Privacy-Preserving Intelligent Content-Caching Scheme in Heterogeneous Aerial Access Networks

Arooj Masood[a] (arooj@uclab.re.kr), Nhu-Ngoc Dao[b*] (nndao@sejong.ac.kr),
Sungrae Cho[a*] (srcho@cau.ac.kr)

[a] School of Computer Science and Engineering,
Chung-Ang University, Seoul, 06974, Republic of Korea

[b]Department of Computer Science and Engineering,
Sejong University, Seoul, 05006, Republic of Korea

*Corresponding Authors: Sungrae Cho, Nhu-Ngoc Dao

Highlights

- Heterogeneous aerial access networks (AANs) enhance GTs' service in remote areas.
- Aerial edge caching meets on-demand content request challenges for GTs.
- HierFL-PCC enhances cache efficiency, ensures privacy, and reduces delays in AANs.
- HierFL-PCC reduces communication overhead in model training.

Abstract

Heterogeneous aerial access networks (AANs), comprising a hierarchical model of a low Earth orbit (LEO) satellite constellation and multiple high-altitude platforms (HAPs), provide a radio access medium from the sky to enhance the service experience of ground terminals (GTs) in remote areas. In this scenario, aerial edge caching addresses the challenge of delivering on-demand content requests to GTs. Although existing caching schemes for AANs have improved caching efficiency, they often overlook important concerns related to privacy preservation and overhead. This paper proposes an intelligent, proactive content-caching scheme to maximize cache efficiency while ensuring privacy preservation for GTs. The proposed caching scheme employs hierarchical federated learning (HierFL) that involves GTs, HAPs, and LEO satellites in heterogeneous AANs. In particular, the proposed HierFL-based proactive content-caching (HierFL-PCC) scheme leverages a deep neural network employing multiple linear regression to predict the dynamically changing content popularity of GTs, maximizing the cache efficiency. The HierFL-PCC also reduces the delay and overhead associated with content delivery. Additionally, the proposed HierFL-PCC scheme reduces communication overhead in model training owing to the hierarchical learning architecture. Simulation results demonstrate that the proposed HierFL-PCC scheme exhibits improvement of about 13%, 43%, 178%, and 457% in cache efficiency, and reduces average content delivery delay by approximately 10%, 20%, 31%, and 37%, along with an improvement of around 89% in training overhead, respectively, compared to the cloud-based FAVG, LFU, LRU, and Random caching schemes.

Keywords:
Aerial access network (AAN), privacy-preserving caching, aerial edge caching, hierarchical federated learning (HierFL), proactive content-caching

1. Introduction

Video traffic currently dominates approximately 70% of all mobile data traffic, which is expected to increase to 80% by 2028 (Ericsson, 2023). The surge in video content has increased the demand for a higher quality of service in content requests. While existing ground-based mobile access networks, characterized by high throughput, low latency, and high capacity, have played a crucial role in delivering on-demand contents to ground terminals (GTs), challenges persist in extending this service to rural communities in remote areas, where deploying ground base stations (BSs, e.g., gNBs and eNBs) is impractical.

In response to this challenge, we explore the capabilities of heterogeneous aerial access networks (AANs) (Dao et al., 2021). Heterogeneous aerial access platforms, including multiple high-altitude platforms (HAPs), and low Earth orbit (LEO) satellites, emerge as a complementary solution to overcome the limitations of existing ground-based mobile access networks in remote areas. HAPs, deployed at the stratosphere for line of sight access, play a vital role in providing wireless access to GTs Qiu et al. (2019). Simultaneously, LEO satellites have been exploited to provide global Internet access to serve rural communities in such areas. However, direct communication links between satellites and GTs are intermittent due to their long distances and high-speed movement. To address this, HAPs act as aerial BSs, providing closer and stable Internet access to GTs, thereby improving communication links between satellites and GTs with enhanced channel gain and reduced path loss (Kurt et al., 2021; Jia et al., 2020). In addition to wireless access, these aerial platforms offer edge computing and storage resources, functioning as aerial caching servers (ACSs). Aerial edge caching realized by these ACSs provides new opportunities to cache popular contents at the HAPs so that the requested content can be directly delivered from HAPs, instead of remote servers, reducing the end-to-end delay in content delivery.

However, proactively caching popular contents in the ACSs requires accurate predictions of content popularity. Although it has recently been widely researched, a notable gap exists in learning-based solutions for proactive content-caching within an integrated aerial and terrestrial networks (Bera et al., 2020; Wang et al., 2020; Zhang et al., 2020; Kang et al., 2020; Yu et al., 2018; Fadlullah and Kato, 2020; Yu et al., 2020; Cui et al., 2020). Specifically, few studies (Chen et al., 2017; Kang et al., 2020; Chen et al., 2019) have explored learning models trained on centralized collected data, where GTs upload data to a central server. Nonetheless, data are generated and distributed on mobile devices, and sending data to a central server poses security and privacy risks. While some studies have applied FL for privacy-preserving content-caching (Yu et al., 2018; Fadlullah and Kato, 2020; Yu et al., 2020; Cui et al., 2020), they often focus on a central parameter server, either at the edge or in the cloud. Edge-based FL enables efficient communication with GTs, but suffers from limited data access, resulting in suboptimal training performance. On the other hand, FL in the cloud accesses the numerous data required for model training, but costly communication with the cloud reduces training efficiency. Therefore, it is beneficial to combine edge-based and cloud-based FL approaches, ensuring efficient communication and training while providing access to many data for optimal training performance. To this end, a hierarchical FL (HierFL) framework was proposed by extending the federated averaging (FAVG) algorithm to a hierarchical configuration of a cloud parameter server and multiple edge parameter servers for model aggregations (Liu et al., 2020). This approach combined the strengths of edge- and cloud-based FL systems, allowing the cloud parameter server to access massive data samples and enabling edge parameter servers to facilitate efficient communications with GTs (Liu et al., 2020).

Thus, motivated by the advantages of heterogeneous AANs in extending network coverage to remote areas, we address the challenge of proactive content-caching in aerial access platforms to minimize content delivery delays. Additionally, by harnessing the privacy-preserving features of FL for decision-making problems, we apply FL to the content-caching challenge, ensuring secure model training without compromising the privacy of GTs. Specifically, we leverage the HierFL framework for proactive content-caching in heterogeneous AANs and propose a HierFL-based proactive content-caching (HierFL-PCC) scheme for heterogeneous AANs. The proposed HierFL-PCC scheme not only improves communication-computation tradeoffs by reducing costly communications with LEO satellites and substituting efficient communications with HAPs, it also ensures privacy of GTs, improves cache efficiency, and reduces content delivery delays.

In particular, the proposed HierFL-PCC scheme utilizes a deep neural network (DNN)-based multiple linear regression (MLR) approach to predict content popularity. GTs download a global model from LEO satellite via HAPs, train the DNN-MLR model with local data and upload updates to the HAP. The FAVG algorithm aggregates model updates at HAPs, and the aggregated models are sent to the LEO satellite for further aggregation. This HierFL training process continues until the model achieves desired accuracy. The proposed HierFL-PCC scheme predicts content popularity by leveraging DNN-MLR to learn contextual information between GTs and contents. It strategically caches the most popular contents in the ACSs, maximizing cache efficiency for frequently requested contents. This allows direct access from

HAPs, reducing delays and overhead associated with content delivery via the LEO satellite. In addition, the proposed HierFL-PCC scheme maintains the privacy of the GTs because the GT requests on the content preferences are preserved locally, and only the updates to the model parameters are uploaded to the HAPs and LEO satellite for model training. Moreover, the proposed HierFL-PCC scheme reduces communication overhead in model training due to its hierarchical architecture. The primary contributions of the paper are as follows.

- We investigated heterogeneous AANs scenario comprising a hierarchical model of LEO satellites and multiple HAPs to provide on-demand content requests in remote areas. The system was analyzed from multiple perspectives, such as the transmission model, caching model, delay model, privacy-preservation, and overhead calculation. Finally, a problem formulation was derived from these analyses.

- To address the proactive content-caching problem, we proposed the HierFL-PCC scheme to achieve a better communication-computation trade-off by reducing costly LEO-HAP communications and supplementing efficient HAP-GT communications in the model training. This method reduces the communication overhead involved in model training. In contrast, the HierFL-PCC scheme ensures that private data related to content requests and preferences of GTs are preserved locally, maintaining the privacy of the GTs.

- By improving the accuracy of content popularity prediction at the HAPs, the proposed HierFL-PCC scheme maximizes the cache efficiency, allowing the frequently requested content to be directly accessed from the HAPs, reducing the content delivery delay of the GTs and overhead of the HAPs.

- The simulation results demonstrate that the proposed HierFL-PCC scheme improves approximately 13%, 43%, 178%, and 457% in terms of cache efficiency, reduces average content delivery delay by approximately 10%, 20%, 31%, and 37%, and improves training overhead by around 89%, respectively, compared to the cloud-based FAVG, LFU, LRU, and Random caching schemes.

To the best of our knowledge, the proposed HierFL-PCC scheme is the first proactive content-caching scheme that considers privacy-preserving caching to maximize cache efficiency and reduce content delivery delay and overhead at the level of HAPs under the LEO satellite constellation in remote areas within heterogeneous AANs.

The remainder of this paper is organized as follows. A review of the related studies is provided in Section 2. The system model and problem formulation are discussed in Section 3. Next, Section 4 details the proposed HierFL-based proactive content-caching scheme and the corresponding algorithm. The experimental results are presented in Section 5. Finally, the conclusions are summarized in Section 6.

2. Related Work

This section reviews recent and relevant studies related to this work on content-caching. These studies can be classified into two approaches: conventional optimization schemes and learning-based schemes. Studies dealing with conventional caching optimization methods for cache-enabling single and multiple uncrewed aerial vehicle (UAV) networks are described in Jiang et al. (2018); Xu et al. (2018); Zhang et al. (2020); Wang et al. (2020); Bera et al. (2020). Moreover, studies dealing with learning-based solutions are described in Chen et al. (2017); Kang et al. (2020); Chen et al. (2019); Ndikumana et al. (2020); Masood et al. (2021a); Zhao et al. (2021); Wang et al. (2023); Qiao et al. (2019); Yu et al. (2018, 2020); Fadlullah and Kato (2020); Cui et al. (2020); Maale et al. (2023); Masood et al. (2021b); Feng et al. (2023).

Under the conventional caching optimization schemes, Jiang et al. (2018), considered a cache-enabled UAV in the Internet of Things network and optimized the UAV height and content placement to maximize throughput. In Xu et al. (2018), content-centric UAV communication was considered, where a UAV is deployed to serve a group of ground nodes (GNs) by proactively transmitting the files to selected GNs. The selected GNs collaboratively cache all files, allowing each GN to retrieve a requested file from its local cache or nearest neighbor through device-to-device communication. In addition, Zhang et al. (2020); Wang et al. (2020), investigated a joint optimization problem of UAV deployment, caching placement, and user association in cache-enabling UAV-assisted cellular networks and proposed a joint iterative algorithm to solve the optimization problem Further, Bera et al. (2020), studied the quality of experience analysis in cache-enabled multi-UAV networks. They formulated a linear programming problem to maximize the average content delay index, where the optimized 2D position of each UAV is determined using the K-means algorithm, and optimized cache policies are found using linear regression and the Zipf distribution. While previous studies (Jiang et al., 2018; Xu et al., 2018; Zhang et al., 2020; Wang et al., 2020; Bera et al., 2020) have primarily focused on the content-caching problem using single and multiple UAVs, and have overlooked the practicality of using various aerial platforms. In our work, we consider HAPs to be the edge caching servers that provide wide coverage and high payload capabilities, primarily remaining at one point for a long time. Hence, they can provide long-term, continuous wireless access to GTs in remote areas. Furthermore, these existing studies considered a fixed UAV deployment scenario, and the popularity of contents followed the Zipf distribution. In contrast, we introduce an intelligent proactive content caching solution that leverages advanced algorithms to learn content popularities based on contextual information from GTs and content characteristics.

Recently, machine learning (ML) approaches, such as deep learning (DL) and reinforcement learning, have been widely researched to solve complex decision-making problems in dynamic and uncertain environments (Masood et al., 2020, 2023). For instance, Chen et al. (2017), introduced a conceptor-based echo state network that predicts the content request distribution and mobility patterns of users. Afterward, they deployed cache-enabled UAVs to deliver the requested content to the predicted locations. Kang et al. (2020) proposed a DL-based UAV deployment and content-caching scheme where users are grouped using the K-means clustering algorithm. They proposed a long short-term memory-based caching scheme to cache popular content on UAVs. Chen et al. (2019), investigated joint caching and a resource allocation scheme for cache-enabled UAVs. They proposed a distributed algorithm based on liquid state machines to predict content popularity and self-organizing resource allocation. Ndikumana et al. (2020) proposed a content-caching scheme for self- driving cars considering passenger features obtained using convolutional neural networks. Moreover, Masood et al. (2021a), proposed a deep regression-based video popularity estimation for proactive video caching in multi-access edge computing networks. In addition, Zhao et al. (2021), proposed caching at the vehicle and small BSs by employing the upper confidence bound algorithm to learn the special interests and request preferences of all users. Further, Wang et al. (2023) considered the timeliness index as a measure of content freshness and explored caching placement using proximal policy optimization with deep reinforcement learning. Furthermore, Qiao et al. (2019) proposed cooperative edge caching as a double time-scale Markov decision process to jointly optimize the content placement and delivery in vehicular edge computing networks. These researches in Masood et al. (2020); Chen et al. (2017); Kang et al. (2020); Chen et al. (2019); Ndikumana et al. (2020); Masood et al. (2021a); Zhao et al. (2021); Wang et al. (2023); Qiao et al. (2019) focused on deep and reinforcement learning-based solutions for the content-caching problem, heavily relying on centralized learning models where GTs must upload data to the central server, posing data privacy issues. In addition, as the number of GTs increases, such existing approaches cannot handle the large volume of data due to the high computation and communication costs. In contrast, our approach leverages distributed training of the ML model and focuses on privacy-preservation of GTs, enabling efficient scalability to manage data volume as the number of GTs increases.

Recently, FL has emerged as a solution to address these concerns by enabling distributed training without the need to collect private data and has been employed in a few content-caching studies (Yu et al., 2018, 2020; Fadlullah and Kato, 2020; Cui et al., 2020; Maale et al., 2023; Masood et al., 2021b; Feng et al., 2023). For instance, Yu et al. (2018) proposed an edge-based FL content-caching scheme for multi-access edge computing networks that estimated content popularity using hybrid filtering based on the similarity of users and files, where the latent features were extracted using an autoencoder. Yu et al. (2020), introduced an FL-based mobility-aware proactive edge caching scheme called MPCF for urban vehicular networks. The scheme uses an adversarial autoencoder to estimate content popularity. In addition, Fadlullah and Kato (2020) explored a heterogeneous computing platform for joint computation, communication, and collaborative caching in an integrated aerial-terrestrial network with ultra-dense tiny cells. The heterogeneous computing platform controller trains a global model using a two-stage FL approach with asynchronous parameter updates, whereas local updates are conducted using convolutional neural networks. Additionally, Cui et al. (2020) designed a blockchain-assisted compressed algorithm of FL for content-caching, called CREAT, to improve the cache hit rate in edge computing. Similarly, Maale et al. (2023), proposed a deep federated echo state learning content-caching scheme to predict popular content requests and mobility patterns, using the context information from the user equipment, such as the device type, location, and other data on the device.

However, some studies (Yu et al., 2018, 2020; Fadlullah and Kato, 2020; Cui et al., 2020; Maale et al., 2023) have primarily focused on using a central

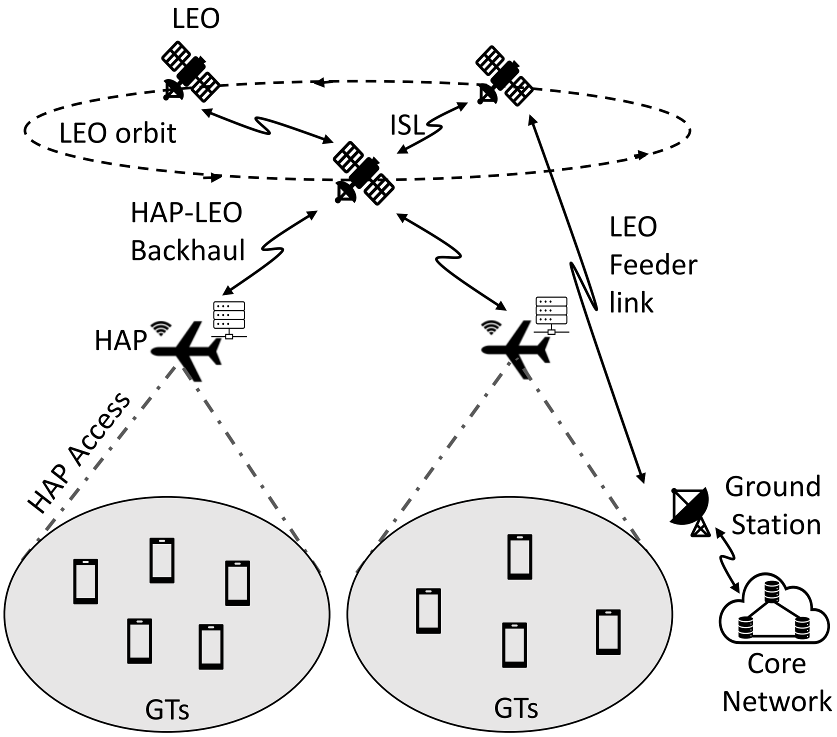

Figure 1: Heterogeneous aerial access network (AAN) model consisting of multiple high-altitude platforms (HAPs) under the coverage of a low Earth orbit (LEO) satellite constellation to provide content requests to ground terminals (GTs) in remote areas.

parameter server, either at the edge or in the cloud, for FL in dense urban scenarios, which faces challenges due to the limited access to data or the significant communication overhead when communicating with the cloud. Masood et al. (2021b), leveraged the HierFL framework based on one cloud parameter server and multiple edge parameter servers in HAP-assisted multi-UAV networks . Similarly, Feng et al. (2023) proposed a HierFL-based cooperative caching network architecture for vehicle trajectory and content popularity prediction in urban vehicular networks. In contrast, we consider a heterogeneous AANs comprising a hierarchical model of LEO satellite constellation and HAPs, and propose a HierFL-PCC scheme to provide on-demand content requests to GTs in remote areas. The proposed HierFL-PCC scheme aims to maximize cache efficiency and reduces content delivery delay and overhead. Additionally, our proposed HierFL-PCC scheme based on the HierFL model reduces communication overhead in training the model.

Table 1: Key Notation Descriptions

Notation	Description
$\mathcal{I}$	Set of all GTs in the heterogeneous AAN system
$\mathcal{H}$	Set of HAPs in the heterogeneous AAN system
$\mathcal{V}$	Set of LEO satellites in the heterogeneous AAN system
$\mathcal{I}_h^t$	Set of GTs associated with HAP h at time slot t
$\mathcal{N}$	Set of available content
$\mathcal{C}_h$	Set of content cached in HAP h
λ_t	Rate parameter of the Poisson process
$a_{i,h}$	Association indicator of the GT with HAP h
$b_{i,h}$	Indicator if GT i places a content request
$\mathcal{B}_h$	Downlink bandwidth of HAP h
$\mathcal{G}_{h,i}$	Channel power gain between HAP h and GT i
g_h, g_i	Antenna power gains of HAP h and GT i, respectively
$d_{h,i}[t]$	Distance between HAP h and GT i
$\mathcal{R}_{h,i}$	Transmission rate between HAP h and GT i
$\mathcal{P}_{h,i}$	Transmission power of HAP h
$b_{h,v}$	Association indicator of HAP with LEO satellite
$R_{v,h}$	Transmission rate of the LEO satellite
$\mathcal{P}_{v,h}$	Transmission power of the LEO satellite
M_h	Cache capacity of HAP h
s_i	Size of the content requested by GT i
$P_{h,c}$	Probability of caching c popular content in HAP h
ρ_c	Predicted popularity of c
$\mathcal{P}_h$	Set of caching policies
ζ_i	Delay during content delivery to GT i
δ_{avg}^t	Average delay for each GT i at time slot t
$\mathcal{O}_{FL}$	Overhead of model training
$\mathcal{O}_{cache}^t$	Overhead of cache inefficiency
D'	Dataset distributed across I GTs
$D_i^{'}$	Dataset of GT i
$l_i(w)$	Local loss function with weight w
$L(w)$	Loss function
w	Weight value
β	Batch size

3. Problem Statement

This section analyzes the heterogeneous AAN model from the communication, caching, delay, and overhead perspectives. Table 1 lists the main notation used in this study.

3.1. System Model

We considered a heterogeneous AAN comprising several LEO satellites, multiple HAPs, and several GTs, as depicted in Fig. 1. We assumed no terrestrial BS is available for the GTs, as in remote areas. Although LEO satellites can directly provide content to the GTs, the severe path loss between GTs and LEO satellites poses challenges in providing high throughput Internet access for latency-sensitive applications. In contrast, HAPs deployed at the stratosphere at an altitude, A_H, of 20 to 50 Km, can provide line-of-sight communication and wide coverage with a radius of 50 to 500 KmKurt et al. (2021). Hence, using HAPs is considered a potential solution to improve the quality of service in remote areas. Thus, the GTs are directly connected to HAPs, which are directly connected to LEO satellite constellations.

On the ground, at an arbitrary time slot, I GTs are denoted as $i \in \mathcal{I} = \{1, 2, \ldots, I\}$, which are randomly distributed. Additionally, H HAPs denoted as $h \in \mathcal{H} = \{1, 2, \ldots, H\}$ are deployed as aerial BSs under the coverage of the LEO satellite constellation $\mathcal{V}$. In this system model, one GT, i, can only be associated with one HAP, h, but one HAP, h can be associated with several GTs. We defined the set of I_h^t GTs connected with HAP h at time slot t as $\mathcal{I}_h^t = \{1, 2, \ldots, I_h^t\}$. We let $\mathcal{N} = \{1, 2, \ldots, N\}$ denote the index set of available content. Each HAP, h, is equipped with an onboard content-caching capability of limited storage resources, which can be used as an ACS to cache popular content that the GTs request. The set of C cached content in the ACS of HAP h is denoted by $\mathcal{C}_h$, where $C \leq N$ and $h \in \mathcal{H}$. Moreover, the LEO satellite constellation $\mathcal{V}$ (denoted by $\mathcal{V} = \{1, 2, \ldots, V\}$), which is equipped with enhanced computing and storage resources, is placed at an orbital altitude, A_v, of 500 kmJia et al. (2017). Without loss of generality, we assumed that the software-defined networking (SDN) technology is used to virtually expose a central satellite access point to cache set $\mathcal{N}$ for all N available contents and facilitate communication and coordination of the limited caching resources (i.e., the ACSs on the HAPs). Furthermore, when a direct connection between a HAP and the central LEO satellite is lost, our SDN controller dynamically reroutes data transmission through other available LEO satellites within the constellation through inter-satellite links. This proactive approach minimizes the delay caused by the loss of direct connection. In this model, LEO satellites provide backhaul services for HAPs. The LEO satellite operates in the Ku band and is also connected to a terrestrial gateway via a feeder link, providing access to the core network and the Internet. The ACS on the HAPs obtains content from the LEO satellite, $v \in \mathcal{V}$, and proactively caches $\mathcal{C}_h$ popular content so that large transmission delays can be reduced as each HAP h can directly transmit its cached content to the GTs. We let $b_{i,h} = 1$ indicate whether a GT i places a content request to HAP h for time slot t; otherwise, $b_{i,h} = 0$.

The downlink bandwidth of HAP h is $\mathcal{B}_h$ Hz and the bandwidth of the LEO satellite v backhaul link is $\mathcal{B}_v$ Hz. As LEO satellites move at high speed along their orbits and have very limited contact time with the HAP (h), they may not be able to learn the global model training during its contact time with the HAP (h). As mentioned, SDN technology is leveraged to create a virtual LEO satellite for model training. This virtualization allows HAPs to logically connect to a central LEO satellite, even though the HAPs may be physically connected to different LEO satellites within a constellation. Hence, SDN-based virtualization enables coordination and communication between HAPs and the central LEO satellite, hierarchically facilitating the distributed learning process.

3.2. Transmission Model

This section introduces the transmission models between GTs and HAPs and HAPs and LEO satellites in the heterogeneous AAN.

3.2.1. HAP-GT Transmission Model

We only consider remote areas where terrestrial BSs do not exist; therefore, line-of-sight communication links can be made between HAPs and GTs. For the HAP-GT (air-to-ground) link, all HAPs operate in the Ku band and that the spectrum allocation for all HAPs is managed to avoid any possible intercell interference, whereas GTs were assigned to different orthogonal sub channels without significant interchannel interference. Consequently, the propagation channel of HAPs and GTs is modeled using the free space path loss model. Each GT is associated with only one HAP in a time slot t, which can be expressed as follows:

$$\sum_{h=1}^{H} a_{i,h}[t] = 1, \forall i \in \mathcal{I} \tag{1}$$

where $a_{i,h} = \{0, 1\}, \forall h \in \mathcal{H}$ is the association indicator for GT i, where $a_{i,h} = 1$ indicates GT i is associated with HAP h, and $a_{i,h} = 0$ otherwise. In addition, the capacity of each HAP is limited and should satisfy the following:

$$\sum_{i=1}^{I} a_{i,h}[t] \cdot B_h^i[t] \leq \mathcal{B}_h, \quad \forall h \in \mathcal{H}, \tag{2}$$

where $\mathcal{B}_h$ is the available bandwidth of HAP h, and B_h^i indicates the allocated bandwidth for GT i by HAP h. Then, (2) ensures that the total bandwidth used by the associated GTs does not exceed the available bandwidth $\mathcal{B}_h$ of HAP h. We considered a line-of-sight communication link between HAP and GT; thus, the channel power gain between HAP h and GT i in time slot t can be obtained as follows:

$$\mathcal{G}_{h,i}[t] = \nu d_{h,i}^{-\gamma_1}[t] = \frac{\nu}{(\,A_H^2 + \parallel \varkappa - \varphi \parallel^2)^{\,0.5\gamma}}, \forall i \in \mathcal{I}, \tag{3}$$

where $\nu = g_h g_i (\frac{\Lambda}{4\pi d_0})^{\,2}$ is the channel power gain at the reference distance $d_0 = 1\ m$. In addition, Λ denotes the wavelength, and g_h and g_i represent the antenna power gains of the HAP h and GT i, respectively. $d_{h,i}[t]$ denotes the distance between HAP h and GT i, and $\gamma_1 \geq 2$ is the path loss exponent. A_H^2 represents the square of the altitude difference that accounts for the vertical distance between the HAP h and the GT i. The horizontal locations of HAP $h \in \mathcal{H}$ and GT $i \in \mathcal{I}$ are denoted as $\varkappa$ and φ, respectively, and the horizontal distance between HAP h and GT i is represented as $\parallel \varkappa - \varphi \parallel^2$. Then the achievable transmission rate between HAP h and GT i for each content transmission can be obtained as follow:

$$\mathcal{R}_{h,i}[t] = B_h^i[t] \cdot \log_2(\,1 + \frac{\mathcal{P}_{h,i}[t]\mathcal{G}_{h,i}[t]}{\sigma_i^2}) \tag{4}$$

where $\mathcal{P}_{h,i}[t]$ is the transmission power of HAP h to GT i in time slot t, and σ_i^2 is the variance of the Gaussian noise.

3.2.2. LEO-HAP Transmission Model

For the LEO-HAP (space-to-air) transmission, each HAP can only associate with one LEO satellite, but the LEO satellite can serve multiple H_v HAPs at a time slot, which can be expressed as follows:

$$\sum_{v=1}^{V} b_{h,v}[t] = 1, \forall h \in \mathcal{H}, \tag{5}$$

$$\sum_{h=1}^{H} b_{h,v}[t] \cdot B_v^h[t] \leq \mathcal{B}_v, \quad \forall v \in \mathcal{V}, \tag{6}$$

where $b_{h,v} = \{0, 1\}, \forall v \in \mathcal{V}$ is the association indicator for HAP h (i.e., $b_{h,v} = 1$ indicates HAP h is associated with LEO satellite v, and $b_{i,h} = 0$ otherwise). In addition, B_v^h is the allocated bandwidth for HAP h, and $\mathcal{B}_v$ is the available bandwidth of the LEO satellite. Unlike the HAP-GT propagation channel, the LEO-HAP (space-to-air) channel can be modeled using the Rician fading channel model with the additive white Gaussian noise, where the channel fading coefficient is modeled as a circular symmetric complex Gaussian random variable (i.e., $S_v = X_1 + X_2 i$), where $X_1 \sim \mathcal{N}(\mu_1, \frac{\sigma^2}{2})$ and $X_2 \sim \mathcal{N}(\mu_2, \frac{\sigma^2}{2})$. Then, the achievable transmission rate of LEO satellite v for the backhaul data transmission at time slot t is expressed as

$$\mathcal{R}_{v,h}[t] = B_v^h[t] \cdot \log_2\left(1 + \frac{\mathcal{P}_{v,h}[t]|S_v|^2 d_{v,h}^{-\gamma_2}[t]}{\sigma_h^2}\right) \tag{7}$$

where B_v^h is the allocated bandwidth for HAP h, $\mathcal{P}_{v,h}$ denotes the transmission power of LEO satellite v to HAP h, $d_{v,h}$ represents the distance from the LEO satellite to HAP h, and σ_h^2 indicates the variance of the additive white Gaussian noise at HAP h.

3.3. Caching Model

This section describes the caching model in the heterogeneous AAN. In the considered heterogeneous AAN, $\mathcal{I}_h^t$ GTs are connected to each HAP, $h \in \mathcal{H}$. These GTs are interested in a set of popular content and send content requests to the connected HAP, h. We assume that the number of such content requests that arrive in a time slot t follows a Poisson distribution with the parameter λ_t. Each HAP, h, proactively caches $\mathcal{C}_h$ content in its ACS locally. If the requested content is cached (i.e., a cache hit), the HAP can directly transmit the content to the GT. Otherwise, the HAP must fetch the requested content from the LEO satellite (i.e., a cache miss).

Without loss of generality, HAP h has a limited caching capacity of M_h. Due to the limited cache space, a HAP cannot store all popular contents. This necessitates to develop a policy for content placement in the cache. The probability of caching popular content c in HAP h can be defined as follows:

$$P_{h,c} = \frac{\rho_c}{\sum_{c \in \mathcal{C}_h} \rho_c}, \quad 0 \leq P_{h,c} \leq 1, \tag{8}$$

where ρ_c denotes the predicted popularity value of $c \in \mathcal{C}_h$. Content popularity is influenced by contextual information, such as age, gender, and occupation. The popularity of content c, denoted as, ρ_c, is computed using our DNN-MLR approach within a HierFL framework, which allows to learn the contextual information between GTs and contents. Then, for each HAP h, the set of cache policies, $\mathcal{P}_h$, can be expressed as follows:

$$\mathcal{P}_h = P_{h,1}, P_{h,2}, \ldots, P_{h,C_h}, \tag{9}$$

where $P_{h,c}$ represents the probability of caching popular content c in HAP h. Employing predicted content popularity for probabilistic caching is advantageous over the conventional popularity-based methods because it caches contents according to the GTs' preferences and enhances cache efficiency by avoiding excessive caching of popular contents Bera et al. (2020). High cache hit rates, which reduce content delivery delays, are crucial for enhancing cache performance, especially by caching popular content in the HAPs.

3.4. Delay Model

Once a content request is generated in a time slot t, the delay during content delivery may occur for one of two reasons. First, the time to transmit the content from HAP h is delayed if the content is cached in the cache of HAP h. Second, if the content is not cached, the delay includes the time required to transmit the content from LEO satellite v to HAP h, followed by the transmission of the requested content from HAP h to GT i. Therefore, we can calculate the delay during the delivery of the requested content from HAP h to GT i as follows Bera et al. (2020):

$$\zeta_i = \frac{s_i}{\mathcal{R}_{h,i}} + (1 - P_{h,c}) \frac{s_i}{\mathcal{R}_{v,h}}. \tag{10}$$

Then, the average delay for each GT i in a time slot t can be calculated as follows:

$$\delta^t_{avg} = \frac{\sum_{h=1}^{H}\sum_{i=1}^{I_h^t} b_{i,h}\cdot\zeta_i}{\sum_{h=1}^{H}\sum_{i=1}^{I_h^t} b_{i,h}}. \tag{11}$$

where $b_{i,h} = 1$ if the GT i generates a content request; otherwise, $b_{i,h} = 0$, and ζ_i indicates a delay during the delivery of requested content from HAP h to GT i.

3.5. Privacy Preservation and Overhead

As discussed in Section 2, most existing ML-based approaches have neglected the privacy concerns associated with GT data. In particular, centralized ML-based approaches involve transmitting GT data to the central server over the network, neglecting privacy protection and infringing upon GT privacy. Moreover, the overhead for model training is proportional to the volume of data required for training the DL model at the central server. In other words, all required data must be delivered throughout the networks and consume significant communication resources.

In contrast, distributed ML approaches, such as FL, allow distributed training of the shared global model across multiple GTs. Specifically, GTs train DL models using their local data and share their model updates with a central server. The central server aggregates these updates and returns an updated global model. This approach ensures that sensitive GT data remain on local devices and that only model updates are shared with the central server. The communication overhead can be determined by the two key factors 1) the size of model updates (Q) transmitted between GT i and HAP h or between HAP h and LEO satellite v, and 2) the total communication rounds required to achieve the desired accuracy or convergence. Accordingly, the training overhead for model updates $\mathcal{O}_{FL}$ is calculated as follows:

$$\mathcal{O}_{FL} = \left(\sum_{h=1}^{H}\sum_{i=1}^{\mathcal{I}_h^t}\frac{2K}{\kappa_1} + \sum_{v=1}^{V}\sum_{i=1}^{H}\frac{2K}{\kappa_1\kappa_2}\right)Q, \tag{12}$$

where K denotes the total number of training epochs, and κ_1 and κ_2 represent the number of epochs executed locally at the GT before updating the model to the HAP and at the HAP before updating the model to the central LEO satellite, respectively. To elaborate further, the first term accounts for the overhead associated with each GT and HAP communication for model updates and the second term represents the overhead between HAPs and LEO satellites in the HierFL process. Specifically $\frac{2K}{\kappa_1}$ corresponds to the number of times HAPs aggregate local updates from GTs and $\frac{2K}{\kappa_1 \kappa_2}$ represents the aggregation of models by LEO satellites. The multiplication by 2 in these terms accounts for bidirectional communication between GT and HAP as well as HAP and LEO satellite.

In terms of the caching operation, the overhead due to caching inefficiency $\mathcal{O}_{cache}^t$ can be defined as the volume of data that must be directly delivered from LEO satellites to respond to GT requests as the requested content has not been cached at the HAPs yet. Accordingly, $\mathcal{O}_{cache}^t$ can be calculated as follows:

$$\mathcal{O}_{cache}^t = P_{h,c} s_i. \tag{13}$$

Then, the average overhead due to caching inefficiency over all GTs in a time slot t can be calculated as follows:

$$\mathcal{O}_{avg}^t = \frac{\sum_{h=1}^{H} \sum_{i=1}^{\mathcal{I}_h^t} b_{i,h} \cdot \mathcal{O}_{cache}^t}{\sum_{h=1}^{H} \sum_{i=1}^{I_h^t} b_{i,h}}. \tag{14}$$

where $b_{i,h} = 1$ if the GT i generates a content request; otherwise, $b_{i,h} = 0$, and $\mathcal{O}_{cache}^t$ denotes the caching inefficiency overhead.

3.6. Problem Formulation

In the considered heterogeneous AANs, GTs generate content requests. Afterward, HAPs deliver this content (i.e., a cache hit), and the LEO satellite serves as a backhaul for the HAPs (i.e., a cache miss). This study aims to predict content popularity and proactively cache popular contents optimizing cache efficiency while preserving privacy in heterogeneous AANs. This objective is achieved by learning the contextual information between GTs and contents using a DNN-MLR approach within a HierFL framework. Cache efficiency and content delivery delay and caching inefficiency overhead are inversely proportional; thus, we aim to minimize content delivery delay and caching inefficiency overhead while optimizing the cache policy, $\mathcal{P}_h$, to achieve this goal. Therefore, the cache policy, $\mathcal{P}_h$, should be optimized to obtain the minimum value of content delivery delay, δ_{avg}^t, and the minimum value of caching inefficiency overhead, $\mathcal{O}_{avg}^t$, for time slot t. In addition, the training overhead, $\mathcal{O}_{FL}$, for model updates can be independently reduced by applying an appropriate Hier-FL model. Hence, we involved, δ_{avg}^t, and, $\mathcal{O}_{avg}^t$, in the objective function and formulate the optimization problem as follows:

$$\begin{aligned} \min_{\mathcal{P}_h} \quad & \gamma_d \delta_{avg}^t + \gamma_o \mathcal{O}_{avg}^t \\ \text{s.t} \quad & (1), (2), (5), (6), (8), \end{aligned} \tag{15}$$

where γ_d and γ_o denote the weights assigned to the average content delivery delay and caching inefficiency overhead, respectively. To address this problem, we employ a DNN-MLR-based HierFL approach, enhancing content popularity prediction accuracy and, consequently, optimizing the cache policy, $\mathcal{P}_h$. This optimized cache policy influences both content delivery delay and caching inefficiency overhead, aligning with the objectives defined in the optimization problem. Thus, the proposed DNN-MLR-based HierFL approach and the formulated optimization problem are intertwined, jointly dedicated to minimizing average content delivery delay and caching inefficiency overhead while optimizing the cache policy to enhance overall network performance.

4. Hierarchical Federated Learning-Based Proactive Content-Caching Scheme

This section elaborates on the proposed HierFL-PCC scheme in heterogeneous AANs. We first describe federated DL based on DNN-MLR and then introduce the HierFL-PCC. The proposed HierFL-PCC scheme aims to enhance cache efficiency by improving the accuracy of the content popularity prediction with privacy preservation while reducing the training overhead, caching inefficiency overhead, and content delivery delay. Determining the optimal value of cache policy $\mathcal{P}_h$ for the optimization problem (15) can be directly translated to finding an accurate prediction of the popularity of cached content at HAP h Fadlullah and Kato (2020); Bera et al. (2020).

Recently, DL has gained popularity in predicting content popularity in edge caching networks. However, traditional DL models are trained using centralized data collection, which poses privacy concerns. Therefore, FL was introduced to enable distributed training without centralizing the data to address this issue. However, most FL-based solutions rely on a single parameter server on the cloud or edge, leading to poor training efficiency or performance loss. HierFL-based approaches have been introduced to overcome these limitations, comprising a hierarchical model with a cloud parameter server and multiple edge parameter servers. This approach achieves a better computation-communication training trade-off and preserves data privacy by preserving sensitive data at the GTs, while the cloud parameter server and multiple edge parameter servers are trained on aggregated data. Similar to FL, the goal of HierFL is to minimize the loss function for an accurate prediction of the content popularity:

$$\min_{w} L(w) = \sum_{i=1}^{I} \frac{|D_i^{'}|}{|D'|} l_i(w), \tag{16}$$

$$l_i(w) = \frac{1}{|D_i^{'}|} \sum_{j \in D_i^{'}} l_j(w), \tag{17}$$

where $D' = \{x_j, y_j\}_{j=1}^{|J|}$ is the dataset distributed across I GTs, x_j denotes the jth data sample, y_j represents the corresponding label, and $|J|$ indicates the total number of data samples. Moreover, $D_i^{'}$ is the local dataset of GT i, and $l_i(w)$ is the loss of the prediction on the $D_i^{'}$ dataset with the parameter w. Minimizing the weighted average of the local loss function optimizes the loss function *L(w)* on FL.

Thus, we use the FL approach to solve the problem in this study. In particular, we propose a proactive content-caching scheme based on HierFL. In the proposed scheme, multiple GTs train a DNN-MLR model, where each HAP is treated as an edge parameter server, and the LEO satellite is considered a cloud parameter server for HierFL model aggregations. The proposed scheme is HierFL-PCC, described in detail in the following section.

4.1. Federated Deep Learning

Federated learning (FL) is a distributed training algorithm that involves two main steps: a local model update based on the dataset stored on the GTs and a global model aggregation at the cloud or edge. In FL, the cloud or edge server acts as the central parameter server, and GTs within its communication range collaborate to train a DL model. To predict the content popularity for proactive content-caching, we trained a DNN-MLR model. A MLR model for GT i with k predictor variables $x_i^1, x_i^2, \cdots, x_i^k$ and a dependent variable P_{y_i} can be defined as follows:

$$P_{y_i} = w_0 + x_1 w_1 + x_2 w_2 + \cdots + x_k w_k + \epsilon, \tag{18}$$

where $w_1, w_2, \cdots, w_k$ represent the coefficients of the predictor variables, and ϵ is a random experimental error. The application of MLR, in our model provides a computationally efficient way to account for the influence of multiple predictor variables on content popularity. Furthermore, the addition of a DNN enhances the model's ability to capture complex and non-linear relationships, thereby improving the prediction accuracy. The DNN is trained with the Adam optimization algorithm and allowed the weights to be adjusted to minimize loss during subsequent evaluations. The proposed DNN-MLR method employs the mean squared error (MSE) as the loss function, measuring the difference between predicted and actual values for each data sample. Specifically, the loss function is defined as follows:

$$L(w, \epsilon) = \frac{1}{|J|} \sum_{j=1}^{J} (P_{y_j} - y_j)^2, \tag{19}$$

where $L(w, \epsilon)$ captures the prediction error of the model for the jth data sample. The step-by-step procedure of DNN-MLR is given in Algorithm 1. The training dataset is distributed across $\mathcal{I}$ GTs in the FL setting. Thus, we computed the loss on local datasets distributed among $\mathcal{I}$ GTs, which can be computed in the form of a weighted average of the local loss function:

$$l_i(w, \epsilon) = \frac{1}{|D_i'|} \sum_{j \in D_i'} l_j(w, \epsilon), \tag{20}$$

$$\min_{w} L(w, \epsilon) = \sum_{i=1}^{I} \frac{|D_i'|}{|D'|} l_i(w, \epsilon). \tag{21}$$

The FL algorithm communicates and aggregates local models every k step of the optimization algorithm (e.g., Adam) performed on the local dataset at each GT i to minimize the communication overhead (McMahan et al., 2017). Specifically, the model is trained locally on GTs for k steps until $k \neq 0$. Then, each central parameter server aggregates local models from its GTs when $k = 0$. The parameters of the local model on GT i are denoted as $w_i(k)$,

Algorithm 1 Deep neural network multiple linear regression (DNN-MLR) execution on a ground terminal

```
procedure DNN-MLR($D_i^{'}, w, \epsilon$)
    $X \longleftarrow$ Features
    $Y \longleftarrow$ Labels
    Initialize $\alpha$
    for each epoch $k$ = 1, ..., $K$ do
        for each batch $\beta$ do
            $x_j \longleftarrow$ Features ($\beta$)
            $y_j \longleftarrow$ Label ($\beta$)
            $P_{y_j} \longleftarrow$ Model($x_j$)
            Calculate loss $l_j(P_{y_j}, y_j)$
        end for
    end for
end procedure
```

and the evolution of $w_i(k)$ in FL is described by the following equation:

$$w_i(k) = \begin{cases} w_i(k-1) - \eta_k \nabla l_i(w_i(k-1)) & k|\kappa \neq 0 \\ \\ \frac{\sum_{i=1}^{\mathcal{I}} |D_i^{'}|[w_i(k-1) - \eta_k \nabla l_i(w_i(k-1))]}{|D^{'}|} & k|\kappa = 0. \end{cases}, \tag{22}$$

where η_k denotes the fixed learning rate (LR). Aggregating global models at the central parameter server can be considered a method of sharing information between GTs. In particular, model aggregation at the cloud parameter server can involve numerous GTs, but it incurs high communication costs. Conversely, model aggregation at the edge parameter server involves only a small number of GTs, and the communication cost is much lower. Therefore, the cloud-based and edge-based FL techniques are similar in architecture but differ in communication costs and the number of participants. A trade-off exists between communication efficiency and the convergence rate for cloud-based FL, where less communication requires more local computation (Li et al., 2019). In contrast, edge-based FL provides computation latency comparable to communication to the edge server. However, one drawback of this approach is that only a limited number of GTs can be accessed by each edge parameter server, reducing training performance. Thus, combining cloud and edge-based FL can be beneficial, allowing access to numerous training samples from cloud-based FL while benefiting from efficient model updates with

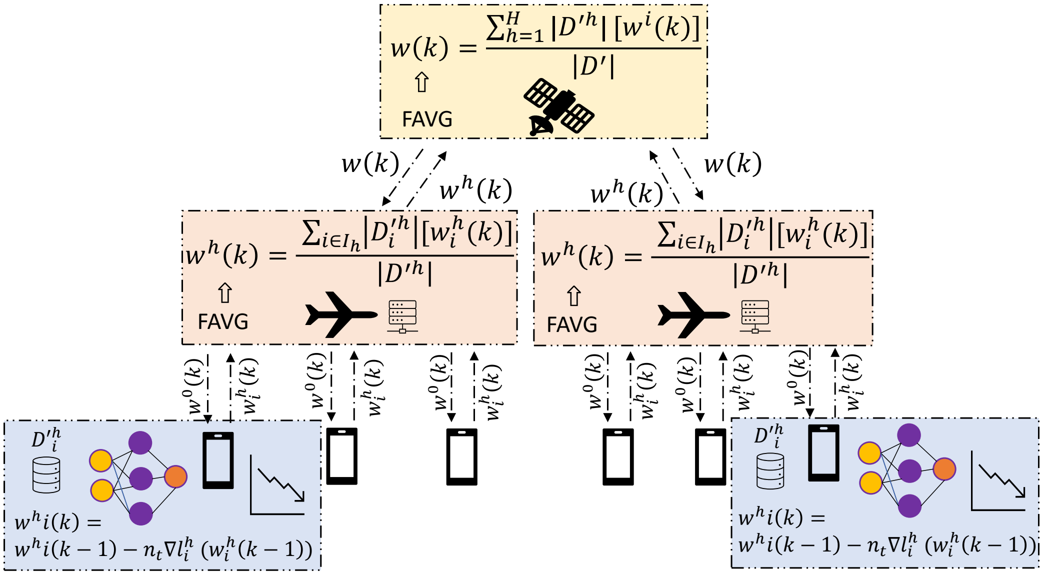
$w(k) = \frac{\sum_{h=1}^{H} |D'^h| \, [w^i(k)]}{|D'|}$
⇧
FAVG
$w(k)$
$w^h(k)$
$w^h(k)$
$w(k)$
$w^h(k) = \frac{\sum_{i \in I_h} |D_i'^h| [w_i^h(k)]}{|D'^h|}$
⇧
FAVG
$w^h(k) = \frac{\sum_{i \in I_h} |D_i'^h| [w_i^h(k)]}{|D'^h|}$
⇧
FAVG
$w^0(k)$
$w_i^h(k)$
$D_i'^h$
$w^h i(k) =$
$w^h i(k-1) - n_t \nabla l_i^h \, (w_i^h(k-1))$
$D_i'^h$
$w^h i(k) =$
$w^h i(k-1) - n_t \nabla l_i^h \, (w_i^h(k-1))$

Figure 2: HierFL-PCC architecture for heterogeneous AANs.

the GTs through the edge server. To address these problems of the existing FL and combine the advantages of cloud-based and edge-based FL into one system, researchers introduced a HierFL framework that enhances the training performance by accessing large data samples from the cloud while reducing costly communications with the cloud, supplementing efficient GT-edge updates (Liu et al., 2020). Hence, we leveraged the proposed HierFL algorithm for proactive content-caching in a heterogeneous AAN. The details of the proposed HierFL-PCC scheme are explained in the following section.

4.2. Hierarchical FL-based Proactive Content-Caching

We considered a heterogeneous AAN comprising a hierarchical model of the LEO satellite, multiple HAPs, and several GTs. The GTs send content requests to the connected HAPs. In the proposed caching scheme, each HAP actively stores a set of content within its ACS. Upon receiving a content request, the HAP checks the content in its local cache and directly delivers it to the GT if it was cached. In the event of a cache miss, the HAP fetches the requested content from the LEO satellite and delivers it to the GT, increasing the content delivery delay. Thus, to maximize the cache efficiency on HAPs, we introduced the HierFL-PCC scheme that predicts content popularity and proactively caches the most frequently requested content in the HAPs.

Algorithm 2 Hierarchical federated learning proactive content-caching prediction in heterogeneous aerial access networks

```
procedure HIERARCHICAL FEDERATED AVERAGING
    Initialized all GTs with parameters: $w$
    for epochs k = 1, . . ., $K$ do
        for each GT i = 1, . . ., $\mathcal{I}$ do
            Update $w_i^h \longleftarrow w_i^h(k-1) - \eta_k \nabla l_i^h(w_i^h(k-1))$
            if $k|k_1 = 0$ then
                for each HAP $h$ = 1, . . ., H do
                    $w^h(k) \longleftarrow HAPAggregation(w_i^h(k))_{i \in \mathcal{I}_h^t}$
                    if $k|k_1k_2 \neq 0$ then
                        for each GT $i \in \mathcal{I}_h^t$ do
                            Update $w_i^h(k) \longleftarrow w^h(k)$
                        end for
                    end if
                end for
            end if
            if $k|k_1k_2 = 0$ then
                $w(k) \longleftarrow LEOAggregation(w^h(k))_{h=1}^H$
                for each GT $i$ = 1, . . ., $\mathcal{I}$ do
                    Update $w_i^h(k) \longleftarrow w(k)$
                end for
            end if
        end for
    end for
end procedure
function HAPAGGREGATION$(h, w_i^h(k))_{i \in \mathcal{I}_h^t})$
    $w^h(k) \longleftarrow \frac{\sum_{i \in \mathcal{I}_h^t} |{D'}_i^h|[w_i^h(k)]}{|{D'}^h|}$
    return $w^i(k)$
end function
function LEOAGGREGATION$(h, w^h(k))_{h=1}^H)$
    $w(k) \longleftarrow \frac{\sum_{h=1}^H |{D'}^h[w^i(k)]}{|D'|}$
    return $w(k)$
end function
```

The proposed HierFL-PCC architecture consists of a virtual central LEO satellite (enabled by SDN technology), v, acting as a cloud, $\mathcal{H}$ HAP servers with disjoint GTs set $\{\mathcal{I}_h^t\}_{h=1}^H$ as edge devices, and $\mathcal{I}_h^t$ GTs with distributed datasets $\{D_i'^h\}_{i=1}^{\mathcal{I}_h^t}$, as depicted in Fig. 2. Each edge parameter server, h, aggregates model updates from its GTs, where the aggregated dataset under HAP h is denoted by D'^h. The proposed HierFL-PCC architecture extends the hierarchical FAVG algorithm at two levels (i.e., edge parameter server at HAP and cloud parameter server at LEO satellite) (Liu et al., 2020). In the hierarchical FAVG learning algorithm, each edge parameter server aggregates its GT models after every κ_1 local updates on each GT. Then, after every κ_2 HAP aggregation instances, the cloud parameter server at LEO aggregates all HAP models; thus, the communication with the LEO happens every $\kappa_1\kappa_2$ local updates on GTs and HAPs. For instance, HAP h aggregates local models from $\mathcal{I}_h^t$ GTs every κ_1 local updates on each GT (e.g., $k|\kappa_1 = 0$ and $k|\kappa_1\kappa_2 \neq 0$), resulting in a partially updated model. In the next step, the partially aggregated model is sent back to the GTs, where GTs perform local model training until $k|\kappa_1 \neq 0$. This GT-HAP training process continues until $k|\kappa_1\kappa_2 = 0$; therefore, after every κ_2 edge model aggregation instances at each HAP h, the cloud parameter server located at the LEO aggregates all partially updated HAP models. Thus, communication with the LEO occurs every $\kappa_1\kappa_2$ local updates. Compared to cloud-based FL, HierFL reduces communication costs with the LEO satellite cloud while still accessing more training data. The parameters of the local model updates for HAPs with a disjoint set of GTs $\{\mathcal{I}_h^t\}_{h=1}^{H}$ at the LEO satellite are denoted by $w_i^h(k)$. Then, the evolution of $w_i^h(k)$ in HierFL is described below (Liu et al., 2020):

$$w_i^h(k) = \begin{cases} w_i^h(k-1) - \eta_t \nabla l_i^h(w_i^h(k-1)) & k|\kappa_1 \neq 0 \\[2em] \frac{\sum_{i \in \mathcal{I}_h} |D_i'^h| [w_i^h(k-1) - \eta_k \nabla L_i^h(w_i^h(k-1))]}{|D'^h|} & \begin{matrix} k|\kappa_1 = 0 \\ k|\kappa_1\kappa_2 \neq 0 \end{matrix} \\[2em] \frac{\sum_{i=1}^{\mathcal{I}} |D'^h| [w_i^h(k-1) - \eta_k \nabla L_i^h(w_i^h(k-1))]}{|D'|} & k|\kappa_1\kappa_2 = 0 \end{cases} \tag{23}$$

where $w_i^h(k-1) - \eta_t \nabla l_i^h(w_i^h(k-1))$ represents the HierFL parameter of the GT, $\frac{\sum_{i \in \mathcal{I}_h} |D_i'^h| [w_i^h(k-1) - \eta_k \nabla L_i^h(w_i^h(k-1))]}{|D'^h|}$ indicates the HierFL parameter of the GT at the edge parameter server h, and $\frac{\sum_{i=1}^{\mathcal{I}} |D'^h| [w_i^h(k-1) - \eta_k \nabla L_i^h(w_i^h(k-1))]}{|D'|}$ represents the HierFL parameter of the GT at the LEO satellite, v. The training process repeats until the model reaches the desired accuracy.

We formally present the proposed HierFL-PCC scheme for a heterogeneous AAN in Algorithm 2. First, the algorithm randomly initializes the GTs with the parameters (w_0). Next, the local model updates at the GTs are computed in parallel using the DNN-MLR method (Lines 4 and 5), and the updated parameters are uploaded to the HAPs. The more frequent communications with the edge parameter server located at HAP h (i.e., fewer local updates κ_1) can accelerate the training process when the communication frequency with the cloud parameter server at the LEO satellite is fixed (i.e., $\kappa_1\kappa_2$ is fixed). Then, the edge parameter servers aggregate uploaded models at the HAP h using the hierarchical FAVG algorithm (Lines 7 and 8). After the edge aggregation, an updated (partial) model is transmitted to the GTs and then used by each GT in the coverage area of h to perform local updates (i.e., $k|\kappa_1 \neq 0$) (lines 10–11). After the number of edge aggregations (i.e., $\kappa_1\kappa_2 = 0$), the model is aggregated globally at the cloud parameter server in Lines 17 to 19. Finally, Lines 25 to 32 describe the HAP aggregation and LEO aggregation methods, respectively. The process repeats until the model reaches the desired accuracy. The HierFL algorithm can be deployed in a distributed computing environment consisting of GTs, HAPs, and the LEO satellite cloud in this study. It is a hierarchical model comprising GTs, HAPs, and the LEO satellite that combines the advantages of cloud-based and edge-based FL systems. Specifically, GTs perform local updates on their devices, edge parameter servers at HAPs aggregate GTs models, and the LEO satellite cloud aggregates the HAP models. As soon as the learning process completes, the predicted popular contents are cached based on the caching policy $\mathcal{P}_h$ in HAP h.

4.3. Complexity Analysis

The time complexity of the HierFL-PCC algorithm primarily depends on the computational complexity of training the DNN-MLR model, which serves as a function approximator. During the DNN-MLR model training, data is organized into batches, and the time complexity per data sample is influenced by the number of layers in the model denoted as h_L, and the number of neurons within each layer, represented as n_L. The number of training iterations required for the neural network training depends on the maximum number of training epochs, denoted as K, and the batch size, referred to as β, as shown in Algorithm 1. Thus, according to Yu et al. (2020), the time complexity for updating the model parameters at the GT i can be estimated as $O(\,K(\,\beta \cdot h_L \cdot n_L^2\,)\,)$, which directly characterizes the model parameters, denoted as M_P, for the GT i. Furthermore, the time complexity for updating the model parameters across all GTs in $\mathcal{I}$ can be estimated as $\mathcal{O}(\,\mathcal{I} \cdot M_P\,)$. In the proposed HierFL-PCC scheme, the computational complexity transi-

Table 2: Simulation Parameters

Parameters	**Value**
Number of GTs, $\|I^t\|$	3^3
Number of GTs under one HAP, $\|I^t_h\|$	3^2
Number of HAPs, $\|H^t\|$	3^1
HAP altitude, A_H	20 km
Coverage radius of the HAP	46 km
LEO satellite height, A_v	500 km
Distance between the HAP and GT, $d_{h,i}[t]$	$[20, 50]$
Transmission power, $\mathcal{P}_{h,i}[t]$	2 W
Downlink frequency, $f_{h,i}[t]$	12 GHz
Bandwidth, $\mathcal{B}_h$	10 MHz
Antenna power gain at the HAP and GT, g_h, g_i	30 dBi
Transmission power, $\mathcal{P}_{v,h}[t]$	2 W
Downlink frequency, $f_{v,h}[t]$	15 GHz
Bandwidth, $\mathcal{B}_v$	20 MGz
Training epochs, K	5x10^2
Hidden layers	3
Neurons per hidden layer	32,24,16
Activation function for the hidden layer	ReLU
Activation function for the output layer	ReLU
Optimizer	Adam
Learning rate α	$1e-3$
Batch size, β	10^4
Update frequencies, κ_1, κ_2	10, 5
Size of the content, s	1 Mb

tions beyond GTs' local updates as the HierFL-PCC algorithm introduces two aggregation levels: edge parameter server at HAP h and cloud parameter server at a LEO satellite v. Specifically, each HAP h aggregates its GT model parameters after every κ_1 local updates on each GT and after every κ_2 HAP aggregations, the LEO satellite v aggregates all HAP models. The time complexity for model aggregations at HAPs and LEO satellite can be estimated respectively as $\mathcal{O}(\frac{K}{\kappa_1}(\mathcal{H}\cdot\mathcal{I}^t_h\cdot M_P))$ and $\mathcal{O}(\frac{K}{\kappa_1\kappa_2}(\mathcal{H}\cdot M_P))$. Thus, the time complexity of the proposed HierFL-PCC algorithm can be expressed as $\mathcal{O}(\mathcal{I}\cdot M_P) + \mathcal{O}(\frac{K}{\kappa_1}(\mathcal{H}\cdot\mathcal{I}^t_h\cdot M_P)) + \mathcal{O}(\frac{K}{\kappa_1\kappa_2}(\mathcal{H}\cdot M_P))$.

In terms of considering the potential deployment scenarios, we envision that HierFL-PCC can be seamlessly integrated into existing networks. Due to its hierarchical structure, it can accommodate different network sizes and architectures, making it suitable for real-world applications such as content delivery networks, edge networks, and satellite communications. These distinctive characteristics enable HierFL-PCC as an effective solution for enhancing cache efficiency across different network environments.

5. Numerical Simulations

This section evaluates the proposed HierFL-PCC scheme based on the HierFL model among the GTs, HAPs, and the LEO satellite to collaboratively predict the popular content for GTs with privacy preservation. First, the simulation setup and dataset are described. Subsequently, the performance evaluation of the HierFL model is explained. We then compared the proposed HierFL-PCC scheme with baseline caching schemes regarding caching efficiency, delivery delay, and training overhead.

5.1. Simulation Setup and Dataset

We simulated a heterogeneous AAN in remote areas consisting of an LEO satellite, three HAPs ($\mathcal{H} = 3$) under the coverage of an LEO satellite, and several GTs distributed within the coverage area of the HAPs and request contents. The number of GTs under the coverage of each HAP varies from one to nine. The LEO satellite, deployed at an altitude of 500 km is moving periodically and may depart from the connections with HAPs under coverage over time. However, when the data relayed by HAPs arrive at the LEO satellite, the data can be guaranteed back to Earth through inter-satellite links between LEO satellites. The maximum coverage radius of the HAP is set to 46 km, and the downlink frequency is set to 12 GHz. The transmission power of the HAP is 2 W, and the bandwidth is assumed to be 10 MHz.

The proposed HierFL-PCC scheme uses PyTorch 2.0 with Python 3.11.3 on a computer powered by an Intel (R) Core (TM) i7-10700F CPU with a frequency of 2.9 GHz and RAM of 16 GB. The graphics card was an Nvidia GeForce GTX 1050Ti with 12 GB of total memory. Prior to running the simulations, we developed a fully connected DNN-MLR model with three hidden layers, and the number of neurons in the corresponding hidden layers was set to 32, 24, and 16. We used the rectified linear unit activation function for the output and other layers. The LR for the DNN was set to $1e-3$, and the training epochs were set to 5x10^2.

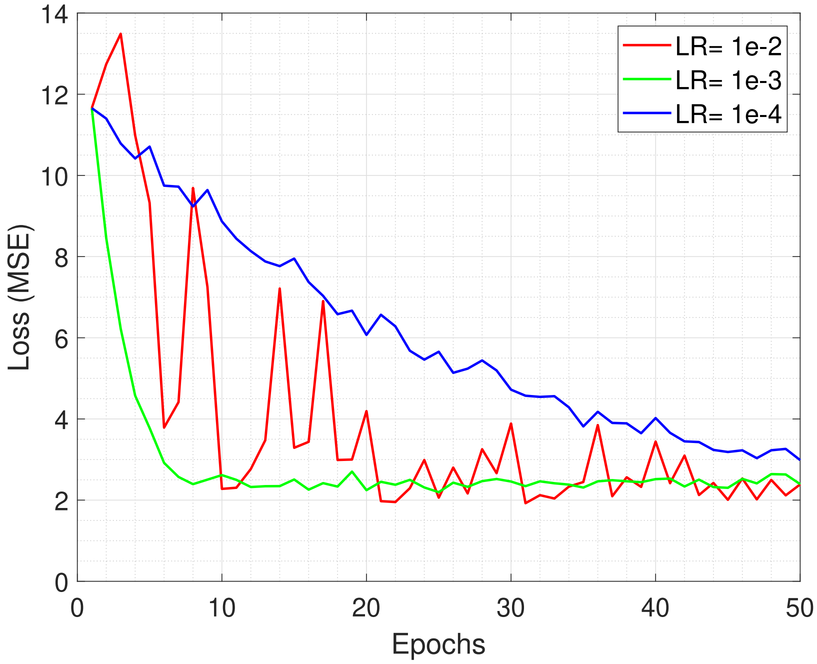

Figure 3: Loss function value over the growing number of epochs for various learning rates.

The proposed HierFL-PCC scheme is evaluated using the real-world dataset MovieLens (Harper and Konstan, 2015). The MovieLens 100K dataset contains 100,000 ratings on 1,682 movies from 943 users. This dataset provides the contextual information of users, such as gender, age, and occupation. The rating scale is 0 to 5, where each user rates 20 movies. To simulate the content requests, we follow an approach where the rated movies represent the files requested by the GTs, where each movie rating corresponds to a streaming request from the GT (Müller et al., 2016; Li et al., 2016; Yu et al., 2018). It is notable that while the dataset is distributed among the GTs to capture diverse user preferences, the DNN-MLR model training occurs exclusively at the GTs, leveraging the contextual information and content ratings to predict popular contents. The model weights derived from the GTs' DNN-MLR training are shared among the HAPs and LEO satellites for HierFL model training while the MovieLens dataset is not shared among the HAPs or LEO satellites, which ensures the integrity of the dataset as the contents of the dataset remain unchanged during the simulation. Table 2

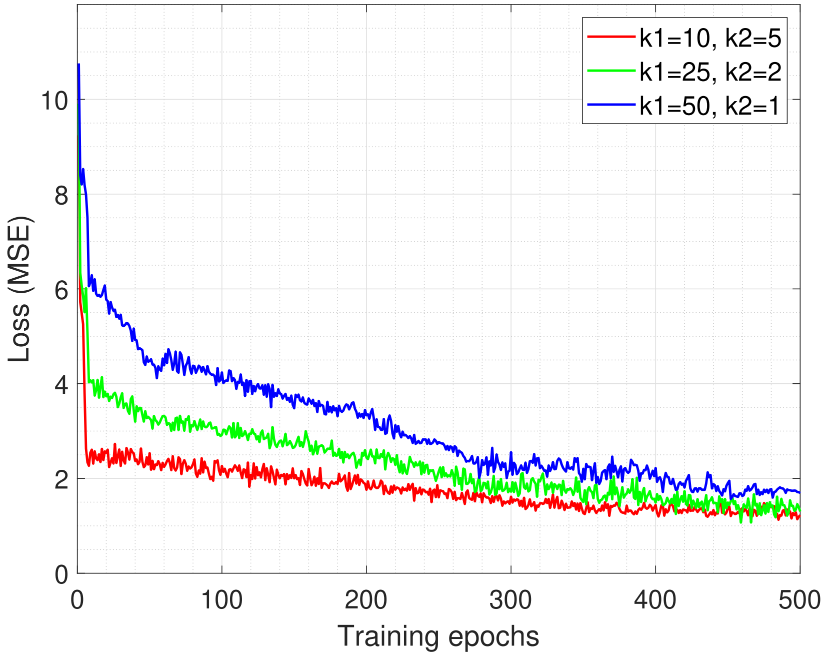

Figure 4: Loss value over training epochs for values of k_1 and k_2.

describes the parameters and values used during the simulation. The proposed HierFL-PCC scheme was compared to four baseline caching schemes, which have been employed as benchmarks in recent related works focusing on proactive caching (Yu et al., 2018, 2020; Cui et al., 2020; Feng et al., 2023), as detailed below.

- *Random caching scheme*: In a random caching scheme, content stored in the cache is randomly selected by the HAPs.

- *Least frequently used (LFU) caching scheme*: In an LFU caching scheme, the LFU cache content is removed when the capacity is reached.

- *Least recently used (LRU) caching scheme*: In an LRU caching scheme, the least recently accessed or used content is removed when the cache is full.

- *Cloud-based FAVG caching scheme*: We implemented a cloud-based FAVG method using the LEO satellite as a central cloud for model

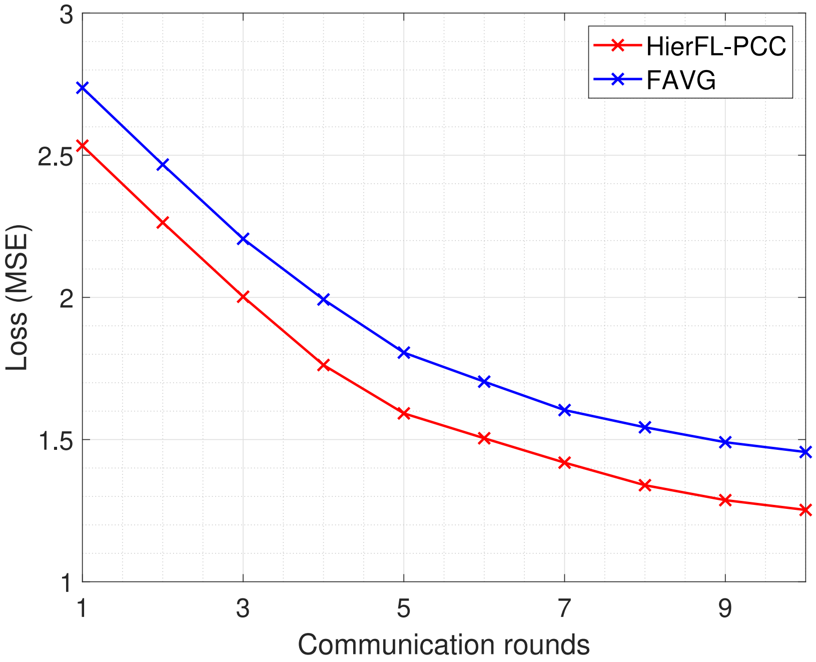

Figure 5: Loss comparison of the HierFL-PCC scheme and existing cloud-based FL using the FAVG method over communication rounds

aggregation. It employs the same DNN-based predictive approach as HierFL-PCC, caching popular contents to optimize cache performance. It allows to evaluate proposed caching scheme against FL-based proactive content-caching scheme.

- *Proposed caching scheme (HierFL-PCC)*: The proposed HierFL-PCC scheme leverages a DNN-MLR-based HierFL model to predict the popularity of contents requested by GTs and maximizes the cache efficiency and reduces content delivery delay and training overhead.

5.2. Performance Evaluations

This section studies the performance of the proposed HierFL-PCC scheme in terms of cache efficiency, delivery delay, and training overhead reduction. First, we perform simulations to evaluate the performance of the HierFL method. The results are plotted in Figs. 3 and 4. Figure 3 demonstrates the value of the loss function over the increasing number of epochs for different

LR values. The Adam optimizer employs the LR of the fully connected DNN to update the weight parameter w of the DNN to make the prediction. As the number of epochs approaches 10, the loss (MSE) achieved with an LR = $1e-3$ is minimized and continues to be lower throughout the remaining number of epochs than those with an LR = $1e-2$ and $1e-4$. Therefore, we selected the training result with LR = $1e-3$ to evaluate the remaining performance comparison of the proposed HierFL-PCC scheme. Figure 4 illustrates the convergence performance of the proposed HierFL-PCC scheme in terms of the value of loss (MSE) function over training epochs. Based on the convergence analysis provided in Liu et al. (2020), we selected the update frequencies such that the product of κ_1 and κ_2 is constant to maintain a fixed number of local updates between the two LEO aggregations. Therefore, we fixed the communication frequency with the LEO satellite cloud at 50 local epochs (i.e., $\kappa_1\kappa_2 = 50$) and changed the value of κ_1 and κ_2. Then, we have three combinations of the values of κ_1 and κ_2 for the product of κ_1 and κ_2 to be fixed (i.e., $\kappa_1\kappa_2 = 50$). The lowest loss value can be reached when κ_1 = 10, minimizing the loss with fewer local computations. It also reveals that frequent communications with HAP for partial model aggregation accelerate the training process when the communication frequency with the LEO satellite for global model aggregation is fixed.

Next, simulations were conducted using the HierFL-PCC scheme and cloud-based FAVG method to evaluate the loss performance. The result is plotted in Fig. 5. The communication rounds varied from one to 10. Figure 5 reveals that the loss value using the FAVG method is higher than that of HierFL, even after the first communication round. In contrast, HierFL achieves superior loss function improvement during the early communication rounds and continues its performance throughout the considered communication rounds. HierFL outperforms the cloud-based FAVG method due to fewer local computations and more frequent HAP aggregation occurrences. Cache efficiency is a performance metric for the proposed HierFL-PCC scheme, measuring the ratio of cache hits to the total number of content requests on the cache. In Fig. 6, we compared the cache efficiency over the communication rounds for 100%, 60%, and 20% GT participation. More communication rounds are needed to achieve a suboptimal cache efficiency of 30% with a 20% GT participation, whereas it only takes only five communication rounds to reach the suboptimal cache efficiency of 30% with full (100%) GT participation. However, a cache efficiency with 60% GT participation is still considered reasonable compared to that of 100% GT participation.

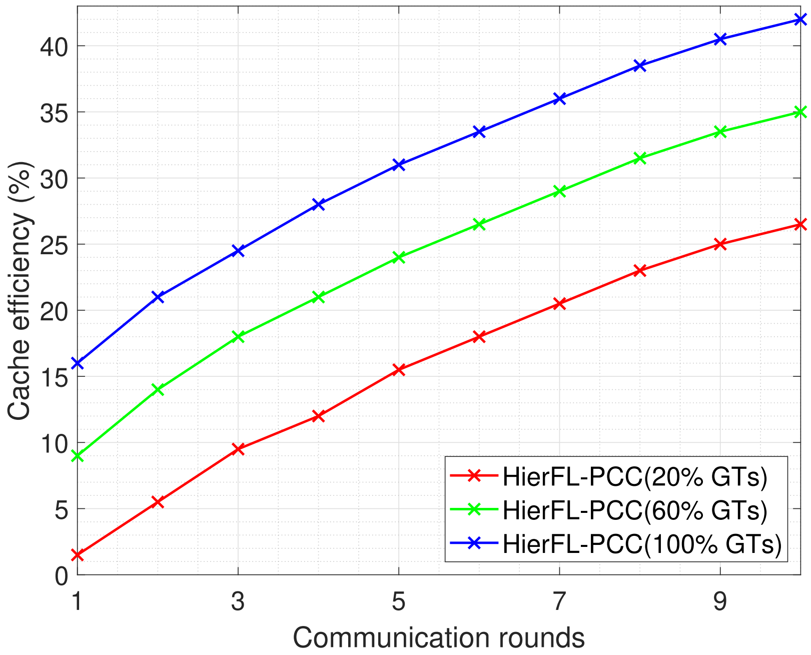

Figure 6: Caching efficiency comparison of the proposed HierFL-PCC for 100%, 60%, and 20% ground terminal participation over the communication rounds.

Figure 7 depicts the cache efficiency for varying cache sizes from 50 to 300 content items. As depicted in Fig. 7, the cache efficiency of all the comparison caching schemes increases as the cache size increases. The results indicate that the proposed HierFL-PCC scheme achieves a cache efficiency of 13%, 43%, 178%, and 457% compared to the cloud-based FAVG, LFU, LRU, and random caching schemes, respectively. The HierFL-PCC scheme outperforms the cloud-based FAVG method because it achieves superior loss function improvement due to a better computation-communication trade-off. However, the cloud-based FAVG scheme indicates better cache performance than the LFU, LRU, and the random caching schemes because the cloud-based FAVG scheme employs the DNN to learn contextual information between GTs and content requests. In contrast, the LFU , LRU, and random caching schemes do not observe past requests. LFU, considering frequency, and LRU, factoring in the recency, both reduce caching efficiency compared to cloud-based FAVG and HierFL-PCC schemes. LFU struggles to adapt to changing content requests, while LRU, lacks the dynamic adaptability demonstrated by

HierFL-PCC. Figure 8 depicts the average delay in delivering the requested

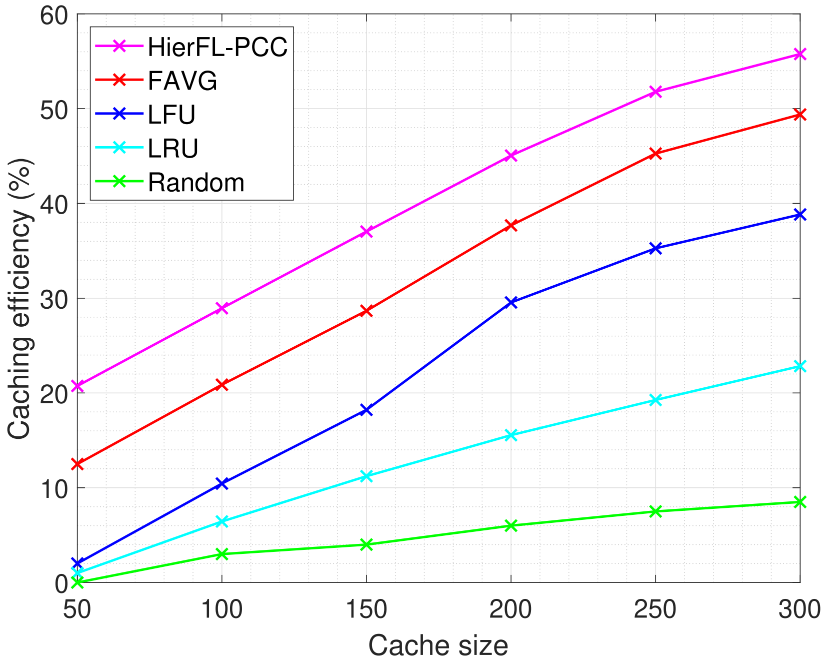

Figure 7: Caching efficiency comparison of the proposed HierFL-PCC and existing schemes over cache size.

content for each GT. The average delay comparison is made against 9, 18, and 27 GTs. As depicted in Fig. 8, the average delay is 10%, 20%, 31%, and 37% less than that in the cloud-based FAVG, LFU, LRU, and random caching schemes, respectively, because the HierFL-PCC scheme improves cache performance and an increase in cache performance directly reduces the delay associated with content delivery. Figure 9 presents the overhead reduction regarding the volume of data transmitted in each communication round and the total number of communication rounds. We compared the communication cost of the HierFL-PCC with that of the cloud-based FAVG, where 10 communication rounds were considered. As depicted in Fig. 9, the communication cost of the proposed HierFL is 89% lower than that of the traditional cloud-based FAVG, explaining the efficiency gains of HierFL compared to the cloud-based FAVG because HierFL significantly reduces costly cloud communications, supplemented by efficient edge communications.

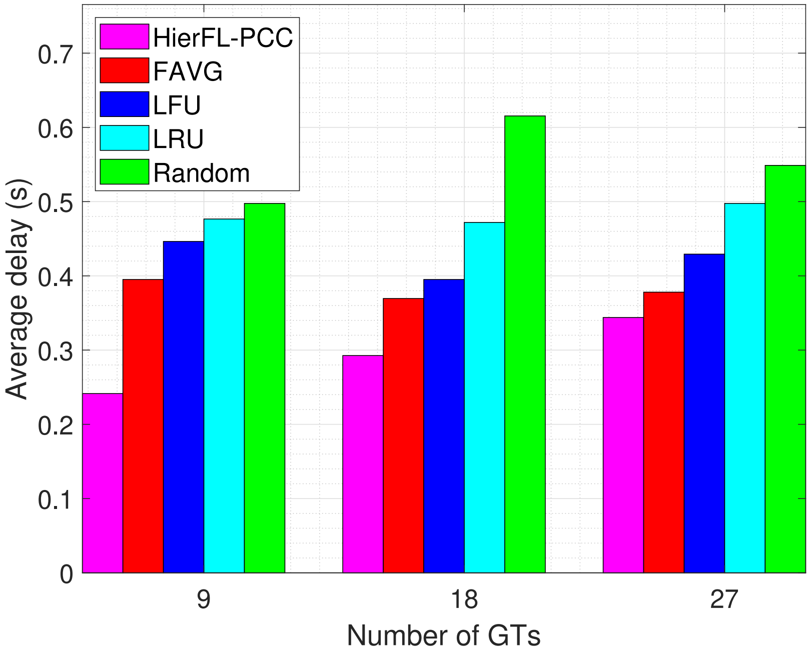

Figure 8: Average delay comparison of the proposed HierFL-PCC and existing caching schemes with varying numbers of ground terminals.

6. Conclusion

The current mobile access networks prioritize high throughput and low delay but lack coverage in remote areas. To address this issue, heterogeneous AANs serve as a complementary solution to overcome the limitations of ground-based networks and provide content requests in remote and underserved areas. This paper introduced the privacy-preserving intelligent proactive content-caching scheme HierFL-PCC. The proposed scheme leverages the HierFL model involving GTs, HAPs, and the LEO satellite constellation in heterogeneous AANs. First, we analyzed the heterogeneous AAN system from multiple perspectives, such as the transmission model, caching model, delay model, privacy preservation, and overhead calculation. Then, we proposed the HierFL-PCC scheme to predict dynamic content requests from GTs to maximize caching efficiency while preserving the privacy of GTs. The requested content is directly accessed from the HAPs instead of the LEO satellite, reducing the delay and overhead associated with content delivery.

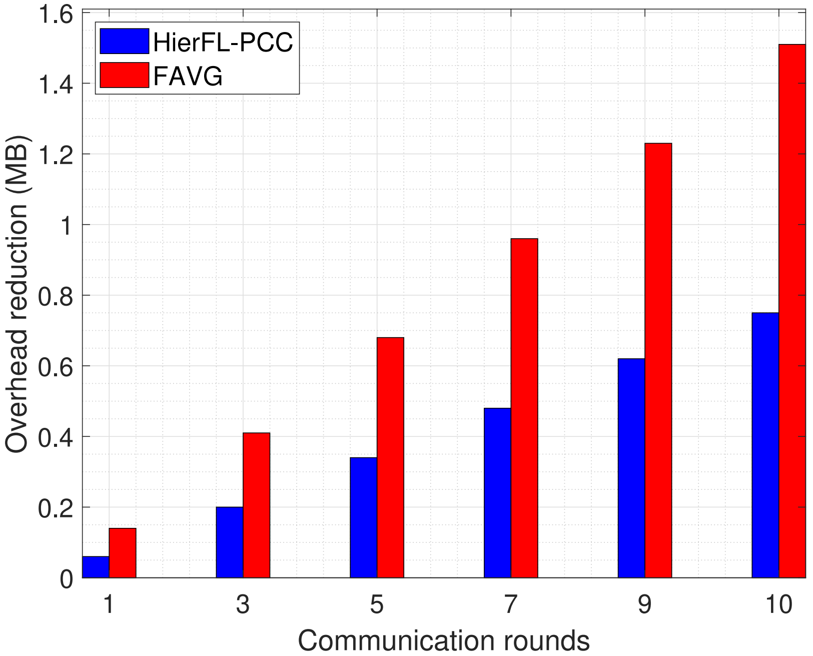

Figure 9: Overhead reduction comparison of the proposed HierFL-PCC and existing cloud-based FAVG method.

In addition, HierFL-PCC reduces the communication overhead involved in model training by reducing the costly LEO-HAP communications and supplementing efficient HAP-GT communications in model training. Furthermore, the numerical experiments revealed that HierFL-PCC outperforms the baseline caching schemes regarding cache efficiency, content delivery delay, and training overhead. In future work, we intend to evaluate the performance of the proposed HierFL-PCC scheme across different network environments and explore the challenges of proactive content-caching, taking into account factors such as network congestion, varying user densities, and heterogeneous device capabilities.

CRediT author statement

Arooj Masood: Conceptualization, Methodology, Software, Formal analysis, Data Curation, Visualization, Writing - Original Draft, Writing - Review & Editing **Nhu-Ngoc Dao**: Conceptualization, Writing - Review &

Editing **Sungrae Cho**: Writing - Review & Editing, Supervision, Project administration ,

References

Bera, A., Misra, S., Chatterjee, C., 2020. Qoe analysis in cache-enabled multi-uav networks. IEEE Transactions on vehicular technology 69, 6680–6687.

Chen, M., Mozaffari, M., Saad, W., Yin, C., Debbah, M., Hong, C.S., 2017. Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience. IEEE Journal on Selected Areas in Communications 35, 1046–1061.

Chen, M., Saad, W., Yin, C., 2019. Liquid state machine learning for resource and cache management in lte-u unmanned aerial vehicle (uav) networks. IEEE Transactions on Wireless Communications 18, 1504–1517.

Cui, L., Su, X., Ming, Z., Chen, Z., Yang, S., Zhou, Y., Xiao, W., 2020. Creat: Blockchain-assisted compression algorithm of federated learning for content caching in edge computing. IEEE Internet of Things Journal 9, 14151–14161.

Dao, N.N., Pham, Q.V., Tu, N.H., Thanh, T.T., Bao, V.N.Q., Lakew, D.S., Cho, S., 2021. Survey on aerial radio access networks: toward a comprehensive 6g access infrastructure. IEEE Communications Surveys & Tutorials 23, 1193–1225.

Ericsson, 2023. Mobile data traffic outlook. Ericsson Mobility Report. URL: https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/mob

Fadlullah, Z.M., Kato, N., 2020. Hcp: Heterogeneous computing platform for federated learning based collaborative content caching towards 6g networks. IEEE Transactions on Emerging Topics in Computing .

Feng, B., Feng, C., Feng, D., Wu, Y., Xia, X.G., 2023. Proactive content caching scheme in urban vehicular networks. arXiv preprint arXiv:2305.07584 .

Harper, F.M., Konstan, J.A., 2015. The movielens datasets: History and context. Acm transactions on interactive intelligent systems (tiis) 5, 1–19.

Jia, X., Lv, T., He, F., Huang, H., 2017. Collaborative data downloading by using inter-satellite links in leo satellite networks. IEEE Transactions on Wireless Communications 16, 1523–1532.

Jia, Z., Sheng, M., Li, J., Zhou, D., Han, Z., 2020. Joint hap access and leo satellite backhaul in 6g: Matching game-based approaches. IEEE Journal on Selected Areas in Communications 39, 1147–1159.

Jiang, B., Yang, J., Xu, H., Song, H., Zheng, G., 2018. Multimedia data throughput maximization in internet-of-things system based on optimization of cache-enabled uav. IEEE Internet of Things Journal 6, 3525–3532.

Kang, S.W., Thar, K., Hong, C.S., 2020. Unmanned aerial vehicle allocation and deep learning based content caching in wireless network, in: 2020 International Conference on Information Networking (ICOIN), IEEE. pp. 793–796.

Kurt, G.K., Khoshkholgh, M.G., Alfattani, S., Ibrahim, A., Darwish, T.S., Alam, M.S., Yanikomeroglu, H., Yongacoglu, A., 2021. A vision and framework for the high altitude platform station (haps) networks of the future. IEEE Communications Surveys & Tutorials 23, 729–779.

Li, S., Xu, J., Van Der Schaar, M., Li, W., 2016. Popularity-driven content caching, in: IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE. pp. 1–9.

Li, X., Huang, K., Yang, W., Wang, S., Zhang, Z., 2019. On the convergence of fedavg on non-iid data. arXiv preprint arXiv:1907.02189 .

Liu, L., Zhang, J., Song, S., Letaief, K.B., 2020. Client-edge-cloud hierarchical federated learning, in: ICC 2020-2020 IEEE International Conference on Communications (ICC), IEEE. pp. 1–6.

Maale, G.T., Sun, G., Kuadey, N.A.E., Kwantwi, T., Ou, R., Liu, G., 2023. Deepfesl: Deep federated echo state learning-based proactive content caching in uav-assisted networks. IEEE Transactions on Vehicular Technology .

Masood, A., Lakew, D.S., Cho, S., 2020. Learning based content caching for wireless networks, in: 2020 International Conference on Information Networking (ICOIN), IEEE. pp. 74–75.

Masood, A., Lakew, D.S., Dao, N.N., Cho, S., et al., 2023. A review on ai-enabled content caching in vehicular edge caching and networks, in: 2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), IEEE. pp. 713–717.

Masood, A., Nguyen, T.V., Cho, S., 2021a. Deep regression model for videos popularity prediction in mobile edge caching networks, in: 2021 International Conference on Information Networking (ICOIN), IEEE. pp. 291–294.

Masood, A., Nguyen, T.V., Truong, T.P., Cho, S., 2021b. Content caching in hap-assisted multi-uav networks using hierarchical federated learning, in: 2021 International Conference on Information and Communication Technology Convergence (ICTC), IEEE. pp. 1160–1162.

McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A., 2017. Communication-efficient learning of deep networks from decentralized data, in: Artificial intelligence and statistics, PMLR. pp. 1273–1282.

Müller, S., Atan, O., van der Schaar, M., Klein, A., 2016. Context-aware proactive content caching with service differentiation in wireless networks. IEEE Transactions on Wireless Communications 16, 1024–1036.

Ndikumana, A., Tran, N.H., Kim, K.T., Hong, C.S., et al., 2020. Deep learning based caching for self-driving cars in multi-access edge computing. IEEE Transactions on Intelligent Transportation Systems 22, 2862–2877.

Qiao, G., Leng, S., Maharjan, S., Zhang, Y., Ansari, N., 2019. Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks. IEEE Internet of Things Journal 7, 247–257.

Qiu, J., Grace, D., Ding, G., Zakaria, M.D., Wu, Q., 2019. Air-ground heterogeneous networks for 5g and beyond via integrating high and low altitude platforms. IEEE Wireless Communications 26, 140–148.

Wang, Y., Feng, C., Zhang, T., Liu, Y., Nallanathan, A., 2020. Qoe based network deployment and caching placement for cache-enabling uav networks, in: ICC 2020-2020 IEEE International Conference on Communications (ICC), IEEE. pp. 1–6.

Wang, Y., Fu, S., Yao, C., Zhang, H., Yu, F.R., 2023. Caching placement optimization in uav-assisted cellular networks: A deep reinforcement learning based framework. IEEE Wireless Communications Letters .

Xu, X., Zeng, Y., Guan, Y.L., Zhang, R., 2018. Overcoming endurance issue: Uav-enabled communications with proactive caching. IEEE Journal on Selected Areas in Communications 36, 1231–1244.

Yu, Z., Hu, J., Min, G., Lu, H., Zhao, Z., Wang, H., Georgalas, N., 2018. Federated learning based proactive content caching in edge computing, in: 2018 IEEE Global Communications Conference (GLOBECOM), IEEE. pp. 1–6.

Yu, Z., Hu, J., Min, G., Zhao, Z., Miao, W., Hossain, M.S., 2020. Mobility-aware proactive edge caching for connected vehicles using federated learning. IEEE Transactions on Intelligent Transportation Systems .

Zhang, T., Wang, Y., Liu, Y., Xu, W., Nallanathan, A., 2020. Cache-enabling uav communications: Network deployment and resource allocation. IEEE Transactions on Wireless Communications 19, 7470–7483.

Zhao, L., Li, H., Lin, N., Lin, M., Fan, C., Shi, J., 2021. Intelligent content caching strategy in autonomous driving toward 6g. IEEE Transactions on Intelligent Transportation Systems .