This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2020.2986238, IEEE Transactions on Emerging Topics in Computing



# HCP: Heterogeneous Computing Platform for Federated Learning Based Collaborative Content Caching Towards 6G Networks

Zubair Md. Fadlullah, *Senior Member, IEEE* and Nei Kato, *Fellow, IEEE.*

**Abstract**—A heterogeneous computing architecture is essential to facilitate intelligent network traffic control for a joint computation, communication, and collaborative caching optimization in 6G networks to provide stringent Quality of Experience (QoE) guarantees. In this paper, we consider a 6G integrated aerial-terrestrial network model where Unmanned Aerial Vehicles (UAVs) and terrestrial Remote Radio Heads (RRHs) jointly serve as heterogeneous Base Stations (hgNBs) of a Cloud Radio Access Network (HCRAN) serving different mobile user (UE) types. We propose a distributed heterogeneous computing platform (HCP) across the UAVs and terrestrial Base Stations (BSs) by utilizing their caching and cooperative communication capabilities. In order to preserve the privacy of the content of the UEs, we propose a 2-stage federated learning algorithm among the UEs, UAVs/BSs, and HCP to collaboratively predict the content caching placement by jointly considering traffic distribution, UE mobility and localized content popularity. An asynchronous weight updating method is adopted to avoid redundant learning transfer in the federated learning. Once the global model is learnt by the HCP, it transfers the learned model to the UEs to facilitate the much desired edge intelligence in the considered 6G tiny cell. The effectiveness of the proposal is evaluated by extensive numerical analysis.

**Index Terms**—6G, collaborative caching, federated learning, edge computing, Unmanned Aerial Vehicle (UAV), Heterogeneous Computing Platform (HCP).

## 1 Introduction

Recently, the Fifth Generation (5G) mobile networks emerged as a solution to meet the exponential demand of bandwidth-intensive applications and services of numerous mobile users and Internet of Things (IoT) devices. As a consequence, mobile data will be the dominant network load by 2022, exceeding 77 exabytes monthly [1] according to Cisco Visual Networking Index. The annual mobile traffic of almost one zettabyte is anticipated to be dominated by feature-rich, high-definition interactive as well as streaming contents as well as other applications such as tactile Internet [2], robotic interactions [3], haptic communication [2], [4], augmented and virtual reality [5], and so forth [6]. This surge of mobile traffic, however, contributes to increasing communication delay while placing a tremendous burden on the backhaul links. To address this issue, content caching has reappeared as an interesting topic for Beyond 5G (also referred to as 6G) networks, particularly in the network edge serviced by heterogeneous base stations or hgNBs. Content caching-enabled Base Stations in 6G [7], [8], [9], [10], [11], [12], [13], [14], [15] will feature a wide deployment of Unmanned Aerial Vehicles (UAVs) or drones acting as flying Base Stations (BSs) [16], which will complement the terrestrial mobile networks [17], [18] to extend their service coverage in ultra-dense tiny cell scenarios. Despite the proliferation of high-performance yet affordable processing and memory technologies, the hgNBs, particularly the flying BSs of 6G small cells cannot have large storage resources for caching, and therefore, it is critical to carry out an intelligent and coordinated caching on the hgNBs along with the participation of mobile users (commonly referred to as User Equipment (UEs)). Since privacy-preserving will appear as a huge issue for maintaining privacy of the UEs roaming in the 6G ultra-dense tiny cells, existing content caching techniques based on explicit user reporting on their content preferences can, no longer, be used. In this paper, we address this issue and propose a collaborative caching based on a two-step federated learning [19] by asynchronously updating the general and specific weight parameters of the learning models locally at UEs as well as hgNBs and globally at a Heterogeneous Computing Platform (HCP). This allows the HCP to accurately predict the appropriate content cache placement at the appropriate hgNBs that may be frequently requested by UEs so as to reduce backhaul link congestion while reducing user delay. It is worth noting that while existing UEs can be, at best, equipped with pre-trained AI models, in 6G networks, edge-AI enabled nodes will proliferate, allowing them to fully exploit our proposed two-step federated learning model [20], [21]. In addition to the proposed method's effectiveness in assuring the UE's content privacy requirement, the mobility and network traffic features are also not revealed to the content server explicitly. The terrestrial and aerial hgNBs exchange only the local models constructed from the mobility and network traffic features of UEs to improve the global model at the HCP for subsequently improving their own local models that may enable seamless content streaming upon handover

- *Z. M. Fadlullah is with the Computer Science Department, Lakehead University, and Thunder Bay Regional Health Research Institute (TBRHRI), Thunder Bay, Ontario, Canada.*
  *E-mail: Zubair.Fadlullah@lakeheadu.ca*
- *N. Kato is with the Graduate School of Information Sciences (GSIS), Tohoku University, Sendai, Japan. Email: kato@it.is.tohoku.ac.jp*

2168-6750 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Univ of Science and Tech Beijing. Downloaded on December 15,2021 at 01:02:31 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2020.2986238, IEEE Transactions on Emerging Topics in Computing



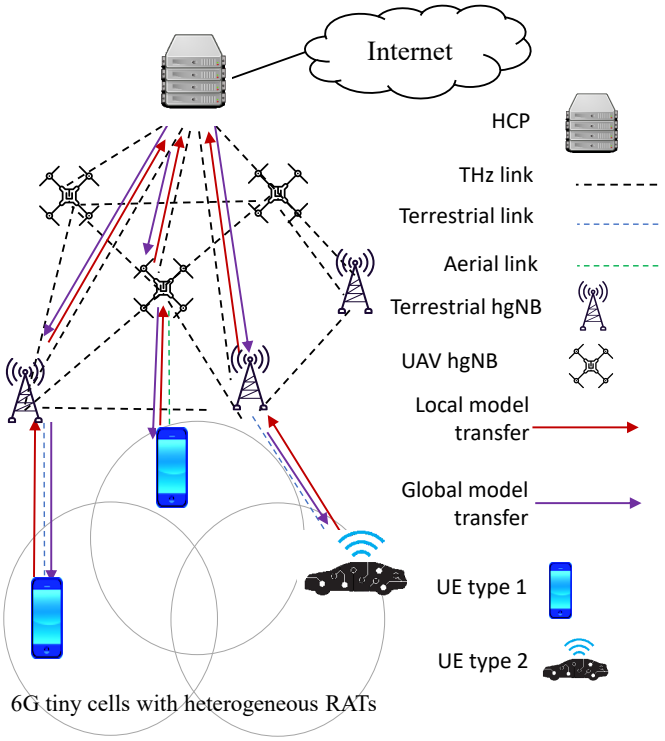



Fig. 1: Considered RAN architecture based on the HCP controlling multiple RATs based on UAV and terrestrial hgNBs.

of UEs to other hgNBs.

The remainder of the paper is structured as follows. The relevant research work are surveyd in Sec. 2. Our considered system model is presented in Sec. 3. In Sec. 4, we provide a formal formulation of the distributed learning need for UEs to preserve their privacy while assigning the most appropriate contents to the hgNB caches to efficiently serve the users. Our proposed two-step federated learning using asynchronous weight update model is presented in Sec. 5. The performance of our proposed approach is evaluated in Sec. 6. Finally, the paper is concluded in Sec. 7.

## 2 Related Work

A number of content recommendation systems appeared in the literature for the emerging 5G and beyond networks to fulfill different criteria [16], [22]. However, these existing recommendation systems do not take into account the privacy preserving condition of mobile users. For example, in [16], Location Based Social Networks (LBSNs) with highly mobile UEs were served by a UAV-based offloading backbone to perform data sensing and relevant data computations in the UAV-mounted cloudlets. The localized cloud computation provided adaptive recommendation in a distributed fashion to reduce computing and traffic load. Because UAVs have limited cache storage, it is essential to place the recommended contents (i.e., the most likely ones to be requested by UEs) in the local cache. Conventional caching schemes [23] perform update of the cache based on First-In-First-Out (FIFO), Least Recently Used (LRU), and Least Frequently Used (LFU) algorithms. These methods are not robust to dynamically varying popularity of contents in the beyond 5G networks. Therefore, renewed interest has arisen to design smart algorithms to dynamically allocate popular contents in the rather limited cache resource of BSs. The dynamic cache assignment algorithms are broadly categorized into types: with and without prior knowledge regarding the popularity distribution probability of the content. The work in [24] used coded caching to improve cache-efficiency based on a prior knowledge of zipf-based content distribution. On the other hand, AI-based methods such as reinforcement learning and proactive, collaborative filtering were proposed in [25] to calculate the popularity of contents in small cells with roaming UEs. The reinforcement learning-based approach observes the demands of cached content and then updates it over fixed time-intervals. The algorithm in [26] developed a coded caching scheme in a terrestrial Base Station (eNB) based on demand-history as well as UEs' context information, density of UEs, and request file time, to evaluate the content popularity using a combinatorial multi-armed bandit formulation.

The aforementioned methods, however, share an inherent shortcoming. Using these methods, the UEs are unable to preserve their privacy because they need to explicitly exchange their content-related data with a centralized server. While distributed learning or transfer learning have been proposed in the literature [27], they are usually dependent on the collaboration of UEs which also force the users to share content-specific data with one another. In order to mitigate the privacy concern of the UEs, a novel proactive content caching method is required in 6G multi-tier UAV and terrestrial hgNBs that we investigate in this paper.

## 3 Considered System Model

Our considered 6G network model, as depicted in Fig. 1, consists of an integrated aerial and terrestrial network with heterogeneous Radio Access Technologies (RATs). A number of UAVs establish drone-cells, along with a number of terrestrial Base Stations (BSs), serve user equipment (UEs). Each UAV (acting as a flying BS) and each ground BS is referred to as a "hgNB" (heterogeneous next generation Node B). On the other hand, the UEs are comprised of high-speed mobile users such as autonomous vehicles as well as pedestrian users and Internet of Things (IoT) devices. A Software Defined Network (SDN) controller is considered to manage the integrated aerial and terrestrial links. For the aerial component, the SDN can programmatically control the flying hgNBs to deploy and manage their changing link quality and topologies. Thus, it allows the aerial hgNBs to adaptively work in different environments of the terrestrial RATs. The SDN controller is based on the Heterogeneous Computing Platform (HCP) which was conceptualized in one of our earlier work [28], as depicted in Fig. 2. The HCP based SDN controller can revolutionize the 6G network to virtualize the network management algorithms as applications running on commodity hardware which usually consists of various computing resources. For instance, it can utilize Centralized Processing Units (CPUs), Graphics Processing Units (GPUs), Baseband Unit (BBUs) pool, and other computing resources to carry out the computation of routing, link scheduling, and other network management work for both terrestrial and aerial hgNBs. Thus,

2168-6750 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Univ of Science and Tech Beijing. Downloaded on December 15,2021 at 01:02:31 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2020.2986238, IEEE Transactions on Emerging Topics in Computing



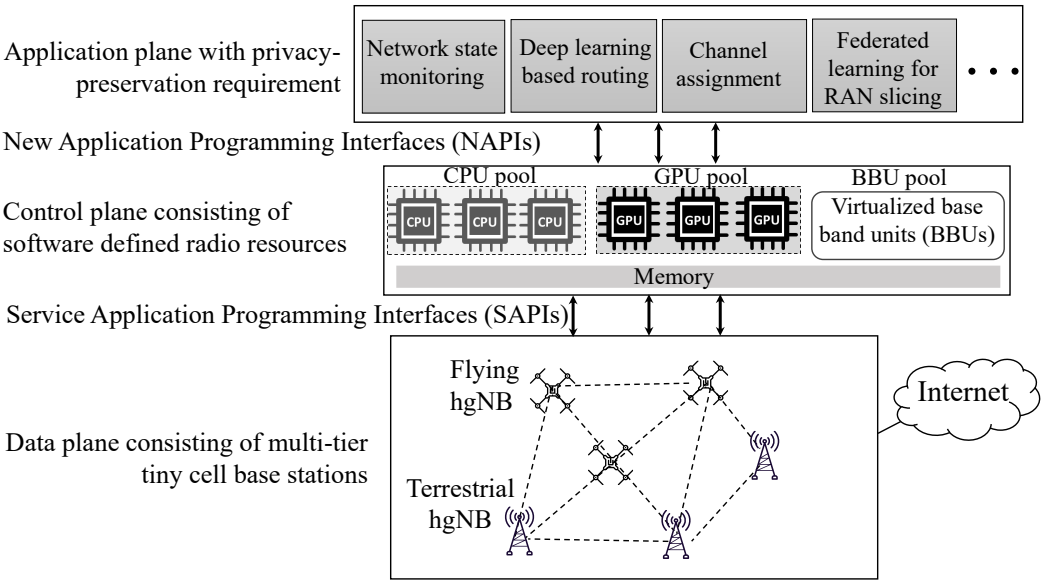



Fig. 2: Proposed heterogeneous computing platform (HCP).

the flexibility of network management can get significantly improved since the upgrade of some management methods can be fulfilled by updating the corresponding applications. Moreover, to further enhance the usage of the computing resources, caching services can be also facilitated by our proposed HCP, such as optimal caching placement, content distribution, storage services, as well as recommendation systems. The computation and networking models, in the context of performing federated learning, is delineated in the remainder of this section.

### 3.1 Proposed Computation Model

We consider a federated learning task where our proposed HCP acts as a task-coordinator while a set of hgNBs act as data collectors from a set of UEs $i = 1, ...N$. Each $UE_i \in N$ comprises a local training dataset. $UE_i$ employs a sample $s_i$ of its local data for the federated learning task. We assume labeled examples of input and output. Different features within the content are characterized by the input sample vector.

The caching entities of UEs and hgNBs are assumed to have finite storage. The UEs and hgNBs use their local learning models to make intelligent decisions on which contents should be cached. The cache is assumed to be able to store $m$ files of a constant size. UEs request their serving hgNB for the contents. If the requested content is cached in a hgNB, a cache hit occurs. In this case, the content file is directly delivered from the flying or terrestrial hgNB. In the absence of a cache hit, however, the content needs to be fetched from the original location (typically a content server on the Internet) through the serving hgNB. Hence, the aim of this work is to exploit the cache-enabled hgNBs in order to enhance the cache efficiency while reducing the service response time of UEs. In this vein, based on its local data, each UE is assumed to be able to independently calculate an update to the current global learning model and exchange the update to the HCP's parameter controller, via the serving hgNB, to aggregate a new global model. Each UE also trains a local dataset, which is generated from the usage patterns such as daily, weekly and monthly video demands based on different times of a day/week/month, different current activities, current location, contextual information derived from personal settings of mobile device sensors, and so forth. It is worth noting that each UE may have an individual context space. The local learning model in each UE and hgNB is based on Convolutional Neural Network (CNN) which will be described in Sec. 5. The local learning model update of $UE_i$ is impacted by the quality of its locally available data that can be expressed as the local accuracy, $\epsilon_i$. If the value of $\epsilon_i$ is high, we need to perform fewer local (as well as global) iterations bounded by $log(\frac{1}{\epsilon_i})$. This can be used to predict the content popularity more precisely and quickly. The computation time of a local iteration, denoted by $T_{i_L}$ in the local model training for $UE_i$ is expressed as:

$$T_{i_L} = \frac{s_i c_i}{f_i}, \tag{1}$$

where $s_i$ denotes the size of the local data of $UE_i$. The CPU cycle and number of CPU cycles needed to conduct the local model training with the sample-size $s_i$ are denoted by $f_i$ and $c_i$, respectively.

### 3.2 Considered Transmission Model

Similar to the communication model in [29], in each transmission time-round, UE-side local training data are sent to the HCP's parameter server, which are aggregated to generate an improved model. The improved model is then shared with the UEs. Therefore, each time-round involves uplink and downlink transmission overhead considering channels with Additive White Gaussian Noise (AWGN) as assumed in recent research work [30], [31]. Until the global model reaches a stable state, the communication rounds are continued. According to the recommendation list generated at the stable state, the $m$ most popular files are selected by

2168-6750 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Univ of Science and Tech Beijing. Downloaded on December 15,2021 at 01:02:31 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2020.2986238, IEEE Transactions on Emerging Topics in Computing



HCP for caching at the serving hgNBs. Let $d_i$ denote the transmission rate of $UE_i$, which can be estimated as:

$$d_i = Bln(1 + \frac{Pt_i g_i}{\zeta}), \tag{2}$$

where $B$, $Pt_i$, $h_i$, and $\zeta$ denote the transmission bandwidth of the serving hgNB, transmission power of $UE_i$, channel gain of $UE_i$, and background noise power, respectively. We assume the same data-size, $\theta$ for all UEs. Then, the transmission time, $T_{i_{Tx}}$, of a local model update with this data-size is expressed as:

$$T_{i_{Tx}} = \frac{\theta}{d_i}. \tag{3}$$

Then, the time taken for one global iteration by $UE_i$, denoted by $T_{i_G}$ is given by:

$$T_{i_G} = log(\frac{1}{\epsilon_i})T_{i_L} + T_{i_{Tx}}. \tag{4}$$

The energy consumption by $UE_i$ to transmit local model updates in a global iteration is expressed as $E_{i_{Tx}} = (T_{i_{Tx}} \cdot \rho_i = \frac{\Omega \rho_i}{B \cdot ln(1+\frac{\rho_i h_i}{N_0})})$, where $\rho_i$ and $h_i$ denote the transmit power of $UE_i$ and the channel gain, respectively. $\Omega$ refers to the local model update-size. Let $E_{i_L}$ denote the energy required for a local iteration of $UE_i$. Therefore, for a global iteration during time $t$, the total energy consumption of $UE_i$ is expressed as:

$$E_i^t = log(\frac{1}{\epsilon_i})E_{i_L} + E_{i_{Tx}}. \tag{5}$$

## 4 Problem Formulation

In the considered 6G network model, the UEs of a tiny cell, served by a hgNB (a UAV or terrestrial BS), consume high quality data content, regarding which they do not explicitly report to the hgNBs to preserve their privacy. Furthermore, the aerial/flying hgNBs are considered in the Beyond 5G network to extend network coverage in remote, rural locations or disaster-affected sites. They are assumed to be energy-limited also and can be replaced by alternative UAVs. In order to avoid the additional burden on the energy consumption of the aerial hgNBs, the participation is considered only for those UAV nodes, which have adequate energy budget during its flight-time. With the aforementioned assumptions, the terrestrial and aerial hgNBs need to find a technique to cache the most appropriate contents for the UEs to minimize the cache-miss ratio. Therefore, there is a need to rank the contents based on their popularity. Since the hgNBs do not have the global picture of all the UEs in the numerous tiny cells, they leverage the proposed HCP to jointly predict the highly popular content and user mobility so that the appropriate content can be effectively placed and cached on the most appropriate set of hgNBs. In this vein, the hgNBs collect the local training model from the UEs along with their network traffic and mobility patterns, and share these information with HCP which can train a global model to reach a reasonable decision. In addition, the hgNBs need to collaborate with each other in another level of federated learning to predict traffic rate, UE density and UE mobility to decide which contents should be relevant enough to be cached. In other words, this is a two-step federated problem addressed by the HCP to minimize a loss function while considering the 6G network parameters as follows.

$$min \sum_{i=1}^{N} \frac{\mathcal{S}_i}{\mathcal{S}} f_i(w, x_k, y_k), \tag{6}$$

where $f_i(w, x_k, y_k)$, $\mathcal{S}_i$, and $\mathcal{S}$ denote the loss function of $UE_i$, the sample size (local data) of $UE_i$ and the total number of training samples at HCP. The loss function can be further elaborated as follows:

$$f_i(w, x_k, y_k) = \frac{1}{2}(x_k^\intercal w - y_k)^2, \tag{7}$$

$$s.t.\ w_1 = w_2 = w_3 = ... = w_N = \mathcal{G},\ \forall i \in N, \tag{8}$$

where $x_k$ and $y_k$ denote the input and output of sample $k$ at $UE_i$, respectively. For each UE, $w$ refers to the weight vector to capture the parameters of its local training model using $x_k$ and $y_k$. With the increase of the prediction error $(x_k^\intercal w - y_k)$, the loss function $f_i(w, x_k, y_k)$ increases. Hence, constraint 8 ensures that upon convergence of the global learning at HCP, all UEs and HCP will share the same model for their learning tasks without the explicit data transfer from the UEs.

Thus, the problem is to design a technique so that (1) every UE may be able to train a shared global model with its local model update based on its own data, and (2) all UEs share their updated local models with HCP for revising the global model. Similarly, the hgNBs confront their own local model training problem to predict the traffic and mobility patterns of UEs and need to participate in a second-level federated learning with the global model at the HCP. The training process is repeated until the loss function (7) is minimized and the accuracy of the global model becomes acceptable.

## 5 Proposal

The parameter controller of our proposed HCP orchestrates a proactive content caching algorithm through hgNBs for UEs of the tiny cells using federated learning. Since it is vital to leverage the UEs' requests along with their contextual information to learn the future caching decisions, HCP must learn the context-specific content popularity to cache the most popular files in the hgNBs for these UEs. In order to find the hidden features from such a complex dataset, we adopt a Convolutional Neural Network (CNN) structure [32], [33]. An overview of our adopted CNN model is provided in this section, followed by our proposed learning algorithms at UEs, hgNB, and HCP.

### 5.1 Convolutional Neural Network (CNN) layer

On the UE-side, the input to a CNN structure is the time series of content access and contextual information of a UE. The CNN extracts these information using several filters of variable lengths. Initially, a 1D-convolution over a number of input matrices is applied by using a sliding window that transforms the input signal into representative values. The convolution operation preserves the spatial relationship of the input by learning feature maps. Each UE data is treated as a separate sample and input to the neural network. Given

2168-6750 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Univ of Science and Tech Beijing. Downloaded on December 15,2021 at 01:02:31 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2020.2986238, IEEE Transactions on Emerging Topics in Computing



the input size ($N$, $C_{in}$, L), the output of the CNN structure ($N$, $C_{out}$, $L_{out}$) is given by:

$$\text{output}_{\text{CNN}}(N_i, C_{out_j}) = \text{bias}(C_{out_j}) + \sum_{k=0}^{C_{in}-1} w(C_{out_j}, k) * \text{input}(N_i, k), \quad (9)$$

where $*$, $N$, $C$, $L$, and $w$ refer to the valid cross-correlation operator, batch size, number of channels, the signal sequence length, and the weights of the connections, respectively. Then, the learnable weights and bias variables are set, and the pooling layer is inserted between the convolutional layers to perform down-sampling by reducing the matrix-operations for the subsequent colvolutional layer. The pooling layer employs a sliding window, which is moved over the input matrix. At each sliding step $i$, the input with length $l$ is fed to the model. Rectified Linear Unit (RELU) is employed as the activation function. The output of the convolutional layer, i.e., the feature set or feature matrix, is obtained as follows:

$$\text{feature}_{\text{set}}(N_i, C_j, l) = \frac{1}{k} \sum_{m=0}^{k} \text{input}(N_i, C_j, \text{stride} * l + m). \quad (10)$$

In the fully connected layer, backpropagation [28] is used to fine-tune the weight vector ($w_i$) of $UE_i$.

During each sliding step $t$, an input $x_k$ is fed to the CNN model for $UE_i$. The output is set to the corresponding point $y_k$.

### 5.2 Federated Learning Algorithm Exploiting Asynchronous Model Update

The DNN structures in the local model at UEs are supposed to consist of shallow and dense layers. While in conventional federated averaging [19], the parameters of the entire DNN structure are updated at the same time, this contributes to a huge communication overhead. The parameters of the shallow layers help the system to learn general features of the content access. On the other hand, a large number of parameters are generated at the deep layers to learn specific features related to specific content features and context information of the UEs. As a consequence, the parameters of the shallow layers could be updated more frequently in contrast with those of the deep layers in an asynchronous fashion as shown in Algorithms 1, 2, and 3 on the UE, hgNB and HCP, respectively.

Algorithm 1 shows the local model update at $UE_i$. There are three inputs to this algorithm: the UE identifier $i$, the weight vector of the local model $w$, and a control parameter $checkDepth$. The content data and context information are split into mini-batches and $\mathcal{B}$ represents the mini-batch size. Additionally, $\epsilon$ denotes the epoch of the local model. The control parameter $checkDepth$ is exploited to select whether all the layers or the shallow layers will be updated. In line 8 of the algorithm, Stochastic Gradient Descent (SGD) is conducted which is given by:

$$w = w - \eta * \nabla l(w; b), \quad (11)$$

where $\eta$ is the learning rate.

Next, in Algorithm 2, the scheduling of $hgNB_x$ for polling the UEs, coordinated by the HCP, is implemented. As described in the steps of this algorithm, $hgNB_x$ first receives the timestamp list for each of its UEs from the HCP and initializes a temporary buffer. The timestamp list is used for deciding whether to perform a shallow layer parameter fetch or a complete parameter fetch for all the layers. In addition, $hgNB_x$ starts a clock and synchronizes it with that of HCP. Then, for each round, it polls its served UEs and invokes the LocalUpdate($i, w, checkDepth$) function (Algorithm 1) to receive the relevant parameters from $UE_i$ given the current timestamp and the $checkDepth$ control variable. Then, it concatenates the fetched parameters of $UE_i$ into the buffer. After all the UEs are polled, the buffer holds the information of shallow-only or complete weight vector parameters for the respective timestamps of each served UE of $hgNB_x$. The clock, then, points to the timestamp of the next UE to be polled. The buffer is transmitted to HCP. Note that $hgNB_x$ also locally runs its deep learning structure for predicting traffic and UEs mobility and shares the shallow/deep parameters of its own learning method in Line 1 of Algorithm 2.

On the other hand, the temporally weighted aggregation asynchronous federated learning is proposed in Algorithm 3 at HCP. First, HCP carries out an initialization step followed by several communication rounds [34]. Timestamps for general parameters ($timestamp_g$) and specific parameters ($timestamp_s$) corresponding to shallow-only and deep-structure weights are initialized. The federated learning takes place in each round $T$, which is divided into $\Delta$ time-slots. Prior to each round, HCP sends the timestamp list it initialized for each UE to the serving hgNB. Then, for each round $T$, the time-slots which are in $\delta$ (the set of time-slots during which specific parameters are fetched from UEs), the control variable $checkDepth$ is set to true. A participating subset of the clients is randomly selected per round. The function $Scheduling_{hgNB_x}$ is called in parallel to get the specific and general parameters while updating the respective timestamps. Thus, the aggregation is performed to update $w_g$ and $w_s$ in lines 19 to 26. Furthermore, line 27 shows the parallel execution of asynchronous update steps for general and specific parameters of $hgNB_x$ similar to those for $UE_i$.

### 5.3 Algorithmic Analysis

To solve the optimization problem 7, HCP transmits the parameters $g = w$ of the global federated learning model to UEs through the serving hgNB so that they are able to train their local models. Then, UEs transmit their local models to the serving hgNB, which polls UEs during specific timestamps (informed by HCP) and exploits the control parameter to selectively fetch shallow-only and deep parameters. In the proposed federated learning using asynchronous model update, the update of each $UE_i$'s local model parameter $w_i$ depends on the global model $g$ while the update of the global model $g$ relies on all UEs' local federated learning models given in Algorithm 1. The update of the local model $w_i$ depends on the learning algorithm such as Gradient Descent, Stochastic Gradient Descent (SGD) or Randomized Coordinate Descent (RCD). On the other hand, the update of the global model $g$ is obtained as $\frac{\sum_{i=1}^{N} K_i w_i}{K}$.

2168-6750 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Univ of Science and Tech Beijing. Downloaded on December 15,2021 at 01:02:31 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2020.2986238, IEEE Transactions on Emerging Topics in Computing



**Algorithm 1** 1st stage Federated Learning at $UE_i$

**Input:** $i, w, checkDepth$
**Output:** $w_{general}, w_{specific}$
**Function:** $LocalUpdate(i, w, checkDepth)$
*Initialization* : $\mathcal{B}$ = Divide $s_k$ into batches of size $B$
**if** (checkDepth = "Deep") **then**
  $w_s$=$w$
**else**
  $w_g$=$w$
**end if**
*Loop Process*
**for** each local epoch $t$ from 1 to $\mathcal{E}$ **do**
  **for** each batch $b$ from 1 to $\mathcal{B}$ **do**
    $w = w - \eta * \nabla l(w; b)$
  **end for**
**end for**
**if** (checkDepth= "Deep") **then**
  Transmit $w_g$ to $hgNB_x$
**else**
  Transmit $w_s$ to $hgNB_x$
**end if**

**Algorithm 2** $hgNB_x$ Scheduling (2nd stage federated learning).

**Input:** Clock, timeround $T$
**Function:** $Scheduling_h gNBx(T)$
LocalUpdate($x$,$w_x$,$checkDepth_x$)
Retrieve timestamp list from $HCP$ for each $UE_i$ for shallow or deep layer parameter fetch and initialize a temporary buffer.
**while** clock **do**
  Reset buffer
  **for** each $UE_i$ from 1 to $N$ **do**
    **if** (timestamp of $UE_i$ is in retrieved list) **then**
      concatentate (buffer, "$UE_i$, timestamp: ", $LocalUpdate(i, w, checkDepth)$)
    **end if**
    Transmit buffer to $HCP$
    Increment timestamp of $UE_i$ to next instant
  **end for**
**end while**

For the local model update, during the training process, the update of $UE_i$'s local model $w_i$ at time $t$ is given by:

$$w_{i,t+1} = g_t - \frac{\eta}{S_i} \sum_{k}^{S_i} = 1 \nabla f(g_t, x_{ik}, y_{ik}), \tag{12}$$

where $\eta$ is the learning rate and $\nabla f(g_t, x_{ik}, y_{ik})$ denotes the gradient of $f(g_t, x_{ik}, y_{ik})$ with respect to $g_t$.

Assuming that $F(g) = \frac{1}{S} \sum_{i=1}^{N} \sum_{k=1}^{S_i} f(g, x_{ik}, y_{ik})$ and $F_i(g) = \sum_{k=1}^{S_i} f(g, x_{ik}, y_{ik})$, the update of global model $g$ at time $t$ can be given by:

$$g_{t+1} = g_t - \eta(\nabla F(g_t) - \lambda). \tag{13}$$

Here, $\lambda = \frac{\sum_{i=1}^{N} K_i a_i w_i C(w_i)}{\sum_{i=1}^{N} K_i a_i C(w_i)}$, where $C(w_i)$ is 1 if a resource block is assigned by hgNB to $UE_i$ for transmitting its local model parameters with sufficient transmit-energy.

**Algorithm 3** Federated Learning of $HCP$

**Input:** $\Delta$ (number of time-slots per round), $\kappa$ (fraction of participating UEs per round), $\psi$ (set of UEs who are participating in current round for federated learning), $\delta$ (set of time-slots during which specific parameters are fetched from UEs).
1: **for** each UE $i$ from 1 to $N$ **do**
2:  $timestamp_g^i = 0, timestamp_s^k = 0$
3: **end for**
4: Send timestamp list to serving hgNBs
5: **for** each round $T$ **do**
6:  **if** ($T \mod \Delta \in \delta$) **then**
7:  checkDepth = True
8:  **else**
9:  checkDepth = False
10:  **end if**
11:  $n = max(\kappa \cdot N, 1)$
12:  $\psi = (random\ set\ of\ n\ UEs)$
13:  **for** each $UE_i \in \psi$ in parallel **do**
14:  **if** (checkDepth=True) **then**
15:  $w_s^i = Scheduling_h gNBx(T)$
16:  $timestamp_g^i = t$
17:  $timestamp_s^i = t$
18:  **else**
19:  $w_g^i = Scheduling_h gNBx(T)$
20:  $timestamp_g^i = t$
21:  **end if**
22:  **end for**
23:  $w_{g,(T+1)} = \sum_{i=1}^{S} \frac{n_S}{n} * f_g(T, i) * w_g^i$
24:  **if** (checkDepth=True) **then**
25:  $w_{s,(T+1)} = \sum_{i=1}^{S} \frac{n_S}{n} * f_g(T, i) * w_s^i$
26:  **end if**
27: **end for**
28: **for** each hgNB$x$ from 1 to $X$ **do**
29:  Execute asynchronous update steps for general and specific parameters of hgNB$x$ similar to lines 1-27
30:  Use global model to instruct hgNB$x$ to update its by content
31: **end for**

Suppose that the federated learning algorithm converges to an optimal global model $g*$ after the learning steps in Algorithm 3. The expected convergence rate of the federated learning can be referred to in the work in [35].

## 6 Performance Evaluation

In this section, the proposed HCP based federated learning for content caching algorithm is evaluated using two real-world datasets containing 100,000 ratings on 1682 movies by 943 users and over a million ratings on 3883 movies by 6040 users, respectively. Other types of contents such as health data, social posts, user location data, music/photos, can also be supported by extending our proposed framework. However, please note that these other use-cases or applications are beyond the scope of this work. Therefore, in this paper, we provided a proof-of-concept of the proposed two-step federated learning process for content recommendation for UEs with privacy-preservation.

2168-6750 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Univ of Science and Tech Beijing. Downloaded on December 15,2021 at 01:02:31 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2020.2986238, IEEE Transactions on Emerging Topics in Computing



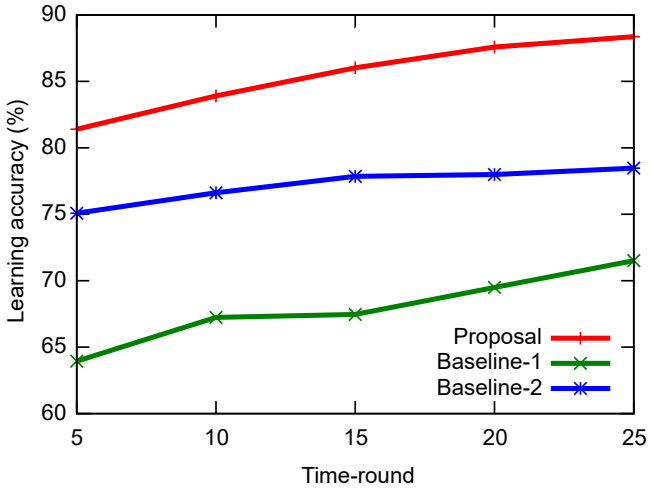



Fig. 3: Learning accuracy comparison of existing federated learning method and proposal over time-rounds.

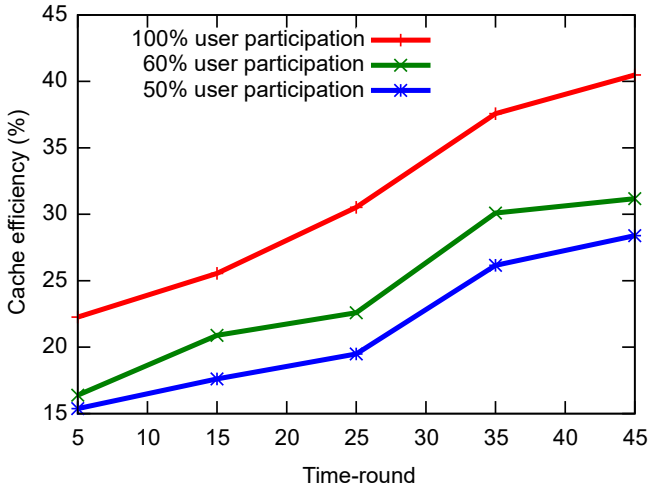



Fig. 5: Cache efficiency comparison over time-rounds for 60% and 100% UE-participation ratios.

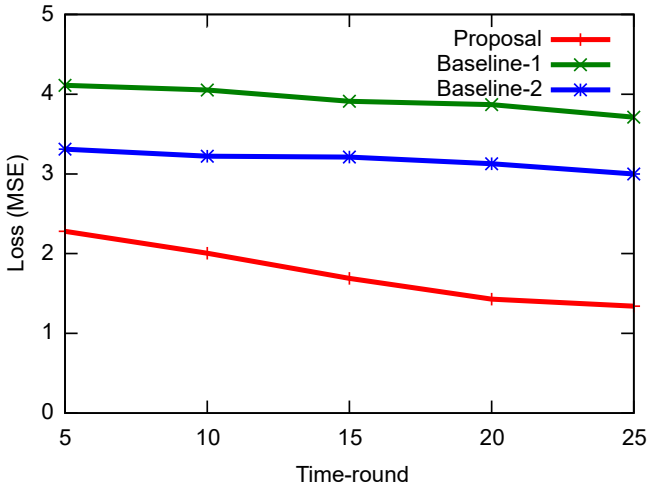



Fig. 4: Loss comparison of existing federated learning method and proposal over time-rounds.


Fig. 6: The value of the loss function over growing number of iterations.

For both datasets, a rating scale of zero to five is used. Each user is considered to be a UE that performs 20 movie ratings or more. The contextual information of the UEs are also obtained such as gender, age, occupation and location. Assuming that the rated movies are the files requested by users, the streaming content requests are simulated as [36], [37]. The considered simulation parameters are listed in Table 1. The simulation scenario is considered for 50 tiny cells, each with radius of 2 to 3m that serve a population of UEs varying from 943 to 6040. Aerial-terrestrial hgNB link capacity is assumed to be 3-Dimensional (in terms of bps/Hz/$m^3$) allowing an operating frequency of up to 1 THz. The peak data rate each UE experiences is 10 Gbps. Therefore, 21.2 GHz of unlicensed spectrum and 95 GHz to 3 THz for experimental licenses are considered that correspond to encouraging performance for forthcoming 6G networks [38]. The wireless channels of UEs and hgNBs using mmWave and THz frequencies are expected to experience approximately 30 dB/km of rain attenuation above 100 GHz in Non-Line of Sight (NLOS) environments. [39]. Generalized Index Modulation is considered to guarantee best utilization of the available spectrum at the THz-band of the considered 6G tiny cell using ultra-massive Multiple Input Multiple Output (um-MIMO) [40], [41] supporting cross-polarization array with 512 physical antenna elements having 5dBi max gain per physical element and up to 1024-Quadrature Amplitude Modulation (QAM). Two UE types are considered, pedestrian mobile users with average speed of 1 m/s and autonomous vehicles with average speed up to 25 m/s. Each UE is assumed to be edge-AI enabled node consisting of 64 physical antenna elements with 23 dBm transmit power feed. On the other hand, HCP is considered to be based on a high performance node as shown in Fig. 2 which has pools of CPU, GPU and BBU resources for serving the aerial and terrestrial hgNBs over 5 $km^2$ area. For simplicity, equal number of UAV and terrestrial hgNB is considered that are arbitrarily distributed over the 50 tiny cells considered for the evaluation. The UEs and

2168-6750 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Univ of Science and Tech Beijing. Downloaded on December 15,2021 at 01:02:31 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2020.2986238, IEEE Transactions on Emerging Topics in Computing



TABLE 1: Considered simulation parameters.

| Parameter | Value |
|---|---|
| Per UE peak data rate | 10 Gbps |
| Per aerial-terrestrial hgNB link capacity | 1 Tbps, 3-D (bps/Hz/$m^3$) |
| Operating frequency (backhaul) | up to 1 THz |
| Tiny cell radius | 2-3 m |
| Downlink spectral efficiency | 100 bps/Hz |
| UE mobility speed | 1 to 25 m/s |
| UE types | $UE_1$ (pedestrian UE), $UE_2$ (autonomous vehicle) |
| Number of UEs per tiny cell | 2 to 20 |
| Interference suppression | $UE_1$ (Delta Orthogonal), $UE_2$ (Multiple Access) |
| Content type | Movie |
| Learning model type (UE, ghNB, HCP) | 1-D convolution CNN |
| Optimizer type | Combined adaptive learning rates and momentums |

aerial hgNBs are assumed to be edge-AI enabled nodes to have sufficient computational capability to train local models. The learning model type for UE, ghNB and HCP is 1-Dimensional CNN whereby the local and global models share combined adaptive learning rates and momentums. For the proposed federated learning, two one-dimensional (1D) convolution layers were constructed using Python 3 and TensorFlow 2.0. Each 1-D convolution layer consisted of one hundred filters. The dropout was set to 0.2. Maxpooling was employed to obtain down-sampled feature set followed by batch normalization. Rectified Linear Unit (ReLU) was used as the activation function in the convolution layers. Softmax was used as the activation function in the output layer. Next, for the baseline-1 method, a shallow ANN was constructed with a single hidden layer of one hundred units. On the other hand, for the baseline-2 method, a DNN was constructed with an input layer size of 200 units. The DNN model comprised two hidden layers with 100 and 50 units, respectively, followed by an output layer having 25 units. In both the baseline approaches, ReLU and Softmax were adopted as activation functions in the hidden and output layers, respectively. In the proposed and the two baseline approaches, Adaptive Moment Estimation (ADAM), categorical cross-entropy, and mean squared error were used for optimizing the loss function, estimating loss, and measuring training and validation accuracies, respectively. The aforementioned datasets were split for training and testing. The proposed and baseline models were constructed on a workstation with Intel Core i7-9700 CPU (operating at 3.00 GHz), 16 GB Random Access Memory (RAM), and a single NVIDIA RTX 2060 GPU.

First, simulations are conducted to evaluate the learning accuracy using our federated learning approach with CNN and two baseline federated learning methods using ANN and DNN [19], referred to as baseline-1 and baseline-2 methods, respectively. The results are plotted in Fig. 3. The time-rounds were varied from 5 to 25. The learning accuracy for baseline-1 method using a shallow ANN was the worst (approximately 64%) and it was found to saturate toward 70% after 30 time-rounds. The baseline-2 method using a deep neural network model improved the learning accuracy (approximately 75%) during the early time-rounds and maintained this with slight accuracy improvement until time-round 45. In other words, the accuracy gradually increased in case of the baseline methods and became stable during the 25th time-round. On the other hand, our proposed federated learning, using CNN with asynchronous update, achieved high learning accuracy (over 80%) during the early time-rounds and continued this high performance throughout the considered time-rounds. In addition, it significantly outperformed the two baseline approaches throughout all the time-rounds in terms of learning accuracy. Furthermore, Fig. 4 demonstrates the improvement of the proposed federated learning method in terms of Mean Squared Error (MSE) loss in contrast with the baseline methods.

Next, in Fig. 5, we compare the cache efficiency over 45 time-rounds for three scenarios. In the first scenario, HCP took advantage of federated learning from all, 60% and 50% participating UEs in each time-round, respectively. The result indicates that the cache efficiencies when 60% and 50% UEs are considered remain reasonable compared to that during the complete UE-participation. This good performance can be attributed to the asynchronous weight update for the global model where not all UEs need to transmit their shallow and deep parameters ($w_g$ and $w_s$) to hgNBs and HCP.

Next, Fig. 6 demonstrates the value of the loss function over a growing number of iterations. As the number of iterations approaches 12, the loss function is minimized and continues to be below one throughout the remaining number of iterations up to 20.

Additionally, the loss values for increasing number of UEs for the proposed method are compared with two baseline methods in Fig. 7. The proposal achieves superior loss function improvement compared to the baseline methods for two UEs. When the number of UEs exceeds 12, the distributed nature of the proposal along with the asynchronous weight update model allows the loss function to improve more aggressively in case of the proposed method, and it achieves the loss function value close to one when 16 UEs are considered. Furthermore, the loss function values for various values of average sample size per UE are plotted in Fig. 8. The result indicates that the proposed approach outperforms the baseline methods even when the UE has higher sample size, because the specific local parameters are used to update the global model at HCP only at specific time-slots during each time-round. On the other hand, the number of iterations required for convergence for a growing number of UEs is demonstrated in Fig. 9. The results elucidate that the proposal requires much lower number of iterations compared to the baseline methods for just two UEs. When the number of UEs exceeds 8, the distributed nature of the proposal along with the asynchronous weight update model allows the loss function to improve with increasing number of iterations in case of the proposed method. However, the number of needed iterations by the proposal still remained significantly lower compared to those required by the baseline methods.

Cache efficiency is used as the evaluation metric for our proposal that measures the ratio of cache hits to the number of UE requests on the cache. The CNN-based proposed model is employed to find latent features between UEs and

2168-6750 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Univ of Science and Tech Beijing. Downloaded on December 15,2021 at 01:02:31 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2020.2986238, IEEE Transactions on Emerging Topics in Computing



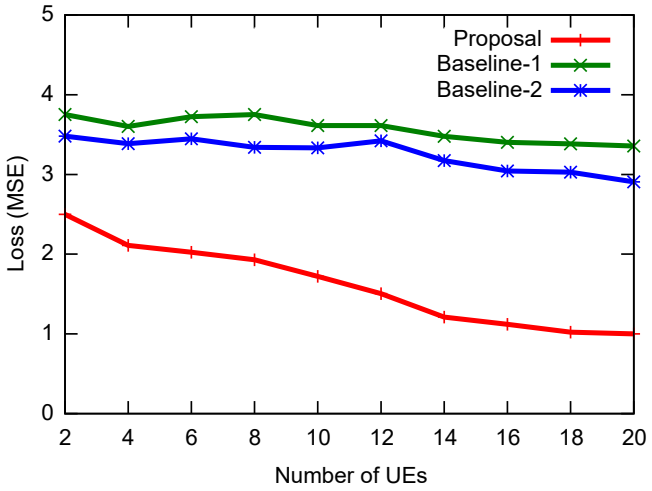



Fig. 7: The value of the loss function for growing number of UEs for two baseline methods and the proposal.

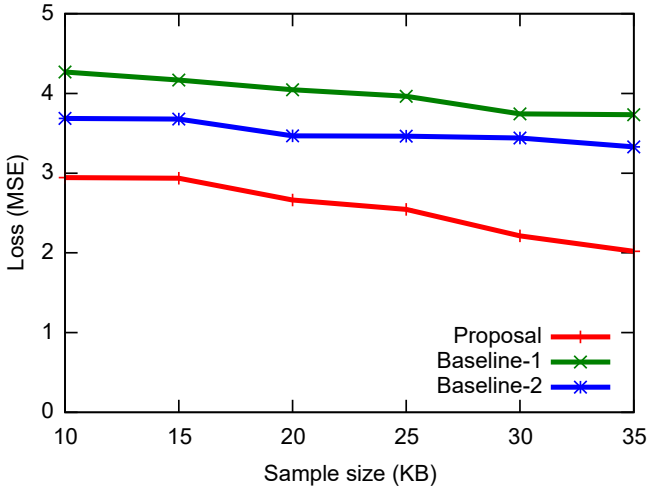



Fig. 8: The value of the loss function for average sample size per UE for the baseline methods and the proposal.

between files to estimate the popularity of the content. The cache efficiency for varying cache sizes between 50 to 4300 files is evaluated. The baseline performances are obtained from a fully knowledgeable algorithm and baseline-2 for the 100k and 1M datasets in Fig. 10 where our proposed federated learning method achieves close performance to the theoretic maximum of the fully knowledgeable algorithm and stays well above the baseline-2 algorithm.

Finally, Fig. 11 demonstrates the overhead reduction in terms of $S_s$, i.e., the specific weight parameter contributed by the deep structures using the asynchronous update model in the proposed federated learning method. A total of 20 rounds are considered in this scenario. As shown in the result, the higher the value of $S_s$, the more overhead reduction is possible by performing more generic parameter exchange from the local model through the hgNB to HCP and only transferring the specific parameters after a relatively high number of time-rounds.

## 7 Conclusion

As most countries are poised to embrace full-scale deployment of 5G networks, academia-industry initiatives

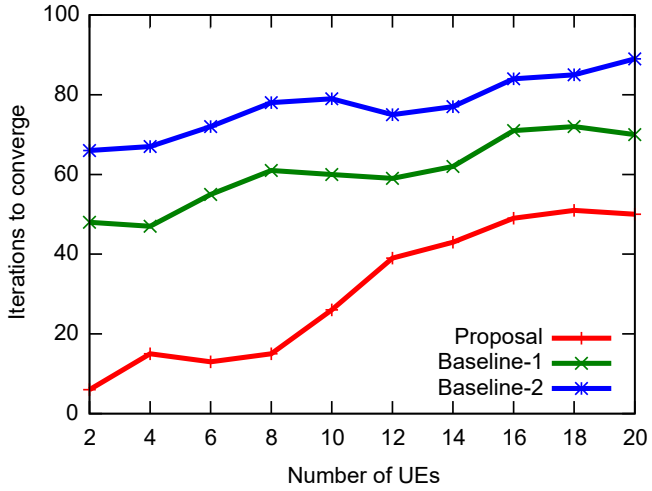



Fig. 9: The number of iterations required for convergence for growing number of UEs.

are already identifying their shortcomings in supporting multi-tier radio access networks with heterogeneous radio access technologies. Thus, there is already a push for 6G communication networks. Based on the ongoing conceptualization of 6G networks, researchers have pointed out that edge-embedded AI will be a key feature to assure the performance guarantee of futuristic services such as massive IoT communications, tactile Internet, robotic surgery, augmented and virtual reality, and so forth. In this paper, we capitalized the edge intelligence of the mobile users (UEs) to build local training models regarding their content experience and preference. We also used the aerial as well as terrestrial hgNBs to schedule and build their own local model to predict traffic flow and UE-distribution. The local training models from the UEs and hgNBs are transferred to our proposed HCP controller, which trains its global model based on our proposed two-stage federated learning using asynchronous parameter update algorithm. Thus, the proposed HCP utilized the cooperative communication capabilities of the UEs and hgNBs and leveraged the two-stage federated learning to collaboratively predict the content caching placement by jointly considering traffic distribution, UE-mobility and localized content popularity. The learned model is transferred to the hgNBs as well as UEs to further refine the local learning process. Numerical results demonstrated the effectiveness of our proposed method.

## References

[1] "Cisco visual networking index: Global mobile data traffic forecast update, 2017–2022 white paper," (accessed October 28, 2019). [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html

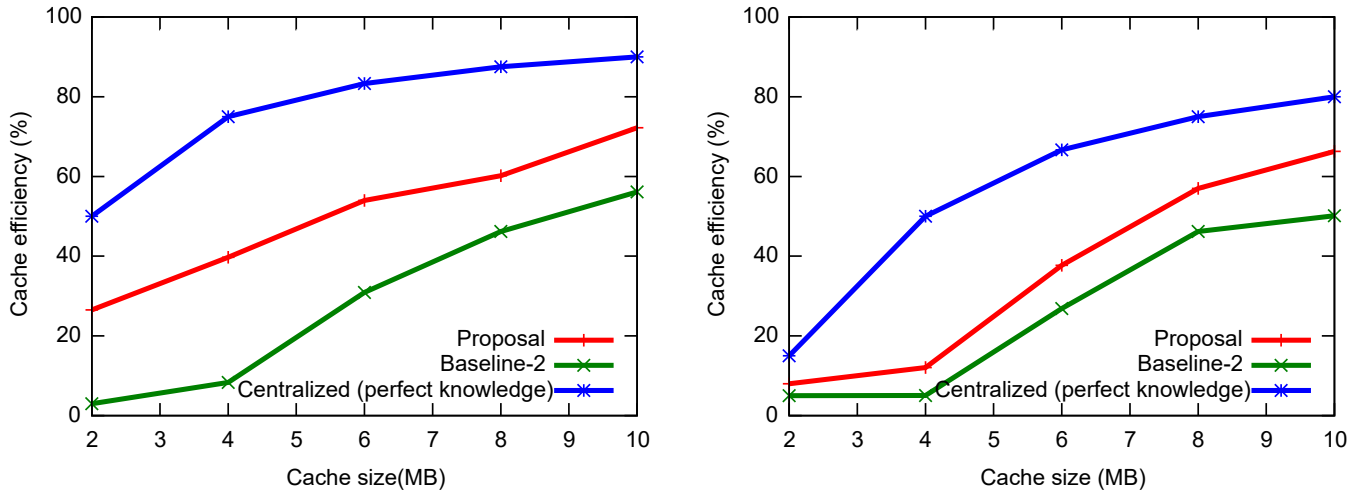
[2] S. M. A. Oteafy and H. S. Hassanein, "Leveraging tactile internet cognizance and operation via iot and edge technologies," *Proceedings of the IEEE*, vol. 107, no. 2, pp. 364–375, Feb 2019.

[3] G. Piumatti, F. Lamberti, A. Sanna, and P. A. Montuschi, "Robust robot tracking for next-generation collaborative robotics-based gaming environments," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2017.

2168-6750 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Univ of Science and Tech Beijing. Downloaded on December 15,2021 at 01:02:31 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2020.2986238, IEEE Transactions on Emerging Topics in Computing






(a) Cache efficiency for different cache sizes for the 100k dataset. (b) Cache efficiency for different cache sizes for the 1M dataset.

Fig. 10: Cache efficiency for different cache sizes for a random algorithm (the worst case), fully knowledgeable algorithm (theoretical maximum) and proposed federated learning algorithm.

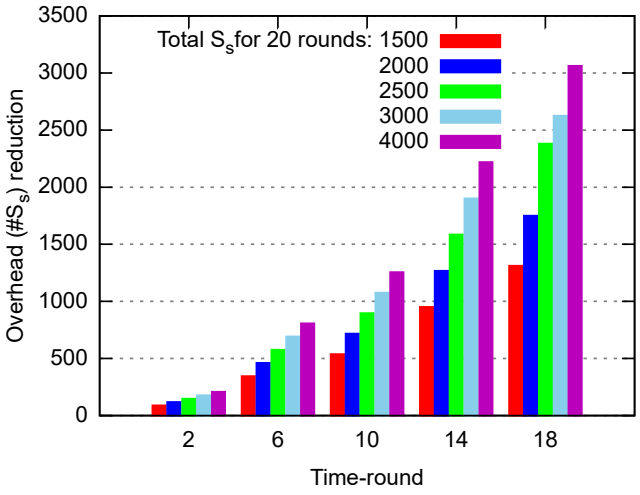



Fig. 11: The overhead reduction (special weight parameter contributed by the deep structures) using the asynchronous update model in the proposed federated learning method.


[4] F. Tariq, M. R. A. Khandaker, K. Wong, M. A. Imran, M. Bennis, and M. Debbah, "A Speculative Study on 6G," *CoRR*, vol. abs/1902.06700, 2019. [Online]. Available: http://arxiv.org/abs/1902.06700

[5] F. Lamberti, A. Cannavò, and P. Montuschi, "Is immersive virtual reality the ultimate interface for 3D animators?" *Computer*, to appear, 2019.

[6] S. Dang, O. Amin, B. Shihada, and M. Alouini, "From a human-centric perspective: What might 6g be?" *CoRR*, vol. abs/1906.00741, 2019. [Online]. Available: http://arxiv.org/abs/1906.00741

[7] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y. A. Zhang, "The Roadmap to 6G: AI Empowered Wireless Networks," *IEEE Communications Magazine*, vol. 57, no. 8, pp. 84–90, August 2019.

[8] Y. Al-Eryani and E. Hossain, "The D-OMA Method for Massive Multiple Access in 6G: Performance, Security, and Challenges," *IEEE Vehicular Technology Magazine*, vol. 14, no. 3, pp. 92–99, Sep. 2019.

[9] Y. L. Lee, D. Qin, L.-C. Wang, G. Hong, and Sim, "6g massive radio access networks: Key issues, technologies, and future challenges," 2019.

[10] P. Yang, Y. Xiao, M. Xiao, and S. Li, "6G Wireless Communications: Vision and Potential Techniques," *IEEE Network*, vol. 33, no. 4, pp. 70–75, July 2019.

[11] N. Kato, B. Mao, F. Tang, Y. Kawamoto, and J. Liu, "Ten Challenges in Advancing Machine Learning Technologies towards 6G," *IEEE Wireless Communications Magazine*, 2020.

[12] F. Tang, Y. Kawwamoto, N. Kato, and J. Liu, "Edge Cloud Server Deployment with Transmission Power Control through Machine Learning for 6G Internet of Things," *Transactions on Emerging Topics in Computing (TETC)*, 2020.

[13] ——, "Future Intelligent and Secure Vehicular Network Towards 6G: Machine-Learning Approaches," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 292–307, Feb. 2020.

[14] S. Zhang, J. Liu, H. Guo, M. Qi, and N. Kato, "Envisioning Device-to-Device Communications in 6G," *IEEE Network Magazine*, 2020.

[15] G. Gui, M. Liu, F. Tang, N. Kato, and F. Adachi, "6G: Opening New Horizons for Integration of Comfort, Security and Intelligence," *IEEE Wireless Communications*, pp. 1–7, 2020.

[16] F. Tang, Z. M. Fadlullah, B. Mao, N. Kato, F. Ono, and R. Miura, "On A Novel Adaptive UAV-Mounted Cloudlet-Aided Recommendation System for LBSNs," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2018.

[17] N. Kato, Z. M. Fadlullah, F. Tang, B. Mao, S. Tani, A. Okamura, and J. Liu, "Optimizing space-air-ground integrated networks by artificial intelligence," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 140–147, August 2019.

[18] D. Baek, Y. Chen, A. Bocca, L. Bottaccioli, S. D. Cataldo, V. Gatteschi, D. J. Pagliari, E. Patti, G. Urgese, N. Chang, A. Macii, E. Macii, P. Montuschi, and M. Poncino, "Battery-Aware Operation Range Estimation for Terrestrial and Aerial Electric Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5471–5482, June 2019.

[19] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecný, S. Mazzocchi, H. B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards Federated Learning at Scale: System Design," *CoRR*, vol. abs/1902.01046, 2019. [Online]. Available: http://arxiv.org/abs/1902.01046

[20] B. McMahan and D. Ramage, "Federated Learning: Collaborative Machine Learning without Centralized Training Data," (accessed November 20, 2019). [Online]. Available: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

[21] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017. [Online]. Available: http://arxiv.org/abs/1602.05629

[22] P. Montuschi, F. Lamberti, V. Gatteschi, and C. Demartini, "A Semantic Recommender System for Adaptive Learning," *IT Professional*, vol. 17, no. 5, pp. 50–58, Sep. 2015.


2168-6750 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Univ of Science and Tech Beijing. Downloaded on December 15,2021 at 01:02:31 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2020.2986238, IEEE Transactions on Emerging Topics in Computing



[23] R. Fares, B. Romoser, Z. Zong, M. Nijim, and X. Qin, "Performance Evaluation of Traditional Caching Policies on a Large System with Petabytes of Data," in *2012 IEEE Seventh International Conference on Networking, Architecture, and Storage*, June 2012, pp. 227–234.
[24] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *CoRR*, vol. abs/1209.5807, 2012. [Online]. Available: http://arxiv.org/abs/1209.5807
[25] E. Bastug, M. Bennis, and M. Debbah, "Living on the Edge: The Role of Proactive Caching in 5G Wireless Networks," *CoRR*, vol. abs/1405.5974, 2014. [Online]. Available: http://arxiv.org/abs/1405.5974
[26] S. Müller, O. Atan, M. van der Schaar, and A. Klein, "Context-Aware Proactive Content Caching With Service Differentiation in Wireless Networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 2, pp. 1024–1036, Feb 2017.
[27] A. Sengupta, R. Tandon, and T. C. Clancy, "Fundamental Limits of Caching With Secure Delivery," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 355–370, Feb 2015.
[28] B. Mao, Z. M. Fadlullah, F. Tang, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "Routing or Computing? The Paradigm Shift Towards Intelligent Computer Network Packet Transmission Based on Deep Learning," *IEEE Transactions on Computers*, vol. 66, no. 11, pp. 1946–1960, Nov 2017.
[29] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y. Liang, and D. I. Kim, "Incentive Design for Efficient Federated Learning in Mobile Networks: A Contract Theory Approach," *CoRR*, vol. abs/1905.07479, 2019. [Online]. Available: http://arxiv.org/abs/1905.07479
[30] N. Nomikos, E. T. Michailidis, P. Trakadas, D. Vouyioukas, H. Karl, J. Martrat, T. Zahariadis, K. Papadopoulos, and S. Voliotis, "A UAV-based moving 5G RAN for massive connectivity of mobile users and IoT devices," *Vehicular Communications*, vol. 25, p. 100250, 2020.
[31] L. Xiang, L. Lei, S. Chatzinotas, B. Ottersten, and R. Schober, "Towards Power-Efficient Aerial Communications via Dynamic Multi-UAV Cooperation," 2020. [Online]. Available: https://arxiv.org/abs/2001.11255
[32] N. Q. K. Le, E. K. Y. Yapp, and H.-Y. Yeh, "ET-GRU: Using Multi-layer Gated Recurrent Units to Identify Electron Transport Proteins," *BMC Bioinformatics*, vol. 20, no. 1, p. 377, 2019. [Online]. Available: https://doi.org/10.1186/s12859-019-2972-5
[33] Y. Xu, Q. Kong, Q. Huang, W. Wang, and M. D. Plumbley, "Convolutional gated recurrent neural network incorporating spatial features for audio tagging," *CoRR*, vol. abs/1702.07787, 2017. [Online]. Available: http://arxiv.org/abs/1702.07787
[34] Y. Chen, X. Sun, and Y. Jin, "Communication-Efficient Federated Deep Learning with Asynchronous Model Update and Temporally Weighted Aggregation," *CoRR*, vol. abs/1903.07424, 2019. [Online]. Available: http://arxiv.org/abs/1903.07424
[35] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," 2019.
[36] Z. Yu, J. Hu, G. Min, H. Lu, Z. Zhao, H. Wang, and N. Georgalas, "Federated learning based proactive content caching in edge computing," in *2018 IEEE Global Communications Conference (GLOBECOM)*, Dec 2018, pp. 1–6.
[37] S. Li, J. Xu, M. van der Schaar, and W. Li, "Popularity-driven content caching," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, April 2016, pp. 1–9.
[38] Y. Xing, O. Kanhere, S. Ju, and T. S. Rappaport, "Indoor Wireless Channel Properties at Millimeter Wave and Sub-Terahertz Frequencies," 2019.
[39] T. S. Rappaport, Y. Xing, O. Kanhere, S. Ju, A. Madanayake, S. Mandal, A. Alkhateeb, and G. C. Trichopoulos, "Wireless Communications and Applications Above 100 GHz: Opportunities and Challenges for 6G and Beyond," *IEEE Access*, vol. 7, pp. 78 729–78 757, 2019.
[40] H. Sarieddeen, M. Alouini, and T. Y. Al-Naffouri, "Terahertz-Band Ultra-Massive Spatial Modulation MIMO," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 9, pp. 2040–2052, Sep. 2019.
[41] E. Basar, "Reconfigurable Intelligent Surface-Based Index Modulation: A New Beyond MIMO Paradigm for 6G," 2019.

**Zubair Md Fadlullah** is currently an Associate Professor with the Computer Science Department, Lakehead University, and a Research Chair of the Thunder Bay Regional Health Research Institute (TBRHRI), Thunder Bay, Ontario, Canada. He was an Associate Professor at the Graduate School of Information Sciences (GSIS), Tohoku University, Japan, from 2017 to 2019. He also served at GSIS as an Assistant Professor from 2011 to 2017. His main research interests are in the areas of emerging communication systems like 5G New Radio and beyond, deep learning applications on solving computer science and communication system problems, UAV based systems, smart health technology, cyber security, game theory, smart grid, and emerging communication systems. He was a recipient of the prestigious Dean's and President's Awards from Tohoku University in March 2011, and the IEEE Asia Pacific Outstanding Researcher Award in 2015 and NEC Tokin Award for research in 2016, for his outstanding contributions. He has also received several best paper awards at conferences including IWCMC, Globecom, and IC-NIDC. He is a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE), and IEEE Communications Society (ComSoc).

**Nei Kato** is a full professor (Deputy Dean) with Graduate School of Information Sciences (GSIS) and the Director of Research Organization of Electrical Communication (ROEC), Tohoku University, Japan. He has been engaged in research on computer networking, wireless mobile communications, satellite communications, ad hoc & sensor & mesh networks, UAV networks, smart grid, AI, IoT, Big Data, and pattern recognition. He has published more than 400 papers in prestigious peer-reviewed journals and conferences. He is the Vice-President(Member Global Activities) of IEEE Communications Society(2018-), the Editor-in-Chief of IEEE Transactions on Vehicular Technology(2017- ), and the Chair of IEEE Communications Society Sendai Chapter. He served as the Editor-in-Chief of IEEE Network Magazine(2015- 2017) Nei Kato is a Distinguished Lecturer of IEEE Communications Society and Vehicular Technology Society. He is a fellow of The Engineering Academy of Japan, IEEE, IEICE, and a Clarivate Analytics highly cited researcher.

2168-6750 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Univ of Science and Tech Beijing. Downloaded on December 15,2021 at 01:02:31 UTC from IEEE Xplore. Restrictions apply.