vartheta)\left(\sum_{e=1}^{E} tr_e H_{\mathcal{S}_e}^{(l)}\right)\Bigg) - (Q_H(l)\hat{T_{ove}}(l) + Q_E(l)\hat{E_{ove}}(l))$$
$$\text{s.t.} \quad (31b) - (31e), (31h), (31i), (32a). \tag{41}$$

As shown in Algorithm 3, CUSA outputs optimal values for b_k, T, tr_e and f_k for $\mathcal{M}^{\mathcal{S}_e *}$ after solving P5. Generally, sorting $k \in \mathcal{K}$ based on M_{MD}^k has a time complexity of $O(KlogK)$ while the total complexity for the iterative update is $K - |\mathcal{M}_0^{\mathcal{S}_e}|$. In the worst case, where $\mathcal{M}$ contains all possible subsets 2^K, the overall complexity of the CUSA algorithm can be approximated as $O(KlogK) + (K - |\mathcal{M}_0^{\mathcal{S}_e}|) + O(2^K)$.

VI. Performance Evaluation

In this section, we evaluate the performance of the proposed multi-criteria and multi-shard user selection framework in the MAH-enabled environment. As a use case and proof of concept, we designed and implemented a multi-layer user selection architecture and incorporated the proposed CeM-CUSA solution. To establish an MAH-enabled environment, we initiated the integration of AIGC into the virtual environment by encoding input data into tensors and aligning such with the requirements of the generation model. Subsequently, a fine-tuned GPT-2 model was employed to generate content that accurately reflects the states of PT in the physical environment. To ensure the desired balance of randomness, creativity, and coherence in the generated outputs, generative AI hyperparameters such as temperature, top-k, and top-p were carefully adjusted. The resulting tensors were then decoded into high-quality predictive text, facilitating the maintenance and continuous evolution of VTs. The proposed multi-layer user selection scheme analyzes the collaborative effects of multi-source and multi-destination users with diverse datasets and requirements on the overall performance of MAH. To demonstrate the performance improvements of the proposed user selection approach, we also implemented three existing solutions as benchmarks for comparison:

- *Traditional Approach (Random Selection Strategy):* The random selection strategy is a traditional communication protocol for a multi-source, single-destination mechanism. Clients (i.e., multiple users) are placed into different shards following a random selection strategy to meet the destination (receiver) requirements.
- *Unilateral Selection Strategy:* This approach creates shards based solely on the receiver's needs, focusing on optimizing information parsing and understanding efficiency at the receiving side.
- *Greedy Algorithm-based Strategy:* This framework uses a greedy algorithm [37], centering on marginal gains in the sharding process to optimize the selection of information elements. This enhances the synergistic effect of information within the shard.

A. Implementation Settings

We implemented the FSL-enabled MAH following the approach discussed in [6]. To incorporate the proposed user selection strategy, we leveraged a publicly available dementia patient health dataset[2] containing 10,000 entries with eight different features. The dataset was divided into two parts: half for model training and the other half as client samples. For simplifying the simulation, we deployed three edge servers and one cloud server. Using the cosine similarity method, clients were initially divided into six pre-groups following Algorithm 2. The most compatible client group for each edge server was then selected using $P4$ as the reward function. During the execution phase, the initially selected groups were sorted by MMD values, and client distribution was continuously adjusted using $P5$, as the reward function, to optimize model performance while meeting the dynamic changes requirement in the environment and client needs. This sharding process was repeated at the end of each learning cycle until 10 cycles were completed.

[2][Online]. Available: https://www.kaggle.com/datasets/kaggler2412/dementia-patient-health-and-prescriptions-dataset

TABLE II
Simulation Parameters

Parameter	Value	Parameter	Value
B	1 MHz	$\phi_{k,e}$	$[1, 100]$ km
T	20	L	$[1, 10]$
η	0.01	D_k	$[2, 10]$
D_e	$[10, 20]$	f_k	$[2, 3.5]$ GHz
f_e	10 GHz	f_c	20 GHz
f_r^k	1 cycles	f_r^e	2 cycles
f_r^c	4 cycles	a_{err}^{up}	3
c_{th}	1	ω_1, ω_2	0.5
$a_e^{(k)}, a_j^{(e)}$	$[0, 1]$	V	0.5
θ	0.5	tr_e	$[0, 1]$
ρ_k, ρ_e	10^{-27} F	$\Lambda_{Se}^k, \Lambda_e$	$\{0, 1\}$

In the initial phase of the model development, we implemented a variational autoencoder (VAE) to facilitate data compilation and augmentation. The VAE consists of an encoder and a decoder. The encoder, using fully connected layers with ReLU activation functions, maps input data to the latent space's mean and log variance. The decoder then reconstructs the original data from samples drawn from the latent space using a sampling layer and fully connected layers with linear activation functions. Following this, we integrated the BERT model [39] as the primary tool for semantic communication, utilizing its pre-trained features for a deep semantic understanding of texts. The feature vectors generated by the BERT model were fed into a hybrid neural network that includes densely connected layers, batch normalization, and ReLU activation functions, with residual connections to prevent information loss in deeper layers. Finally, the processed data served as inputs to the GPT-2 model, which was fine-tuned to enhance the diversity and quality of text generation through adjustments in temperature settings and top-k filtering. This setup allowed GPT-2 to produce text that is not only creative and relevant but also of high textual quality.

For the evaluation, we integrated an online learning mechanism into the training process to maintain objectivity in assessment. The online learning mechanism allows the model to continuously adapt to new data and update its parameters in real-time, enhancing the model's adaptability and long-term performance in dynamic environments. For illustration, the values of various parameters were selected following [6], [30] and are presented in Table II for completeness.

B. Evaluation Results

Following the presented implementation settings, we first focus on investigating the performance of the proposed solutions using accuracy and loss as performance metrics. To demonstrate performance improvement during testing, we incorporate the normalized performance metric (NPM) – a method that ensures fair comparison, eliminates scale differences, and

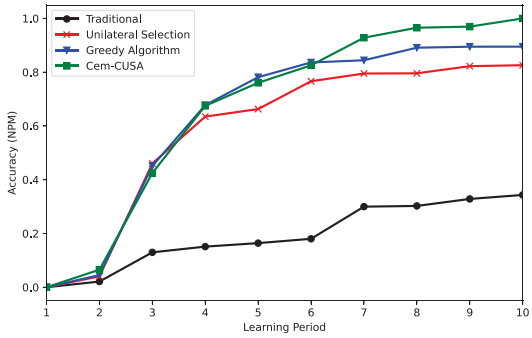


Fig. 3. Accuracy of the considered selection strategies.

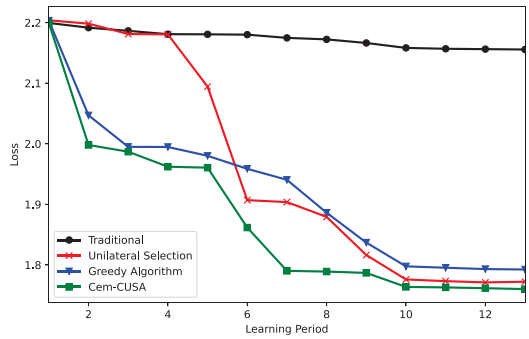


Fig. 4. Performance in terms of loss.

provides meaningful, consistent insights across diverse datasets and models. NPM maps the mean absolute error (MAE) to a consistent range, where 1 indicates nearly perfect predictions and 0 denotes extremely poor performance. By inverting the normalized MAE, the NPM effectively measures performance, with higher NPM values indicating performance closer to the ideal state. This method simplifies the assessment of the model's effectiveness in reducing prediction errors, making high NPM values intuitively represent superior predictive performance. As presented in Fig. 3, all user selection strategies demonstrated significant accuracy improvements with learning period L. Although the Greedy algorithm achieved faster convergence, CeM-CUSA achieved a superior accuracy level, highlighting its exceptional performance in continuous learning and adapting to new updates in the physical environment. Similarly, the proposed CeM-CUSA strategy experienced the smallest loss, as shown in Fig. 4, consistently demonstrating its performance improvements over other schemes.

Next, we assessed the sharding (i.e., clustering) efficiency of the considered user selection strategies using average MMD, approximation error, and proximity, as presented in Fig. 5. MMD measures the disparity in the sample sets between each client shard and the corresponding server, helping to ascertain shard balance. The approximation error compares the discrepancy between the data within the selected client shard and the overall client data, determining whether the group data effectively represents the entire client dataset. Average proximity measures the similarity of client data within groups, where a high degree of similarity typically indicates greater efficiency in the sharding process and potential for effective training processes. The results

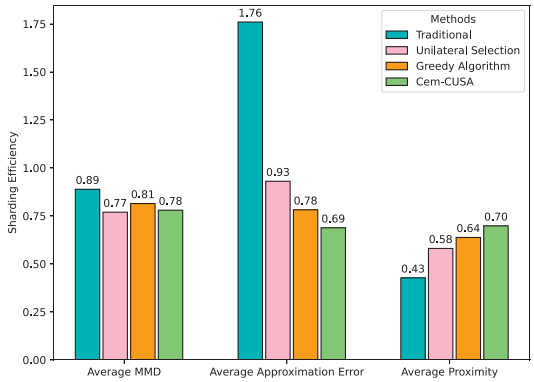


Fig. 5. User selection algorithms efficiency.

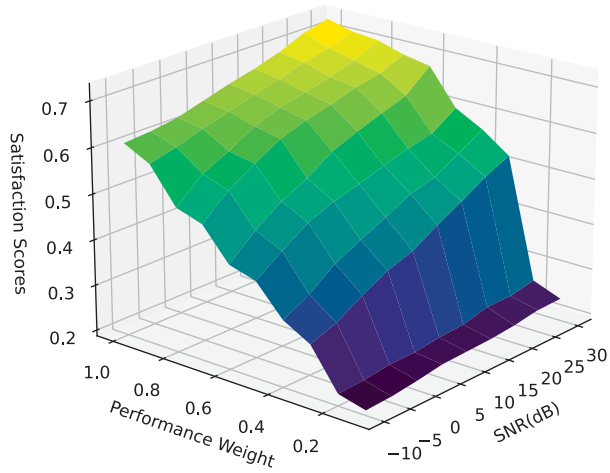


Fig. 6. Performance variations of CeM-CUSA.

in Fig. 5 demonstrate that the proposed solution achieved the best performance levels across the selected parameters for sharding efficiency measurement. The CeM-CUSA strategy proves its effectiveness in ensuring the balance of sample sets between clients and servers, as well as maintaining high similarity within client groups.

Furthermore, we investigate the performance variations of the CeM-CUSA strategy under different configurations, using satisfaction scores – a metric that captures the overall performance from users' perspective – as the primary evaluation criterion. Fig. 6 illustrates the impact of performance weight and signal-to-noise ratio (SNR) on satisfaction scores. The results show that as SNR and performance weight increase, the satisfaction scores also exhibit an upward trend. It is worth noting that, even under worst-case conditions with an SNR of -10 dB, if clients prioritize improved performance, the satisfaction scores can still reach approximately 0.6. This demonstrates that with the CeM-CUSA strategy, appropriately adjusting performance priorities can effectively ensure client satisfaction even in poor network conditions.

To emphasize the superiority of the CeM-CUSA strategy under various network conditions, we carried out a comprehensive evaluation using bandwidth and CPU frequency, as shown in Figs. 7 and 8. While overall performance increases

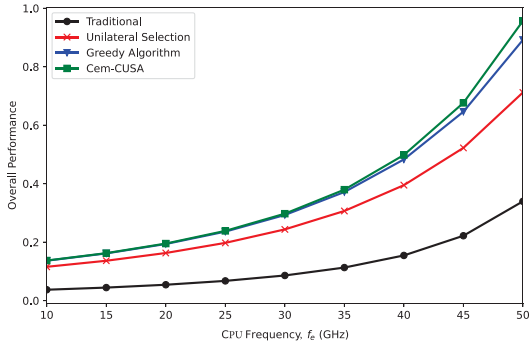


Fig. 7. Effects of f_e on the overall performance.

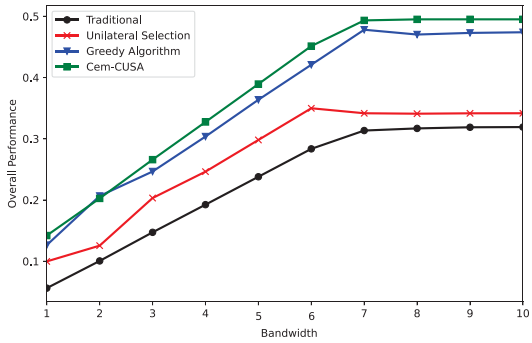


Fig. 8. Effects of B on the overall performance.

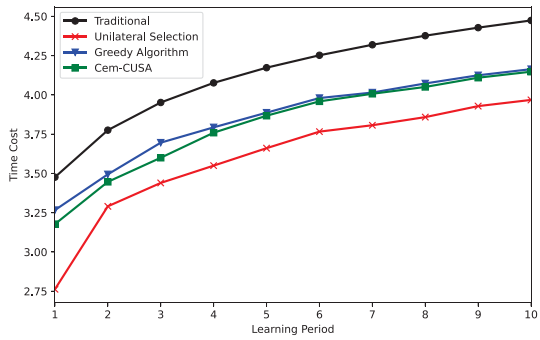


Fig. 9. The temporal cost associated with L.

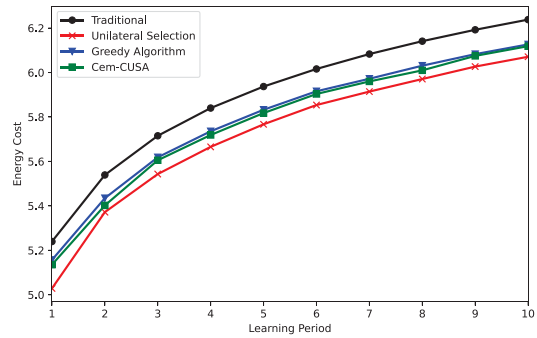


Fig. 10. Energy cost versus L.

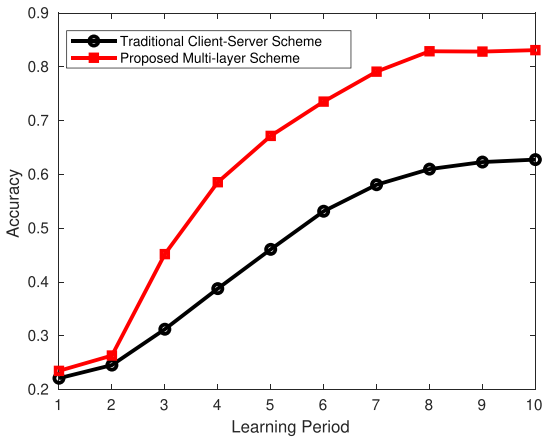


Fig. 11. Advantages of the introduced multilayer scheme.

with CPU frequency, as shown in Fig. 7, the CeM-CUSA strategy consistently outperforms other solutions as CPU frequency rises. This performance advantage is attributed to the two-stage sharding optimization mechanism of the CeM-CUSA strategy, which enhances data transmission and processing efficiency, particularly during high-frequency operations. Although the Greedy Algorithm and the unilateral selection strategy also show performance improvements, the performance gap between these methods and the CeM-CUSA strategy widens at higher frequencies. This highlights the high adaptability and superior efficiency of the CeM-CUSA strategy in handling high-load communication tasks. Similar observations are noted in Fig. 8, where increasing bandwidth leads to improved overall performance of the CeM-CUSA strategy. Performance convergence is observed as bandwidth approaches 7 MHz, further demonstrating the strategy's effectiveness.

Additionally, we analyze the performance of the considered selection strategies using time and energy costs. As demonstrated in Fig. 9, the proposed CeM-CUSA strategy achieves a time cost similar to the Greedy Algorithm-based strategy, with marginal outperformance. The unilateral selection strategy exhibits a lower time cost because it focuses solely on server demands, thereby selecting client groups that are closer in proximity and have higher transmission rates and CPU frequencies. A similar observation is made in Fig. 10, where overall energy cost is investigated. As expected, the unilateral selection strategy shows remarkable performance in terms of time and energy efficiencies, as the number of clients per shard is relatively small. This reduction in client numbers limits the diversity and quantity of data available during model training, which is a noticeable disadvantage for data-driven learning models. This is more evident when compared with the CeM-CUSA and Greedy Algorithm-based strategies presented in Figs. 3 and 4.

Although the unilateral selection strategy offers distinct advantages in terms of time and energy efficiencies, its effectiveness in achieving optimal model performance and generalization is constrained. This trade-off underscores the need for better selection strategies that effectively consider both the efficacy of model training and efficiency to establish a performance balance in MAH.

1) Benefits of the Multi-Layer Framework: Next, we evaluate the advantages of the proposed multi-layer framework and compare its performance with the traditional client-server scheme as presented in Fig. 11. As expected, the multi-layer framework

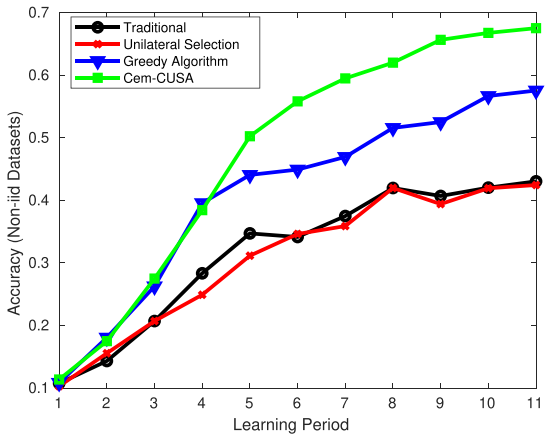


Fig. 12. Accuracy of the schemes under non-iid scenario.

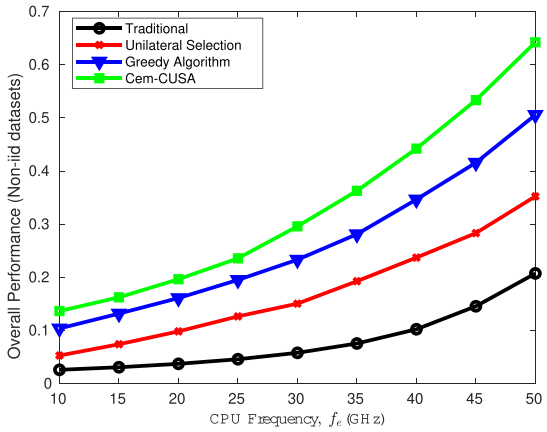


Fig. 13. Overall performance under non-iid scenario.

outperforms the existing approach in terms of accuracy when implemented in MAH. This improvement is attributed to the limited data distribution in traditional client-server schemes and the lack of learning knowledge transfer across multiple shards, a key advantage of the introduced multi-layer framework. The knowledge transfer feature in the multi-layer framework, facilitated during edge-cloud training, enhances model accuracy in MAH by enabling each shard to learn from the experiences of its neighboring shards, improving performance in the process.

2) Evaluation Under non-iid Datasets: Finally, we evaluate the performance of the proposed user selection scheme under non-independent and identically distributed (non-iid) dataset conditions. To create a non-iid dataset, we introduce data heterogeneity by randomly assigning users partial subsets of input features (feature distribution skew) and varying the number of samples allocated to each user (quantity skew).

Using the newly generated non-iid datasets, we rerun the simulation and demonstrate the obtained results for accuracy and overall performance in Figs. 12 and 13, respectively. While the system experiences performance degradation compared to scenarios with iid datasets, Fig. 12 demonstrates that the proposed solution outperforms existing methods in terms of accuracy, measured as a normalized performance metric. Similarly, Fig. 13 illustrates a superior overall performance of the MAH. These findings highlight that the proposed user selection framework is an effective solution for MAH and other related AIGC-enabled systems even under non-iid scenarios.

VII. Conclusion

In this paper, we have addressed the critical challenges associated with MAH. Our proposed multi-criteria and multi-shard framework, as well as the innovative CeM-CUSA strategy, provide effective solutions to ensure ultra-reliable, secure, and privacy-preserving connectivity between PTs and VTs, without compromising accuracy and overall cost requirements. By incorporating batch learning for semantic-channel encoders and decoders, our framework meets both periodic and on-demand training needs, significantly enhancing synchronization between PTs and VTs.

Through comprehensive evaluations, we have demonstrated the superior performance of the CeM-CUSA strategy under various network conditions and configurations. Our results have shown that CeM-CUSA outperformed traditional methods like the Greedy Algorithm and unilateral selection strategy in terms of accuracy, loss, and sharding efficiency. The two-stage sharding optimization mechanism of CeM-CUSA effectively balanced sample sets, maintained high similarity within client groups, and enhanced data transmission and processing efficiency. CeM-CUSA has also proved to be adaptable and robust under different SNR and bandwidth conditions, achieving substantial client satisfaction even in challenging scenarios. Additionally, CeM-CUSA balanced training efficacy and efficiency, ensuring optimal model performance and generalization. These advancements offer a robust solution for MAH, enabling seamless and accurate PT-VT synchronization.

While the proposed solution is effective, it may experience performance degradation as training time increases. In the future, we will expand this framework to obtain solutions that can maintain moderate computation complexity even as training time increases.

References

[1] S. D. Okegbile, J. Cai, C. Yi, and D. Niyato, "Human digital twin for personalized healthcare: Vision, architecture and future directions," *IEEE Netw.*, vol. 37, no. 2, pp. 262–269, Mar./Apr. 2023.

[2] J. Chen, C. Yi, S. D. Okegbile, J. Cai, and X. S. Shen, "Networking architecture and key supporting technologies for human digital twin in personalized healthcare: A comprehensive survey," *IEEE Commun. Surveys Tut.*, vol. 26, no. 1, pp. 706–746, First Quarter, 2024.

[3] C. Zhou, J. Gao, M. Li, N. Cheng, X. Shen, and W. Zhuang, "Digital-twin-Based 3-D map management for edge-assisted device pose tracking in mobile AR," *IEEE Internet of Things J.*, vol. 11, no. 10, pp. 17812–17826, May 2024.

[4] S. Hu, M. Li, J. Gao, C. Zhou, and X. Shen, "Adaptive device-edge collaboration on DNN inference in AIoT: A digital-twin-assisted approach," *IEEE Internet of Things J.*, vol. 11, no. 7, pp. 12893–12908, Apr. 2024.

[5] X. Huang, W. Wu, S. Hu, M. Li, C. Zhou, and X. Shen, "Digital twin based user-centric resource management for multicast short video streaming," *IEEE IEEE J. Sel. Topics Signal Process.*, vol. 18, no. 1, pp. 50–65, Jan. 2024.

[6] S. D. Okegbile, O. Talabi, H. Gao, J. Cai, and C. Yi, "FLeS: A federated learning-enhanced semantic communication framework for mobile AIGC-driven human digital twins," *IEEE Netw.*, early access, Jan. 06, 2025, doi: 10.1109/MNET.2025.3526556.

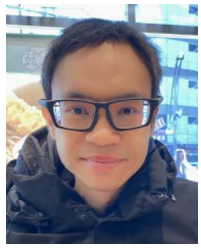
[7] H. Wei, W. Ni, W. Xu, F. Wang, D. Niyato, and P. Zhang, "Federated semantic learning driven by information bottleneck for task-oriented communications," *IEEE Commun. Lett.*, vol. 24, no. 10, pp. 2652–2656, Oct. 2023.
[8] O. Wehbi et al., "FedMint: Intelligent bilateral client selection in federated learning with newcomer IoT devices," *IEEE Internet of Things J.*, vol. 10, no. 23, pp. 20884–20898, Dec. 2023.
[9] L. Fu, H. Zhang, G. Gao, M. Zhang, and X. Liu, "Client selection in federated learning: Principles, challenges, and opportunities," *IEEE Internet of Things J.*, vol. 10, no. 24, pp. 1811–21819, Dec. 2023.
[10] J. Xu and H. Wang, "Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1188–1200, Feb. 2021.
[11] L. Yu, R. Albelaihi, X. Sun, N. Ansari, and M. Devetsikiotis, "Jointly optimizing client selection and resource management in wireless federated learning for Internet of Things," *IEEE Internet of Things J.*, vol. 9, no. 6, pp. 4385–4395, Mar. 2022.
[12] X. Huang et al., "Federated learning-empowered AI-generated content in wireless networks," *IEEE Netw.*, vol. 38, no. 5, pp. 304–313, Sep. 2024.
[13] Y. Lin et al., "Decentralized unlearning for trustworthy AI-Generated content (AIGC) services," *IEEE Netw.*, early access, Aug. 06, 2024, doi: 10.1109/MNET.2024.3439411.
[14] Q. Zhang et al., "Exploring edge-driven collaborative fine-tuning towards customized AIGC services," *IEEE Netw.*, early access, Aug. 09, 2024, doi: 10.1109/MNET.2024.3441040.
[15] D. B. Ami, K. Cohen, and Q. Zhao, "Client selection for generalization in accelerated federated learning: A multi-armed bandit approach," 2023, *arXiv:2303.10373*.
[16] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. IEEE Int. Conf. Commun.*, Shanghai, 2019, pp. 1–7.
[17] S. Abdulrahman, H. Tout, A. Mourad, and C. Talhi, "FedMCCS: Multicriteria client selection model for optimal IoT federated learning," *IEEE Internet of Things J.*, vol. 8, no. 6, pp. 4723–4735, Mar. 2021.
[18] T. Huang, H. Lin, L. Shen, K. Li, and A. Y. Zomaya, "Stochastic client selection for federated learning with volatile clients," *IEEE Internet of Things J.*, vol. 9, no. 20, pp. 20055–20070, Oct. 2022.
[19] Y. Deng et al., "AUCTION: Automated and quality-aware client selection framework for efficient federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 8, pp. 1996–2009, Aug. 2022.
[20] T. Huang, W. Lin, W. Wu, L. He, K. Li, and A. Y. Zomaya, "An efficiency-boosting client selection scheme for federated learning with fairness guarantee," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 7, pp. 1552–1564, Jul. 2021.
[21] Z. Jiang, Y. Xu, H. Xu, Z. Wang, and C. Qian, "Heterogeneity-aware federated learning with adaptive client selection and gradient compression," in *Proc. IEEE Conf. Comput. Commun.*, New York City, 2023, pp. 1–10.
[22] H. Wei, W. Ni, W. Xu, F. Wang, D. Niyato, and P. Zhang, "Federated semantic learning driven by information bottleneck for task-oriented communications," *IEEE Commun. Lett.*, vol. 27, no. 10, pp. 2652–2656, Oct. 2023.
[23] Y. Zhang, W. Xu, H. Gao, and F. Wang, "Multi-user semantic communications for cooperative object identification," in *Proc. IEEE Int. Conf. Commun. Workshops*, Seoul, 2022, pp. 157–162.
[24] H. Xie, Z. Qin, and G. Y. Li, "Task-oriented multi-user semantic communications for VQA," *IEEE Wireless Commun. Lett.*, vol. 11, no. 3, pp. 553–557, Mar. 2022.
[25] H. Xie, Z. Qin, X. Tao, and K. B. Letaief, "Task-oriented multi-user semantic communications," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2584–2597, Sep. 2022.
[26] S. D. Okegbile, B. T. Maharaj, and A. S. Alfa, "A multi-user tasks offloading scheme for integrated edge-fog-cloud computing environments," *IEEE Trans. Veh. Technol.*, vol. 71, no. 7, pp. 7487–7502, Jul. 2022.
[27] S. D. Okegbile, B. T. Maharaj, and A. S. Alfa, "A multi-class channel access scheme for cognitive edge computing-based Internet of Things networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 9912–9924, Sep. 2022.
[28] X. Shen, J. Gao, W. Wu, M. Li, C. Zhou, and W. Zhuang, "Holistic network virtualization and pervasive network intelligence for 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 1–30, First Quarter, 2022.
[29] K. Qu et al., "Stochastic cumulative DNN inference with RL-aided adaptive IoT device-edge collaboration," *IEEE Internet of Things J.*, vol. 10, no. 20, pp. 18000–18015, Oct. 2023.
[30] S. D. Okegbile, J. Cai, H. Zheng, J. Chen, and C. Yi, "Differentially private federated multi-task learning framework for enhancing human-to-virtual connectivity in human digital twin," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 11, pp. 3533–3547, Nov. 2023.
[31] X. He, X. Yi, Y. Zhao, K. H. Johansson, and V. Gupta, "Asymptotic analysis of federated learning under event-triggered communication," *IEEE Trans. Signal Process.*, vol. 71, pp. 2654–2667, Jul. 2023.
[32] B. Wang, J. Fang, H. Li, and B. Zeng, "Communication-efficient federated learning: A variance-reduced stochastic approach with adaptive sparsification," *IEEE Trans. Signal Process.*, vol. 71, pp. 3562–3576, Sep. 2023.
[33] Q. Wang and K. A. Lee, "Cosine scoring with uncertainty for neural speaker embedding," *IEEE Signal Process. Lett.*, vol. 31, pp. 845–849, Mar. 2024.
[34] C. Yi, J. Cai, K. Zhu, and R. Wang, "A queueing game based management framework for fog computing with strategic computing speed control," *IEEE Trans. Mobile Comput.*, vol. 21, no. 5, pp. 1537–1551, May 2022.
[35] L. Qin, S. Chen, and X. Zhu, "Contextual combinatorial bandit and its application on diversified online recommendation," in *Proc. SIAM Int. Conf. Data Mining*, 2014, pp. 461–469.
[36] L. Chen, J. Xu, and Z. Lu, "Contextual combinatorial multi-armed bandits with volatile arms and submodular reward," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 3251–3260.
[37] R. Konda, R. Chandan, D. Grimsman, and J. R. Marden, "Optimal utility design of greedy algorithms in resource allocation games," *IEEE Trans. Autom. Control*, vol. 69, no. 10, pp. 6592–6604, Oct. 2024.
[38] G. L. Nemhauser and L. A. Wolsey, "Best algorithms for approximating the maximum of a submodular set function," *Math. Operations Res.*, vol. 3, no. 3, pp. 177–188, Aug. 1978.
[39] Y. Hu, K. Ye, H. Kim, and N. Lu, "BERT-PIN: A BERT-based framework for recovering missing data segments in time-series load profiles," *IEEE Trans. Ind. Informat.*, vol. 20, no. 10, pp. 12241–12251, Oct. 2024.

Samuel D. Okegbile (Member, IEEE) received the PhD degree in computer engineering from the University of Pretoria, South Africa, in 2021. He is currently an assistant professor with the School of Computing, University of the Fraser Valley, Abbotsford, British Columbia, Canada. His research interests include various interesting topics in digital twin networks, human digital twins, Internet of Things, data sharing, artificial intelligence, wireless communication networks, and blockchain. He has received several awards, including the Horizon Postdoctoral Scholarship, the SENTECH Scholarship, and the University of Pretoria Doctoral Scholarship. He served as the publication chair for the 2023 Biennial Symposium on Communications. He is also a regular reviewer of some IEEE journals and conferences.

Haoran Gao is currently working toward the MSc degree with the Department of Electrical and Computer Engineering, Concordia University, Quebec, Canada. His research interests include federated learning and split learning.

Oluwasegun Talabi received the MASc degree in electrical and computer engineering from Concordia University, Montreal, Canada, in 2025, where he is currently working toward the PhD in electrical and computer engineering. He serves as a research assistant with the Network Intelligence and Innovation Laboratory, Department of Electrical and Computer Engineering, Concordia University. His research focuses on semantic communication, distributed learning, wireless communication networks, and human digital twin systems. He has been recognized with several prestigious awards, including the MTN Science and Technology Award, the Concordia University Merit Scholarship, and the Concordia University International Tuition Award of Excellence.

Jun Cai (Senior Member, IEEE) received the PhD degree in electrical engineering from the University of Waterloo, Waterloo, ON, Canada, in 2004. From 2004 to 2006, he was a post-doctoral fellow with the Natural Sciences and Engineering Research Council of Canada (NSERC), McMaster University, Canada. From 2006 to 2018, he was with the Department of Electrical and Computer Engineering, University of Manitoba, Canada, where he was a full professor and the NSERC industrial research chair. In 2019, he joined the Department of Electrical and Computer Engineering, Concordia University, Canada, as a full professor, and the PERFORM Centre research chair. His current research interests include edge/fog computing, eHealth, radio resource management in wireless communications networks, and performance analysis. He received the Best Paper Award from Chinacom in 2013, the Rh Award for outstanding contributions to research in applied sciences from the University of Manitoba in 2012, and the Outstanding Service Award from the IEEE GLOBECOM 2010. He served as the registration chair for QShine 2005, the Track/Symposium Technical Program Committee (TPC) co-chair for the IWCMC 2008, the IEEE GLOBECOM 2010, the IEEE VTC 2012, the IEEE CCECE 2017, and the IEEE VTC 2019, the Publicity co-chair for the IWCMC 2010, 2011, 2013, 2014, 2015, 2017, and 2020, the TPC co-chair for the IEEE GreenCom 2018, and the general chair for the 2023 Biennial Symposium on Communications. He also served on the editorial board for the *IEEE Internet of Things Journal*, *IET Communications*, and *Wireless Communications and Mobile Computing*.



Dusit Niyato (Fellow, IEEE) received the BEng degree from the King Mongkuts Institute of Technology Ladkrabang (KMITL), Thailand, in 1999, and the PhD degree in electrical and computer engineering from the University of Manitoba, Canada, in 2008. He is a professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include the areas of sustainability, edge intelligence, decentralized machine learning, and incentive mechanism design.

Xuemin (Sherman) Shen (Fellow, IEEE) received the PhD degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990. He is a University professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on network resource management, wireless network security, Internet of Things, AI for networks, and vehicular networks. He is a registered professional engineer of Ontario, Canada, an Engineering Institute of Canada fellow, a Canadian Academy of engineering fellow, a Royal Society of Canada fellow, a Chinese Academy of Engineering foreign member, an international fellow of the Engineering Academy of Japan, and a distinguished lecturer of the IEEE Vehicular Technology Society and Communications Society.