# Federated Learning Based Content Popularity Prediction in Fog Radio Access Networks

Yanxiang Jiang, *Senior Member, IEEE*, Yuting Wu, Fu-Chun Zheng, *Senior Member, IEEE*,
Mehdi Bennis, *Fellow, IEEE*, and Xiaohu You, *Fellow, IEEE*

***Abstract*—In this paper, the content popularity prediction problem in fog radio access networks (F-RANs) is investigated. In order to obtain accurate prediction with low complexity, we propose a novel context-aware popularity prediction policy based on *federated learning* (FL). Firstly, user preference learning is applied by considering that users prefer to request the contents they are interested in. Then, users' context information is utilized to cluster users efficiently by adaptive context space partitioning. After that, we formulate a popularity prediction optimization problem to learn the local model parameters by using the stochastic variance reduced gradient (SVRG) algorithm. Finally, FL based model integration is proposed to learn the global popularity prediction model based on local models using the distributed approximate Newton (DANE) algorithm with SVRG. Our proposed popularity prediction policy not only can predict content popularity accurately, but also can significantly reduce computational complexity. Moreover, we theoretically analyze the convergence bound of our proposed FL based model integration algorithm. Simulation results show that our proposed policy increases the cache hit rate by up to 21.5 $\%$ compared to existing policies.**

***Index Terms*—F-RAN, popularity prediction, user preference learning, context-aware, federated learning.**

## I. Introduction

With the unprecedented rapid proliferation of intelligent devices, wireless networks are confronted with a myriad of challenges and notably data traffic pressure on the fronthaul wireless links [1]. To cope with this issue, fog radio access networks (F-RANs) have emerged as a promising solution to alleviate the traffic burden on fronthaul links by caching popular contents in fog access points (F-APs) [2]. In F-RANs, F-APs with limited caching and computing resources are densely deployed at network edges to provide reliable and stable service for users [3]. By placing contents in the F-APs which are closer to users, a mass of repeated transmission through fronthaul links can be avoided and traffic burden can be reduced [4]. Due to the caching capacity constraints, F-APs need to predict future content popularity accurately in order to prefetch the most popular contents during off-peak traffic periods and improve caching efficiency [5].

Traditional caching policies, such as least recently used (LRU) and least frequently used (LFU), are widely used in wired networks, but their efficiency is limited since content popularity is not considered [6]. When considering content popularity, most of the existing works on edge caching were carried out based on the assumption that the popularity of contents was known in advance, which is not practical. Predicting future content popularity based on available information is therefore of great importance [7]. Recently, improving caching efficiency by predicting content popularity accurately has gained significant interest. An auto-regressive (AR) model based content popularity prediction algorithm was proposed in [8] and the model parameters were learned by least square estimates. Similarly, an auto regressive and moving average (ARMA) model was shown to outperform the LFU caching scheme in [9]. In [10], a deep learning based content popularity prediction scheme was proposed. In [11], the authors proposed to learn popularity prediction model for each content class from the historical popularity series through training a simplified bidirectional long short-term memory (Bi-LSTM) network. In [12], the authors proposed popularity prediction methods to track the various trends with spatial-temporal content popularity and user dynamics based on the research in [11]. In [13], a multilevel probabilistic model was proposed and Bayesian learning was utilized to obtain the model parameters. In [14], a user preference model was proposed to predict content popularity and track the popularity change based on the user preference and the features of the requested content. However, these existing works except for [13] and [14] can not predict the popularity of newly-added or unseen contents whose statistical data are not available in advance. In other words, the prediction process for online contents can not be carried out until enough information about user requests has been collected. In [13], the knowledge of the prior distribution of the parameters is required. In [14], the authors focused more on edge caching than popularity prediction.

By considering the huge computation resources consumption during the model training process, it is impractical to accomplish the whole process on a separate device when

Manuscript received March 28, 2021, revised August 16, 2021, and accepted October 29, 2021. This work was supported in part by the National Natural Science Foundation of China under grant 61971129, the Natural Science Foundation of Jiangsu Province under grant BK20181264, the National Major Research and Development Program of China under Grant 2020YFB1805005, and the Shenzhen Science and Technology Program (Grants Nos. KQTD20190929172545139 and JCYJ20180306171815699).

Y. Jiang is with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China, and the School of Electronic and Information Engineering, Harbin Institute of Technology, Shenzhen 518055, China (e-mail: yxjiang@seu.edu.cn).

Y. Wu and X. You, are with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China (e-mail: 220180880@seu.edu.cn, xhyu@seu.edu.cn).

F. Zheng is with the School of Electronic and Information Engineering, Harbin Institute of Technology, Shenzhen 518055, China, and the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China (e-mail: fzheng@ieee.org).

M. Bennis is with Centre for Wireless Communications, University of Oulu, Oulu 90014, Finland (e-mail:mehdi.bennis@oulu.fi).

there is a huge amount of training data. Besides, it is not sustainable to transmit all the data to a centralized server because of the constrained network bandwidth. Therefore, a new machine learning framework called *Federated Learning* (FL) was proposed for training models in a distributed manner [15]. By adopting this technique, the model training process is distributed over a number of mobile user equipments (UEs) or edge nodes. Each participant contributes to the model by independently computing intermediate gradient values based on its local training data and sending model updates to the aggregator. The learning process of FL mainly includes two steps. In the *local update* step, each participant performs local computation based on its local data in order to update the model parameters broadcasted by the aggregator. In the *global aggregation* step, the aggregator collects the updated models sent by participants and aggregates them to generate an updated global model, which will be broadcasted in the next iteration. The above two steps are repeated until a predefined accuracy is achieved. FL has gained a wide range of interest in recent years. The reasons for supporting FL are as follows. First of all, FL is easy to implement with the rapid development of edge computing [16]. As the edge devices are equipped with a large amount of computing resources, it is unnecessary to collect all the data and complete the computing process in a centralized powerful server. Second, FL makes the process of data collection and model training more flexible and in general consumes less bandwidth. The edge devices can collect local data at any time, and contribute to the global model when needed.

Despite the benefits brought by FL, there are also many challenges to confront with, including learning time allocation, resource allocation, participant selection, etc. In order to deal with the above challenges, much research about FL has been conducted. In [16], for example, the authors proposed two main trade-offs: (i) between the computation and communication time determined by the predefined learning accuracy; (ii) between the learning time and UE energy consumption. In [17], the convergence bound of distributed gradient descent was analyzed theoretically and a control algorithm that determines the best tradeoff between local update and global aggregation was proposed to minimize the loss function under a given resource budget. In [18], the authors proposed a novel model aggregation approach to deal with the problem of bandwidth limitation by exploiting the natural signal superposition of a wireless multiple access channel. In [19], a differentially private asynchronous FL scheme for resource sharing in vehicular networks was proposed to protect the privacy of uploaded models. In [20], a dynamic incentive scheme for FL was proposed to adjust the participants and their level adaptively. In [21], a clustering-based asynchronous FL framework was proposed to adapt to the heterogeneity of industrial IoT. In [22], an effective incentive mechanism for FL was proposed to motivate high-reputation mobile devices with high-quality data to participate in model learning. In conclusion, there has been much research concerned with the development of FL.

Motivated by the aforementioned discussions, our main contributions are summarized below.

1) We propose a new popularity prediction policy based on user preference learning and adaptive context space partitioning. Unlike traditional approaches, the inputs of the prediction model are the popularity scores for contents of the clustered users, which are preprocessed by user preference learning and adaptive context space partitioning. Then, the popularity prediction model is learned automatically by training the model with historical popularity and the preprocessed inputs based on S-tochastic Variant Reduced Gradient (SVRG) Algorithm. Specifically, our proposed popularity prediction policy is effective in predicting the popularity of newly-added contents with no available statistical data.
2) We put forward an FL based model integration approach for global popularity prediction model construction. The proposed algorithm is carried out by combining SVRG with the Distribute Approximate Newton (DANE) Algorithm. This way, the global model can be generated based on local models in a distributed manner. Moreover, the computation and communication overhead is greatly reduced.
3) We analyze the convergence of our proposed FL based model integration approach by comparing it with a centralized method from a theoretical point of view. Based on the result obtained from theoretical analysis, we show the relationship between the frequency of global aggregation and the learning accuracy.
4) We validate our theoretical results with real data. Simulation results show that our proposed popularity prediction policy can predict future content popularity with high precision. Cache placement is performed based on the prediction results and higher cache hit rate is achieved in comparison with traditional approaches.

The remainder of this paper is organized as follows. In Section II, the system model is presented. The proposed popularity prediction policy is described in Section III. The theoretical performance analysis of our proposed FL based model integration approach is provided in Section IV. Simulation results are shown in Section V. Final conclusions are drawn in Section VI.

## II. System Model

As shown in Fig. 1, we consider the F-RAN in a specific region consisting of $M$ F-APs and $N$ users. Let $\mathcal{Q} = \{q_1, q_2, \ldots, q_m, \ldots, q_M\}$ denote the set of F-APs and $\mathcal{U} = \{u_1, u_2, \ldots, u_n, \ldots, u_N\}$ denote the set of users in the region. Every F-AP has its own coverage area, limited storage capacity and computing ability. Mobile users are associated with an F-AP and can be served by it when located in its coverage area. It is assumed that the users in the coverage area of each F-AP remain unchanged during the considered time period. Let $\mathcal{C} = \{c_1, c_2, \ldots, c_i, \ldots, c_I\}$ denote the content library, where $I$ is the cumulative number of contents at the current time. If user $u_n$ sends a request for content $c_i$ that is stored in its associated F-AP $q_m$, $c_i$ can be sent to $u_n$ directly from its local cache. Otherwise, $q_m$ needs to fetch $c_i$ from neighboring F-APs or the cloud server. Assume that all

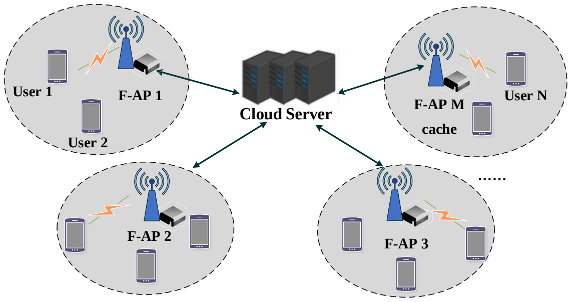

Fig. 1. Illustration of the scenario in F-RAN.

the contents have the same size[1], and the coverage areas of different F-APs do not overlap with each other.

The popularity of contents in F-APs is determined by the requests sent by the associated users. Let $req_i$ denote the cumulative user requests for content $c_i$. If an individual caching strategy in a single F-AP is considered, the local popularity prediction model of the F-AP is required; if a co-operative caching strategy among several F-APs is considered, the global popularity prediction model in the larger coverage area of these F-APs is required. Let $\hat{p}_i$ and $p_i$ denote the predicted and real popularity of $c_i$, respectively. The value of $p_i$ can be calculated as follows:

$$p_i = \frac{req_i}{\sum_{t=1}^{I} req_t}. \tag{1}$$

There exists deviation between $\hat{p}_i$ and $p_i$. Therefore, the mean-square error (MSE) is utilized to measure the accuracy of the prediction as follows:

$$\text{MSE} = \frac{1}{I}\sum_{i=1}^{I} |\hat{p}_i - p_i|^2. \tag{2}$$

The F-APs cache the most popular contents according to the predicted popularity. If a request from users is served by the cache of F-APs instead of by fetching the content from the cloud server through fronthaul links, a cache hit event occurs. Cache hit rate is defined as the ratio of the cache hits to the overall number of requests. It is utilized to evaluate the caching performance.

The objective of this paper is to find a content popularity prediction policy to predict future popularity accurately and minimize the MSE of the prediction results. For convenience, a summary of major notations is presented in Table I.

## III. Proposed Popularity Prediction Policy

In order to minimize the MSE of the prediction results, we propose a novel content popularity prediction policy, which includes offline user preference learning, adaptive context space partitioning, popularity prediction model construction and FL based model integration. The proposed policy can predict content popularity accurately and allow popular contents to be cached based on the prediction results.

### A. Policy Description

The procedure of the proposed popularity prediction policy comprises the following four steps, as shown in Fig. 2.

(1) *Offline user preference learning*: The popularity of contents will change constantly due to the variation of user preference. Therefore, user preference learning is the first step of the proposed policy. User preference learning is conducted independently by training the local data at the UE in an offline manner. If the user preference is similar to the content feature, the probability for the user to request the content will be high, leading to a higher popularity score for the content [23].

(2) *Adaptive context space partitioning*: After offline user preference learning, users send the preference and context information to their associated F-APs. Context information of users includes gender, age, personality, occupation, location, equipment, etc. Then, the F-APs can take the correlation between user activity levels and context information into account, since users sharing similar context are more likely to have similar activity levels. As a consequence, adaptive context space partitioning is applied as the second step in the proposed policy to cluster users effectively [24].

(3) *Popularity prediction model construction*: In the third step, a popularity prediction model is constructed based on the obtained popularity scores of users in the context subspaces after partitioning. The model parameters can be learned by training using the historical data.

(4) *Federated learning based model integration*: If the local popularity prediction models in F-APs have been obtained, a global model consisting of these F-APs can be generated by integrating the local models in a distributed manner [25]. By using this approach, computational complexity can be reduced and the bandwidth for transmitting local data can be saved.

The proposed popularity prediction policy is illustrated in Fig. 2. The details of the four steps of the policy will now be presented below.

### B. Offline User Preference Learning

The contents in the library are not constant, since there is a large amount of contents uploaded every day. The features of content may consist of content categories or labels and each content has different weights for different features. Let $\boldsymbol{\chi}_i = [\chi_{i1}, \chi_{i2}, \ldots, \chi_{ij}, \ldots, \chi_{iJ}]^T$ denote the content feature vector of $c_i$, where $\chi_{ij}$ is the weight of the corresponding content feature and $J$ is the number of content features. Let $\boldsymbol{w}_n = [w_{n1}, w_{n2}, \ldots, w_{nj}, \ldots, w_{nJ}]^T$ denote the user preference vector of user $u_n$, where $w_{nj}$ is the user preference probability of the corresponding content feature. User preference can be learned independently without interaction with each other. Therefore, user preference learning is flexible, which can be conducted at any idle time.

There is a large amount of records of user requests in UE which can be used as training samples. Let $\mathcal{A}_n = \{(\boldsymbol{\chi}_i, y_{n,i}), i \in [1, I]\}$ denote the set of training samples for user $u_n$, where $y_{n,i}$ is the binary request label. If $u_n$ has requested content $c_i$, $y_{n,i} = 1$; otherwise $y_{n,i} = 0$. Due to the offline nature, the privacy of users can be protected since

[1] Note that contents with different sizes can always be split into data segments of the same size, and each data segment can then be considered as a "content". For convenience, we also split the cache space into cache units of the same size and set the size of a cache unit equal to the size of a content.

TABLE I. SUMMARY OF MAJOR NOTATIONS.

| | |
|---|---|
| $M$, $\mathcal{Q}$ | Number of F-APs, set of $M$ F-APs |
| $N$, $\mathcal{U}$ | Number of users, set of $N$ users |
| $I$, $\mathcal{C}$ | Number of contents, set of $I$ contents |
| $\hat{p}_i$, $p_i$ | Predicted popularity of content $c_i$, real popularity of content $c_i$ |
| $\boldsymbol{\chi}_i$ | The content feature vector of content $c_i$ |
| $J$ | Dimension of content feature vector |
| $\boldsymbol{w}_n$, $w_{n,j}$ | The user preference vector of $u_n$, the preference of $u_n$ for feature $j$ |
| $\mathcal{A}_n = \{(\boldsymbol{\chi}_i, y_{n,i})\}$, $y_{n,i}$ | The set of training samples for $u_n$, the binary request label of $u_n$ for $c_i$ |
| $\boldsymbol{\zeta}_n$, $D$ | The context vector of $u_n$, dimension of context spaces |
| $\Theta_1, \Theta_2, \ldots, \Theta_s, \ldots \Theta_S$ | The set of context subspaces after partitioning |
| $\boldsymbol{x}_i = [x_{1,i}, x_{2,i}, \ldots, x_{S,i}]^T$ | The popularity scores for $c_i$ of all the context subspaces after partitioning |
| $\boldsymbol{a} = [a_1, a_2, \ldots, a_s, \ldots, a_S]^T$ | The vector of the parameters of the popularity prediction model |
| $\mathcal{B}_m = \{(p_i, \boldsymbol{x}_i), i \in [1, I]\}$ | The set of the training samples in the considered F-AP $q_m$ |
| $\boldsymbol{a}_m(t)$, $\boldsymbol{a}(t)$ | The vector of model parameters in F-AP $q_m$ learned by federated learning, the vector of global model parameters learned by federated learning. |
| $\boldsymbol{a}_{[l]}(t)$ | The vector of global model parameters for $t \in [(l-1)\tau, l\tau]$ learned by centralized learning. |

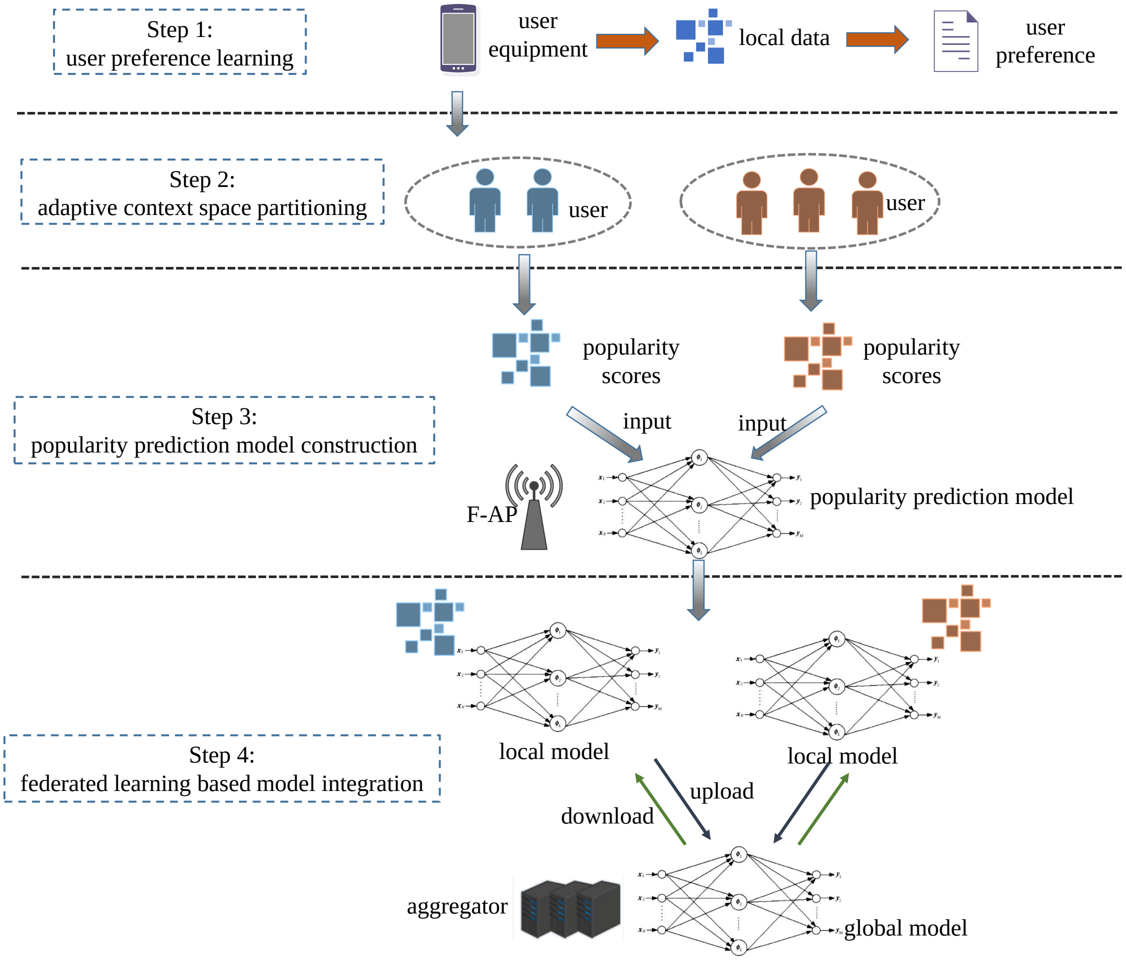

Fig. 2. Illustration of the proposed popularity prediction policy.

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/TWC.2021.3124586, IEEE Transactions on Wireless Communications

numerous user data are kept in private equipment instead of being transmitted to the central server.

Sigmoid function is used to approximate the correspondence between the feature vector and request label of contents. For simplicity, let $h_{\boldsymbol{w}_n}(\boldsymbol{\chi}_i) = 1/(1+e^{-\boldsymbol{w}_n^T \cdot \boldsymbol{\chi}_i})$. The probability function of $u_n$ is formulated to represent the probability for $y_{n,i}$ as follows:

$$p_{\boldsymbol{w}_n}(y_{n,i}|\boldsymbol{\chi}_i) = h_{\boldsymbol{w}_n}(\boldsymbol{\chi}_i)^{y_{n,i}} (1 - h_{\boldsymbol{w}_n}(\boldsymbol{\chi}_i))^{1-y_{n,i}}. \quad (3)$$

Given enough samples, the likelihood function of $u_n$ can be expressed as follows [26]:

$$L(\boldsymbol{w}_n) = \prod_{(\boldsymbol{\chi}_i, y_{n,i}) \in \mathcal{A}_n} p_{\boldsymbol{w}_n}(y_{n,i}|\boldsymbol{\chi}_i). \quad (4)$$

Since maximizing likelihood estimation (MLE) is equivalent to minimizing the negative likelihood function [27], the cross entropy loss function is formulated as the negative log-likelihood function as follows:

$$\begin{aligned} F(\boldsymbol{w}_n) &= -\frac{1}{|\mathcal{A}_n|} \cdot \ln L(\boldsymbol{w}_n) \\ &= \frac{1}{|\mathcal{A}_n|} \sum_{(\boldsymbol{\chi}_i, y_{n,i}) \in \mathcal{A}_n} \left[ \ln \left( 1 + e^{\boldsymbol{w}_n^T \cdot \boldsymbol{\chi}_i} \right) - y_{n,i} \cdot \boldsymbol{w}_n^T \cdot \boldsymbol{\chi}_i \right]. \end{aligned} \quad (5)$$

Consequently, $\boldsymbol{w}_n$ can be obtained by minimizing $F(\boldsymbol{w}_n)$ as follows:

$$\boldsymbol{w}_n = \underset{\boldsymbol{w}_n \in \mathbb{R}^J}{\arg\min} F(\boldsymbol{w}_n). \quad (6)$$

The Follow The (Proximally) Regularized Leader (FTRL-Proximal) algorithm [14] is adopted to learn the user preference. In addition, user preference needs to be updated dynamically when the change of the objective function $F(\boldsymbol{w}_n)$ exceeds a certain threshold.

### C. Adaptive Context Space Partitioning

User preference may differ based on their context information [28]. For example, the movie types favored by boys and girls, young and old, are usually different. Moreover, the activity levels of users are likely to be related to their context information. It is quite intuitive that the young people have higher activity levels than the old because the young is more accustomed to using smart devices. Therefore, we propose to partition the context space uniformly into parts of similar contexts for further popularity prediction. In addition, popularity scores for contents can be learned independently in each context subspace. After partitioning, the users in the same context subspace having similar activity levels can be treated as a group. By using this approach, the number of items to track is significantly reduced. As a consequence, the computational complexity is maintained to a manageable extent.

Let $\boldsymbol{\zeta}_n = [\zeta_{n,1}, \zeta_{n,2}, \ldots, \zeta_{n,d}, \ldots, \zeta_{n,D}]^T \in [0,1]^D$ denote the context vector of user $u_n$, where $\zeta_{n,d}$ is the normalized value for the corresponding context information and $D$ is the dimension of context space. The process of adaptive context space partitioning is illustrated in Fig. 3 and Algorithm 1, which is inspired by the research in [29]. The original context space $\Theta_0$ is constructed and normalized as $[0,1]^D$, and its level $l_0 = 0$. All the users considered are included in $\Theta_0$. Consider $\Theta_i$ which is to be partitioned. Let $\text{Num}(\Theta_i)$ denote the number of users in $\Theta_i$ and $l_i$ be the level of $\Theta_i$. If $\text{Num}(\Theta_i)$ exceeds a certain threshold, $\Theta_i$ is split into $2^D$ subspaces. As one of these child subspaces, $\Theta_j$ has an increased level of $l_j = l_i + 1$. The threshold determines the rate at which the context space is partitioned. Therefore, it needs to be carefully designed. We design the threshold to have the form $r_1 \cdot 2^{r_2 l_i}$, where $r_1 > 0$ and $r_2 > 0$ are hyper-parameters in Algorithm 1. Due to this splitting process, a subspace of level $l_i$ has length $2^{-l_i}$ along each axis. For clarity of description, we rename the final obtained subspaces as $\Theta_1, \Theta_2, \ldots, \Theta_s, \ldots \Theta_S$, where $S$ is the cumulative number of context subspaces. Users are then clustered according to the partitioning results.

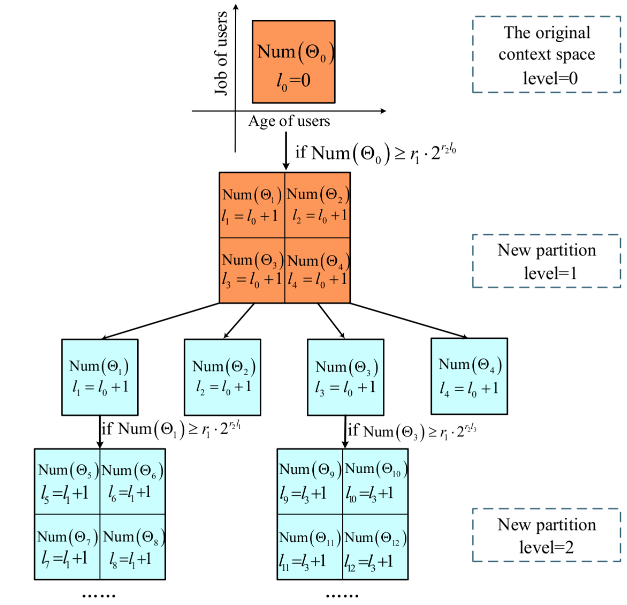
Job of users
Age of users
$\text{Num}(\Theta_0)$
$l_0$=0
The original context space level=0
if $\text{Num}(\Theta_0) \geq r_1 \cdot 2^{r_2 l_0}$
$\text{Num}(\Theta_1)$ $l_1 = l_0 + 1$
$\text{Num}(\Theta_2)$ $l_2 = l_0 + 1$
$\text{Num}(\Theta_3)$ $l_3 = l_0 + 1$
$\text{Num}(\Theta_4)$ $l_4 = l_0 + 1$
New partition level=1
if $\text{Num}(\Theta_1) \geq r_1 \cdot 2^{r_2 l_1}$
if $\text{Num}(\Theta_3) \geq r_1 \cdot 2^{r_2 l_3}$
$\text{Num}(\Theta_5)$ $l_5 = l_1 + 1$
$\text{Num}(\Theta_6)$ $l_6 = l_1 + 1$
$\text{Num}(\Theta_7)$ $l_7 = l_1 + 1$
$\text{Num}(\Theta_8)$ $l_8 = l_1 + 1$
$\text{Num}(\Theta_9)$ $l_9 = l_3 + 1$
$\text{Num}(\Theta_{10})$ $l_{10} = l_3 + 1$
$\text{Num}(\Theta_{11})$ $l_{11} = l_3 + 1$
$\text{Num}(\Theta_{12})$ $l_{12} = l_3 + 1$
New partition level=2
......
......

Fig. 3. Illustration of adaptive context space partitioning ($D$ = 2).

**Algorithm 1** Adaptive Context Space Partitioning

```
Input: $\boldsymbol{\zeta}_n = [\zeta_{n,1}, \zeta_{n,2}, \ldots \zeta_{n,D}]^T$, $D$, $r_1$, $r_2$, $\psi$ = number of the total subspaces
Output: $\Theta_1, \Theta_2, \ldots, \Theta_s, \ldots \Theta_S$
initialize $i = 0$, $l_0 = 0$, $\psi = 1$
while $i < \psi$ do
    if $\text{Num}(\Theta_i) \geq r_1 \cdot 2^{r_2 l_i}$ then
        split $\Theta_i$ into $2^D$ subspaces $\Theta_j$
        $\psi = \psi + 2^D$
        $l_j = l_i + 1$
    end if
    $i = i + 1$
end while
```

Since user preference has been obtained, sigmoid function $h_{\boldsymbol{w}_n}(\boldsymbol{\chi}_i)$ can be used as the mapping function $f(\boldsymbol{w}_n, \boldsymbol{\chi}_i)$ to map the correlation of user preference and content feature to popularity scores. The popularity score $x_{s,i}$ of $\Theta_s$ for content $c_i$ is obtained based on the preference of all the users in $\Theta_s$

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/TWC.2021.3124586, IEEE Transactions on Wireless Communications

as follows:

$$x_{s,i} = \frac{1}{W}\sum_{\boldsymbol{\zeta}_n\in\Theta_s} f\left(\boldsymbol{w}_n, \boldsymbol{\chi}_i\right), \tag{7}$$

where $W$ is the total number of user requests in the coverage area of the F-AP. Let $\boldsymbol{x}_i = \left[x_{1,i}, x_{2,i}, \ldots, x_{S,i}\right]^T$ denote the popularity scores for content $c_i$ of all the context subspaces after partitioning. The popularity scores are used as the inputs of the popularity prediction model in the following subsection. By preprocessing the inputs of the model through adaptive context space partitioning, the number of model parameters will decrease significantly.

### *D. Popularity Prediction Model Construction*

Without loss of generality, we consider the local popularity prediction model in a typical F-AP. Let $\boldsymbol{a} = \left[a_1, a_2, \ldots, a_s, \ldots, a_S\right]^T$ denote the vector of model parameters. By using $\boldsymbol{x}_i$ as the input of the prediction model, $\hat{p}_i$ can be expressed as follows:

$$\hat{p}_i = \boldsymbol{a}^T\boldsymbol{x}_i = a_1x_{1,i} + a_2x_{2,i} + \cdots + a_Sx_{S,i}. \tag{8}$$

The least square method is applied to learn $\boldsymbol{a}$. Let $\mathcal{B} = \{(p_i, \boldsymbol{x}_i), i\in[1,I]\}$ denote the set of training samples in the F-AP. The model parameters can be obtained by minimizing MSE as follows:

$$\begin{array}{ll} \min & \frac{1}{L}\sum_{(p_i,\boldsymbol{x}_i)\in\mathcal{B}}\left|p_i - \boldsymbol{a}^T\boldsymbol{x}_i\right|^2 \\ \text{s.t.} & a_s \geq 0, \forall s\in[1,S] \end{array}, \tag{9}$$

where $L$ is the number of training samples in the F-AP. The model is trained with historical popularity and historical popularity scores in context subspaces.

The elements of $\boldsymbol{a}$ in the prediction model represent the activity levels of the users in the corresponding context subspaces [30]. It implies that their contributions to the network traffic differ from each other. The SVRG algorithm is adopted to solve the optimization problem in (9) [31]. It is a variant of stochastic gradient descent (SGD) with explicit variance reduction, which can achieve faster convergence. Let $v_i(\boldsymbol{a}) = \left|p_i - \boldsymbol{a}^T\boldsymbol{x}_i\right|^2$, $V(\boldsymbol{a}) = \frac{1}{L}\sum_{(p_i,\boldsymbol{x}_i)\in\mathcal{B}} v_i(\boldsymbol{a})$. The detailed procedure of SVRG is shown in Algorithm 2. Specifically, the algorithm operates in two nested loops. In the outer loop, it computes the gradient value of the entire function (Line 4 in Algorithm 2). In the inner loop where $\tau$ fast stochastic updates are performed (Line 6-10 in Algorithm 2), the gradient in iteration $t$ is calculated as follows:

$$\boldsymbol{g}_t = \bigtriangledown v_i\left(\boldsymbol{a}_t\right) - \left(\bigtriangledown v_i\left(\boldsymbol{a}^j\right) - \bigtriangledown V\left(\boldsymbol{a}^j\right)\right), \tag{10}$$

where $\left(\bigtriangledown v_i\left(\boldsymbol{a}^j\right) - \bigtriangledown V\left(\boldsymbol{a}^j\right)\right)$ is regarded as the bias of the gradient estimate $\bigtriangledown v_i\left(\boldsymbol{a}_t\right)$. By adjusting the number of stochastic steps, the tradeoff between convergence rate and computational complexity can be flexibly balanced.

**Algorithm 2** Stochastic Variance Reduced Gradient (SVRG)

```
Input: $\boldsymbol{x}_i = [x_{1,i}, x_{2,i}, \ldots, x_{S,i}]^T, \forall i\in[1,I]$
Output: $\boldsymbol{a} = [a_1, a_2, \ldots, a_s, \ldots, a_S]^T$
 1: initialize $\tau$ = number of stochastic steps per epoch, $\eta$ = stepsize, $\boldsymbol{a}^0$
 2: for $j = 0, 1, 2, \cdots$ do
 3:     compute and store                    ▷ Full pass through data
 4:     $\bigtriangledown V(\boldsymbol{a}^j) = \frac{1}{L}\sum_{(p_i,\boldsymbol{x}_i)\in\mathcal{B}} \bigtriangledown v_i(\boldsymbol{a}^j)$
 5:     set $\boldsymbol{a}_1 = \boldsymbol{a}^j$
 6:     for $t = 1$ to $\tau$ do
 7:         pick $(p_i, \boldsymbol{x}_i)\in\mathcal{B}$ uniformly at random
 8:         calculate $\boldsymbol{g}_t$ according to (10)
 9:         $\boldsymbol{a}_{t+1} = \boldsymbol{a}_t - \eta\boldsymbol{g}_t$          ▷ Stochastic update
10:     end for
11:     $\boldsymbol{a}^{j+1} = \boldsymbol{a}_{\tau+1}$
12: end for
```

### *E. Federated Learning Based Model Integration*

The global popularity prediction model aggregated by $K$ F-APs is considered[2]. Without loss of generality, let $\{q_1, q_2, \cdots, q_m, \cdots, q_K\}$ denote the set of considered F-APs. In order to learn the global model, the cloud server needs to solve the optimization problem as follows:

$$\begin{array}{ll} \min & \frac{1}{L^*}\sum_{(p_i,\boldsymbol{x}_i)\in\mathcal{B}^*}\left|p_i - \boldsymbol{a}^T\boldsymbol{x}_i\right|^2 \\ \text{s.t.} & a_s \geq 0, \forall s\in[1,S] \end{array}, \tag{11}$$

where $L^*$ denotes the total number of training samples, $\mathcal{B}^*$ denotes the corresponding set of training samples. For simplicity, let $V^*(\boldsymbol{a}) = \frac{1}{L^*}\sum_{(p_i,\boldsymbol{x}_i)\in\mathcal{B}^*}\left|p_i - \boldsymbol{a}^T\boldsymbol{x}_i\right|^2$. Due to the expanded coverage area and the increased number of associated users compared with the local model, it is obvious that $L^* \gg L$. As a consequence, the computational load for determining the global model parameters be extremely high. In addition, the training data in the distributed F-APs needs to be transmitted to the cloud server, which will consume the bandwidth resources. To cope with these issues, FL based model integration is proposed to generate the global model based on local models in a distributed manner.

Let $\mathcal{B}_m$ denote the set of training samples in F-AP $q_m$ and $L_m$ be the number of training samples in F-AP $q_m$. The optimized objective function in (11) can be expressed as follows[3]:

$$V^*(\boldsymbol{a}) = \sum_{m=1}^{K}\frac{L_m}{L^*}\cdot\frac{1}{L_m}\sum_{(p_i,\boldsymbol{x}_i)\in\mathcal{B}_m} v_i(\boldsymbol{a}). \tag{12}$$

The MSE in $q_m$ can be expressed as $V_m(\boldsymbol{a}) = \frac{1}{L_m}\sum_{(p_i,\boldsymbol{x}_i)\in\mathcal{B}_m} v_i(\boldsymbol{a})$. Therefore, the optimization problem in (11) can be converted to the following equivalent form:

$$\begin{array}{ll} \min & \sum_{m=1}^{K}\frac{L_m}{L^*}V_m(\boldsymbol{a}) \\ \text{s.t.} & a_s \geq 0, \forall s\in[1,S] \end{array}. \tag{13}$$

It indicates that the global optimization is determined by local optimizations. The most intuitive solution is to make each F-

[2] The value of $K$ can be set according to practical requirements.

[3] The popularity of contents in different F-APs are different due to the diversity of user preference and user activity levels. The global popularity of the $K$ F-APs should be recalculated based on local popularity and user requests. However, in order to solve the optimization problem in a distributed way, we use (12) to approximate the global optimization problem which is expressed by the recalculated global popularity.

**Algorithm 3** Federated Learning Based Model Integration

```
Input: $\boldsymbol{x}_i = [x_{1,i}, x_{2,i}, \ldots, x_{S,i}]^T$, $\tau$ =number of stochastic
    steps per epoch, $h$ =stepsize
Output: $\boldsymbol{a} = [a_1, a_2, \ldots, a_s, \ldots, a_S]^T$
initialize $\tau$ = number of stochastic steps per epoch, $\eta$ =
    stepsize, $\boldsymbol{a}^0$
for $j = 0, 1, 2, \cdots$ do                              ▷ Overall iterations
    compute and store
    $\bigtriangledown V^*\left(\boldsymbol{a}^j\right) = \frac{1}{L^*}\sum_{(p_i,\boldsymbol{x}_i)\in\mathcal{B}^*} \bigtriangledown v_i\left(\boldsymbol{a}^j\right)$
    for $m = 1, 2, \cdots, K$ do                          ▷ Distributed loop
        (in parallel over nodes)
        initialize $\boldsymbol{a}_m^{(1)} = \boldsymbol{a}^j$
        for $t = 1$ to $\tau$ do                          ▷ Actual update loop
            sample $(p_i, \boldsymbol{x}_i) \in \mathcal{B}_m$ uniformly at random
            calculate $\boldsymbol{g}_m^t$ according to (15)
            $\boldsymbol{a}_m^{(t+1)} = \boldsymbol{a}_m^{(t)} - \eta \boldsymbol{g}_m^t$
        end for
    end for
    $\boldsymbol{a}^{j+1} = \boldsymbol{a}^j + \frac{1}{K}\sum_{m=1}^{K}(\boldsymbol{a}_m^{(\tau+1)} - \boldsymbol{a}^j)$     ▷ Aggregate
end for
```

AP minimize its local function, and then average the results of all the F-APs. However, it is impractical unless the local model parameters are all the same. The DANE algorithm [31] is applied to remedy the above method by modifying the local problems before each aggregation step.

*Theorem 1:* A feasible solution is to perturb the local function $V_m(\boldsymbol{a})$ of $q_m$ in each iteration $j$ with the disturbance term: $-(\boldsymbol{b}_m^j)^T\boldsymbol{a}$. It means that in each iteration $j$, the F-APs parallelly solve the optimization problem as follows:

$$\boldsymbol{a}_m^{j+1} = \underset{\boldsymbol{a}\in\mathbb{R}^S}{\arg\min}\left\{V_m(\boldsymbol{a}) - (\boldsymbol{b}_m^j)^T\boldsymbol{a}\right\}, \tag{14}$$

where $\boldsymbol{a}_m^{j+1}$ denotes the vector of model parameters of $q_m$ in iteration $j+1$. The value of $\boldsymbol{b}_m^j$ can be calculated as $(\bigtriangledown V_m(\boldsymbol{a}^j) - \bigtriangledown V^*(\boldsymbol{a}^j))$ [31].

*Proof:* Please see Appendix A.

As described in the previous subsection, (14) can be solved with SVRG. Consequently, FL based model integration is proposed to incorporate DANE and SVRG into learning the popularity prediction model. By using this approach, the global model can be learned in a distributed way. Moreover, the convergence rate can be improved by adopting SVRG. The detailed description is shown in Algorithm 3. Let $\boldsymbol{g}_m^t$ denote the gradient in iteration $t$ in $q_m$. In the actual update loop, $\boldsymbol{g}_m^t$ is calculated by applying (10) to solving (14) as follows:

$$\begin{aligned}\boldsymbol{g}_m^t &= [\bigtriangledown v_i\left(\boldsymbol{a}_m^{(t)}\right) - (\bigtriangledown V_m(\boldsymbol{a}^j) - \bigtriangledown V^*(\boldsymbol{a}^j))] \\ &\quad - [\bigtriangledown v_i(\boldsymbol{a}^j) - (\bigtriangledown V_m(\boldsymbol{a}^j) - \bigtriangledown V^*(\boldsymbol{a}^j))] + \bigtriangledown V^*(\boldsymbol{a}^j) \\ &= \bigtriangledown v_i\left(\boldsymbol{a}_m^{(t)}\right) - \bigtriangledown v_i\left(\boldsymbol{a}^j\right) + \bigtriangledown V^*\left(\boldsymbol{a}^j\right).\end{aligned} \tag{15}$$

### *F. Computational Complexity*

The proposed policy can reduce computational complexity in the following three aspects. Firstly, we divide users into groups that have similar activity levels and in the same context subspace. In this way, we can process these groups instead of tracking each user, which reduces computational complexity. Secondly, SVRG is used in our federated learning, which can avoid traversing all dataset in each iteration. Refer to [31], let $n = \rho/\beta$ be the condition number of the loss function. For the traditional SGD method, to achieve a precision of $\epsilon$, $n^2\ln(1/\epsilon)$ data samples are needed to calculate their gradients, whereas our adopted SVRG method only needs to process $n\ln(1/\epsilon)$ data samples, thereby reducing the gradient calculation process of $\left(n^2 - n\right)\ln(1/\epsilon)$ samples. Finally, the characteristics of federated learning determines that retraining the global model in each round of updates is not needed, but only requires to aggregate the model updates uploaded by each UE. In addition, federated learning makes full use of the computing power of each UE.

According to the above descriptions, the proposed popularity prediction policy not only predicts popularity of existing or newly-added contents with a high accuracy, but also significantly reduces the computational complexity and communication overhead. Nevertheless, the popularity prediction models need to be updated dynamically when the MSE exceeds a predefined threshold.

## IV. Performance Analysis of Federated Learning

In this section, the performance of our proposed FL based popularity prediction policy will be analyzed. Let $\boldsymbol{a}^f$ denote the final vector of model parameters obtained from Algorithm 3. Let $\boldsymbol{a}^*$ denote the optimal solution of problem (13). The convergence of Algorithm 3 will be investigated and the upper bound of $V(\boldsymbol{a}^f) - V(\boldsymbol{a}^*)$ will be derived[4].

According to the above descriptions, FL is applied to the model integration step of the proposed popularity prediction policy. However, it is confronted with lots of constraints in practical applications, including limited communications bandwidth and computation resources [17]. Taking these limitations into consideration, the hyper-parameters in Algorithm 3 should be well designed, especially the value of $\tau$. As illustrated in Fig. 4, the process of FL mainly consists of two parts: local computation and communications, which are carried out iteratively. In each communications round, uploading and downloading model updates costs communications resources. Between two neighbouring communications rounds, each F-AP performs local computation, which costs computation resources. From Algorithm 3, the value of $\tau$ determines the frequency of global aggregation, which thereby will determine the tradeoff between the costs of communications and computation. Therefore, the value of $\tau$ needs to be designed carefully, because it has a direct impact on the performance of FL under given resource budget.

In order to compare the performance of FL with the optimal case, we assume a scenario in which the vector of model parameters is trained in a centralized manner [32]. As shown in Fig. 4, we can divide the FL process into $Z$ intervals,

[4] For simplicity, we use $V(\boldsymbol{a})$ to denote the global optimized objective function among all the considered F-APs, which is denoted as $V^*(\boldsymbol{a})$ in (12).

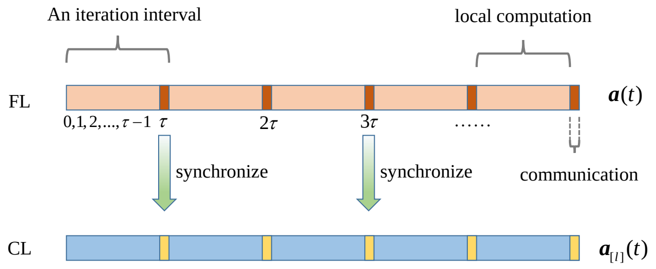

Fig. 4. Time allocation for federated learning.

where $Z$ is the total number of communications rounds. Let $[l]$ denote the interval $[(l-1)\tau, l\tau]$ , for $l = 1, 2, ..., Z$. Let $\boldsymbol{a}_{[l]}(t)$ be an auxiliary parameter vector in interval $[l]$ in the scenario of centralized learning (CL). At the beginning of each interval, $\boldsymbol{a}_{[l]}(t)$ is synchronized with $\boldsymbol{a}(t)$ which is trained by FL. In general, $\boldsymbol{a}_{[l]}(t)$ can be updated by gradient descent (GD) as follows:

$$\boldsymbol{a}_{[l]}(t) = \boldsymbol{a}_{[l]}(t-1) - \eta \bigtriangledown V(\boldsymbol{a}_{[l]}(t-1)), \tag{16}$$

in which $\boldsymbol{a}_{[l]}(t)$ is only defined for $t \in [(l-1)\tau, l\tau]$.

In order to derive the upper bound of $V(\boldsymbol{a}^f) - V(\boldsymbol{a}^*)$, we make the following assumption.

*Assumption 1*: We assume the following for all $i$:

1) $v_i(\boldsymbol{a})$ is convex;
2) $v_i(\boldsymbol{a})$ is $\rho$-Lipschitz, i.e., $\|v_i(\boldsymbol{a}_1) - v_i(\boldsymbol{a}_2)\| \leq \rho\|\boldsymbol{a}_1 - \boldsymbol{a}_2\|$ for any $\boldsymbol{a}_1$, $\boldsymbol{a}_2$;
3) $v_i(\boldsymbol{a})$ is $\beta$-smooth, i.e., $\| \bigtriangledown v_i(\boldsymbol{a}_1) - \bigtriangledown v_i(\boldsymbol{a}_2)\| \leq \beta\|\boldsymbol{a}_1 - \boldsymbol{a}_2\|$ for any $\boldsymbol{a}_1$, $\boldsymbol{a}_2$;
4) There exists an upper bound of the gradient divergence between $v_i(\boldsymbol{a})$ and the global loss function $V(\boldsymbol{a})$, i.e., $\| \bigtriangledown v_i(\boldsymbol{a}) - \bigtriangledown V(\boldsymbol{a})\| \leq \delta$ for any $i$, and $\boldsymbol{a}$.

According to the linear characteristic of the prediction model in (8), Assumption 1 is satisfied. Then, it is easy to infer that $V_m(\boldsymbol{a})$ and $V(\boldsymbol{a})$ are also convex, $\rho$-Lipschitz, and $\beta$-smooth.

*Lemma 1:* For any interval $[l]$ and $t \in [l]$, the upper bound of the difference between the parameters obtained from CL and FL respectively can be defined as follows:

$$\|\boldsymbol{a}_m(t) - \boldsymbol{a}_{[l]}(t)\| \leq g(t-(l-1)\tau), \tag{17}$$

where

$$g(x) \triangleq \frac{2\delta}{\beta}((\eta\beta+1)^x - 1), \tag{18}$$

for any $x = 0, 1, 2, ....$

*Proof:* Please see Appendix B.

*Theorem 2:* For any interval $[l]$ and $t \in [l]$, the upper bound of the difference between the parameters obtained from CL and FL respectively can be expressed as follows:

$$\|\boldsymbol{a}(t) - \boldsymbol{a}_{[l]}(t)\| \leq g(t-(l-1)\tau), \tag{19}$$

in which $g(x)$ is defined as in (18).

*Proof:* Please see Appendix C.

According to Theorem 2, we can see that the difference of the parameters learned from CL and FL is proportional to the gradient divergence $\delta$. If $\delta$ gets smaller, the training samples in different F-APs are more similar, which distinctly will make the difference of the parameters obtained from CL and FL smaller. In extreme cases, if the training samples in F-APs are all the same, apparently $\delta = 0$ and the upper bound of $\|\boldsymbol{a}(t) - \boldsymbol{a}_{[l]}(t)\|$ will be 0. In addition, we can see that as $x$ increases, the value of the function $g(x)$ also increases. It means that the upper bound on the difference between $\boldsymbol{a}(t)$ and $\boldsymbol{a}_{[l]}(t)$ will be higher over time.

Since $V(\boldsymbol{a})$ is also $\rho$-Lipschitz, we can derive that $V(\boldsymbol{a}(t)) - V(\boldsymbol{a}_{[l]}(t)) \leq \rho\|\boldsymbol{a}(t) - \boldsymbol{a}_{[l]}(t)\| \leq \rho g(t-(l-1)\tau)$. When $t = (l-1)\tau$, $g(t-(l-1)\tau) = 0$, $\|\boldsymbol{a}(t) - \boldsymbol{a}_{[l]}(t)\| \leq 0$. It means that at the beginning of the interval, $\boldsymbol{a}(t) = \boldsymbol{a}_{[l]}(t)$. This is consistent with the definition of $\boldsymbol{a}_{[l]}(t)$.

*Theorem 3:* When all the following conditions are satisfied:

1) $\eta\beta < 1$;
2) $(1 - \frac{\beta\eta}{2})\sigma\eta > \frac{\rho g(\tau)}{\tau\epsilon}$;
3) $V(\boldsymbol{a}_{[l]}(t)) - V(\boldsymbol{a}^*) \geq \epsilon$ for all $l$;
4) $V(\boldsymbol{a}^f) - V(\boldsymbol{a}^*) \geq \epsilon$,

for some $\epsilon > 0$, in which we define $\sigma \triangleq (\min_l \frac{1}{\|\boldsymbol{a}_{[l]}((l-1)\tau) - \boldsymbol{a}^*\|})$, then the convergence upper bound of $V(\boldsymbol{a}^f) - V(\boldsymbol{a}^*)$ is given by:

$$V(\boldsymbol{a}^f) - V(\boldsymbol{a}^*) \leq \frac{1}{T(\eta\sigma(1-\frac{\beta\eta}{2}) - \frac{\rho g(\tau)}{\tau\epsilon^2})}. \tag{20}$$

*Proof:* Please see Appendix D.

According to Theorem 3, the convergence upper bound of $V(\boldsymbol{a}^f) - V(\boldsymbol{a}^*)$ is affected by many parameters. As shown in (20), when the value of $\tau$ gets larger, the value of $\frac{g(\tau)}{\tau}$ will become larger, making the convergence upper bound of $V(\boldsymbol{a}^f) - V(\boldsymbol{a}^*)$ higher. The reason is that when $\tau$ gets smaller, the training process of FL will be closer to CL.

## V. Simulation Results

To evaluate the performance of the proposed popularity prediction policy, we perform simulations based on the data extracted from the MovieLens Dataset [33]. The MovieLens 100K Dataset contains 100, 000 ratings of 943 users on 1682 movies. Each user has rated at least 20 movies. Each data set entry consists of an anonymous user ID, a movie ID, a rating (1-5) and a timestamp. We assume that the ratings correspond to the number of requests from users. Therefore, a rating matrix is created. In addition, demographic information about the users is provided in the dataset, including their gender, age, occupation and Zip-code. For numerical evaluations, we select gender and age as context information. Besides, the genres of the movies are provided, which can be used as the content features. In the simulations, we set the parameters in the algorithms as follows: $r_1 = 0.5$, $r_2 = 1$, $\alpha = 0.5$, $\beta = 1$, $\eta = 0.01$, $K = 5$.

In Fig. 5, we show the logarithmic root mean-square error (RMSE) of our proposed popularity prediction policy and the AR model based policy with different number of considered contents [8]. It can be observed that the RMSE value of the proposed policy gradually decreases as the number of contents increases. The reason is that the proposed policy can better learn the relationship between popularity and content

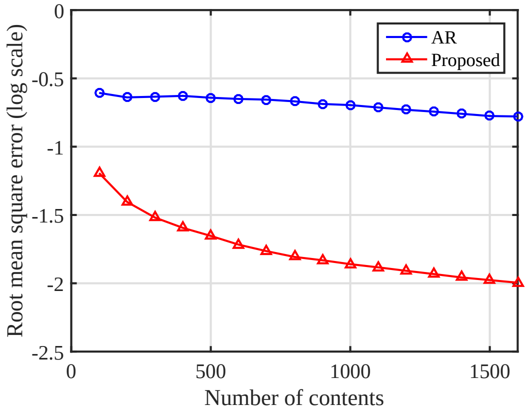

Fig. 5. Root mean-square error versus number of contents.

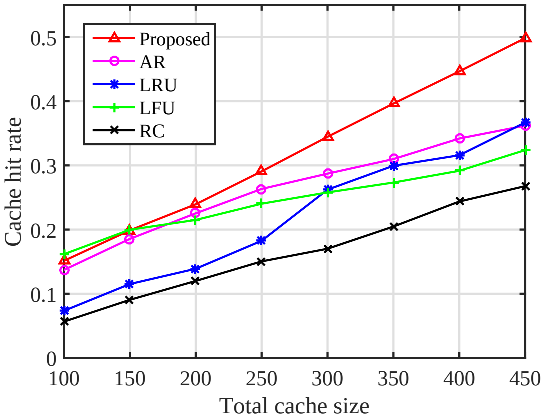

Fig. 6. Cache hit rates versus total cache size.

features with more training samples. It can also be observed that the RMSE of the proposed policy is smaller than the AR model based policy. The reason is that the inputs of these two policies are different. The input of the proposed policy captures the internal characteristics of popularity after a series of data preprocessing, whereas the input in the AR model based policy is processed roughly only.

In Fig. 6, we show the cache hit rates of our proposed policy and four benchmark policies, including the AR model based policy [8], LRU, LFU [6], and Random Caching (RC). Assume that the caching policy for the proposed policy and the AR based policy are to cache the most popular contents preferentially according to the prediction. In spite of the different methods they implement, all the above policies are proposed to maximize cache hit rates. It can be observed that the cache hit rates of all the considered policies increase gradually as the storage capacity increases. It can also be observed that the proposed policy achieves better performance than the four benchmarks with higher cache hit rate. The LRU and LFU are liable to suffer performance degradation due to their neglect of content popularity. Due to the improved prediction accuracy as shown in Fig. 5, the proposed policy achieves higher cache hit rate than the AR based policy by up to 21.5%. Moreover, the proposed policy can predict the popularity of newly-added contents by leveraging user preference, which also leads to better caching performance.

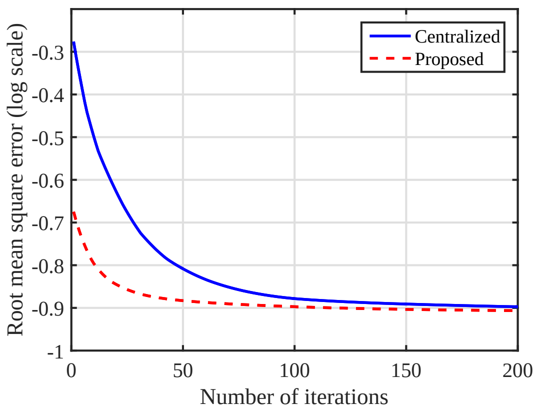

Fig. 7. Root mean-square error versus number of iterations.

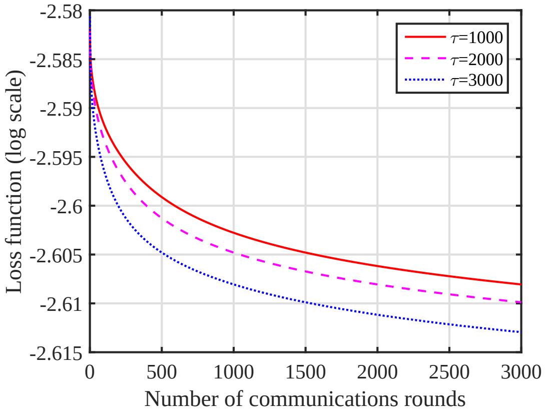

Fig. 8. Loss function versus number of communications rounds.

In Fig. 7, we show the logarithmic RMSE of our proposed policy and the centralized policy as the number of iterations varies when generating the global model. It can be observed that our proposed policy achieves faster convergence than the centralized policy. The reason is that our proposed policy generates the global model by integrating the existing local models based on FL. The global model can be initialized by the local models, instead of random generation. Consequently, a large part of repeated computation is avoided. While the centralized policy has to recalculate the training samples after data transmission, which is resource-wasting.

In Fig. 8, we show the value of the loss function (logarithmic RMSE) of our proposed policy when using different values of $\tau$. It can be observed that the value of the loss function decreases gradually as the number of communications rounds increasing. Obviously, the loss function converges more quickly when $\tau$ is bigger. For larger value of $\tau$, each F-AP using FL is able to achieve high predefined training accuracy with less communications rounds. Therefore, there exists a tradeoff between computation resources and communications resources. The value of $\tau$ can be adjusted flexibly according to the resource constraints.

## VI. Conclusions

In this paper, we have proposed a novel popularity prediction policy based on user preference learning, adaptive context space partitioning and FL. Our proposed policy can predict content popularity more accurately, even when the contents have no statistical data in advance. The reason is that the relation between content popularity and user preference is considered. Specifically, FL based model integration is efficient in reducing computational complexity and communication overhead, which makes our proposed policy more practical. The convergence bound of the FL based model integration approach is analyzed. Based on the theoretical analysis and the obtained expressions, the relationship between the convergence upper bound and the critical parameters is presented. Simulation results have shown that our proposed policy achieves better prediction and caching performance than traditional policies. Future work will explore the idea of combining user mobility prediction with content popularity prediction.

## Appendix A. Proof of Theorem 1

The idea behind the optimization problem of the form in (14) is the following. The intuitive averaging method is not practical in reality, unless the local functions $V_m(\boldsymbol{a})(m = 1, 2, ..., K)$ are all the same. An efficient improved method is to modify the local optimization problems before the aggregation steps. In order to modify the local problems, the local function $V_m(\boldsymbol{a})$ in $q_m$ in iteration $j$ can be perturbed by a quadratic term of the form: $-(\boldsymbol{b}_m^j)^T\boldsymbol{a}+\frac{\mu}{2}\|\boldsymbol{a}-\boldsymbol{a}^j\|^2$. Instead of solving the original optimization problem in (9), each F-AP should solve the perturbed problem. Therefore, the modified problem takes the following form:

$$\boldsymbol{a}_m^{j+1} = \underset{\boldsymbol{a}\in\mathbb{R}^S}{\arg\min}\, V_m(\boldsymbol{a}) - (\boldsymbol{b}_m^j)^T\boldsymbol{a} + \frac{\mu}{2}\|\boldsymbol{a}-\boldsymbol{a}^j\|^2. \tag{21}$$

The regularizer $\frac{\mu}{2}\|\boldsymbol{a}-\boldsymbol{a}^j\|^2$ can help avoid overfitting. For simplicity, we can choose $\mu = 0$. The vector $\boldsymbol{b}_m^j$ is not well defined yet. In order to obtain $\boldsymbol{b}_m^j$ and figure out how $\boldsymbol{b}_m^j$ would change along with iterations, the optimality conditions should be considered. Asymptotically as $j \to \infty$, it is expected that $\boldsymbol{a}_m^j \to \boldsymbol{a}^*$, in which $\boldsymbol{a}^*$ denotes the optimal solution. The value of $\boldsymbol{a}^*$ should be the solution of following equation:

$$\triangledown V_m(\boldsymbol{a}) - \boldsymbol{b}_m^j + \mu(\boldsymbol{a}-\boldsymbol{a}^j) = 0. \tag{22}$$

As a consequence, the vector $\boldsymbol{b}_m^j$ should be calculated as the following form:

$$\boldsymbol{b}_m^j = \triangledown V_m(\boldsymbol{a}^*) + \mu(\boldsymbol{a}^*-\boldsymbol{a}^j) \approx \triangledown V_m(\boldsymbol{a}^*), \tag{23}$$

since $\boldsymbol{a}^j \approx \boldsymbol{a}^*$. However, the vector $\boldsymbol{a}^*$ is unknown. Another solution is to put forward an update rule which makes $\boldsymbol{b}_m^j$ converge to $\triangledown V_m(\boldsymbol{a}^*)$ as $j \to \infty$.

DANE is presented to solve the problems like (13). The core idea of DANE is to form a local subproblem $V_m(\boldsymbol{a})$ which only depends on local data and the gradient of the entire function $\triangledown V^*(\boldsymbol{a})$. DANE operates via the quadratic perturbation trick (21) with

$$\boldsymbol{b}_m^j = \triangledown V_m(\boldsymbol{a}^j) - \eta \triangledown V^*(\boldsymbol{a}^j). \tag{24}$$

When $j \to \infty$, $\boldsymbol{a}^j \to \boldsymbol{a}^*$, so $\triangledown V^*(\boldsymbol{a}^j) \to \triangledown V^*(\boldsymbol{a}^*) = 0$. As a result, $\boldsymbol{b}_m^j$ converges to $\triangledown V_m(\boldsymbol{a}^*)$ as $j \to \infty$ In the default setting, we take $\mu = 0$ and $\eta = 1$.

This completes the proof.

## Appendix B. Proof of Lemma 1

At the beginning of each interval, $\boldsymbol{a}_{[l]}(t)$ and $\boldsymbol{a}_m(t)$ is synchronized with $\boldsymbol{a}(t)$. As a consequence, when $t = (l-1)\tau$, $\boldsymbol{a}_m(t) = \boldsymbol{a}_{[l]}(t)$. Since $g(0) = 0$, we can derive that $\|\boldsymbol{a}_m(t) - \boldsymbol{a}_{[l]}(t)\| = g(0)$ when $t = (l-1)\tau$.

Based on Algorithm 3, we know that the iterative formula of $\boldsymbol{a}_m(t)$ is:

$$\begin{aligned}\boldsymbol{a}_m(t) = \boldsymbol{a}_m(t-1) - \eta[&\triangledown v_i(\boldsymbol{a}_m(t-1)) \\ &- \triangledown v_i(\boldsymbol{a}((l-1)\tau)) + \triangledown V(\boldsymbol{a}((l-1)\tau))].\end{aligned} \tag{25}$$

Then we can derive *Lemma 1* by induction. Firstly, we assume that

$$\|\boldsymbol{a}_m(t-1) - \boldsymbol{a}_{[l]}(t-1)\| \le g(t-1-(l-1)\tau) \tag{26}$$

holds for some $t \in ((l-1)\tau, l\tau)$. Based on (16) and (25), we have:

$$\begin{aligned}&\|\boldsymbol{a}_m(t) - \boldsymbol{a}_{[l]}(t)\| \\ &= \|\boldsymbol{a}_m(t-1) - \eta[\triangledown v_i(\boldsymbol{a}_m(t-1)) - \triangledown v_i(\boldsymbol{a}((l-1)\tau)) \\ &\quad + \triangledown V(\boldsymbol{a}((l-1)\tau))] - [\boldsymbol{a}_{[l]}(t-1) - \eta \triangledown V(\boldsymbol{a}_{[l]}(t-1))]\| \\ &= \|\boldsymbol{a}_m(t-1) - \boldsymbol{a}_{[l]}(t-1) - \eta(\triangledown v_i(\boldsymbol{a}_m(t-1)) \\ &\quad - \triangledown v_i(\boldsymbol{a}_{[l]}(t-1)) + \triangledown v_i(\boldsymbol{a}_{[l]}(t-1)) - \triangledown V(\boldsymbol{a}_{[l]}(t-1))) \\ &\quad - \triangledown v_i(\boldsymbol{a}((l-1)\tau)) + \triangledown V(\boldsymbol{a}((l-1)\tau))\| \\ &\le \|\boldsymbol{a}_m(t-1) - \boldsymbol{a}_{[l]}(t-1)\| + \eta\| \triangledown v_i(\boldsymbol{a}_m(t-1)) \\ &\quad - \triangledown v_i(\boldsymbol{a}_{[l]}(t-1))\| + \eta\| \triangledown v_i(\boldsymbol{a}_{[l]}(t-1)) - \triangledown V(\boldsymbol{a}_{[l]}(t-1))\| \\ &\quad\quad + \eta\| \triangledown v_i(\boldsymbol{a}((l-1)\tau)) - \triangledown V(\boldsymbol{a}((l-1)\tau))\| \\ &\le (\eta\beta+1)\|\boldsymbol{a}_m(t-1) - \boldsymbol{a}_{[l]}(t-1)\| \\ &\quad + 2\eta\delta \le (\eta\beta+1)g(t-1-(l-1)\tau) + 2\eta\delta \\ &= (\eta\beta+1)\frac{2\delta}{\beta}((\eta\beta+1)^{t-1-(l-1)\tau} - 1) + 2\eta\delta \\ &= \frac{2\delta}{\beta}((\eta\beta+1)^{t-(l-1)\tau} - 1) \\ &= g(t-(l-1)\tau).\end{aligned} \tag{27}$$

This completes the proof.

## Appendix C. Proof of Theorem 2

We can prove Theorem 2 based on Lemma 1. According to the aggregation step in Algorithm 3, we have:

$$\begin{aligned}\boldsymbol{a}(t) &= \sum_{m=1}^{K} \frac{L_m}{L}\boldsymbol{a}_m(t) \\ &= \sum_{m=1}^{K} \frac{L_m}{L}(\boldsymbol{a}_m(t-1) - \eta[\triangledown v_i(\boldsymbol{a}_m(t-1)) \\ &\quad - \triangledown v_i(\boldsymbol{a}((l-1)\tau)) + \triangledown V(\boldsymbol{a}((l-1)\tau))]) \\ &= \boldsymbol{a}(t-1) - \sum_{m=1}^{K} \frac{L_m}{L}\eta[\triangledown v_i(\boldsymbol{a}_m(t-1))\end{aligned}$$

$$-\bigtriangledown v_i(\boldsymbol{a}((l-1)\tau))+\bigtriangledown V(\boldsymbol{a}((l-1)\tau))]. \tag{28}$$

For $t\in((l-1)\tau,l\tau)$, we have:

$$\begin{aligned}
&\|\boldsymbol{a}(t)-\boldsymbol{a}_{[l]}(t)\|\\
&=\|\boldsymbol{a}(t-1)-\eta\sum_{m=1}^{K}\frac{L_m}{L}[\bigtriangledown v_i(\boldsymbol{a}_m(t-1))-\bigtriangledown v_i(\boldsymbol{a}((l-1)\tau))\\
&\quad+\bigtriangledown V(\boldsymbol{a}((l-1)\tau))]-[\boldsymbol{a}_{[l]}(t-1)-\eta\bigtriangledown V(\boldsymbol{a}_{[l]}(t-1))]\|\\
&=\|\boldsymbol{a}(t-1)-\boldsymbol{a}_{[l]}(t-1)-\frac{\eta}{L}\sum_{m=1}^{K}L_m[\bigtriangledown v_i(\boldsymbol{a}_m(t-1)\\
&\quad-\frac{1}{L_m}\sum_{(p_j,\boldsymbol{x}_j)\in\mathcal{B}_m}\bigtriangledown v_j(\boldsymbol{a}_{[l]}(t-1)))]\\
&\quad-\frac{\eta}{L}\sum_{m=1}^{K}L_m(\bigtriangledown v_i(\boldsymbol{a}((l-1)\tau))-\bigtriangledown V(\boldsymbol{a}((l-1)\tau)))\|\\
&\le\|\boldsymbol{a}(t-1)-\boldsymbol{a}_{[l]}(t-1)\|+\frac{\eta}{L}\sum_{m=1}^{K}\sum_{(p_j,\boldsymbol{x}_j)\in\mathcal{B}_m}\|\bigtriangledown v_i(\boldsymbol{a}_m(t-1))\\
&\quad-\bigtriangledown v_j(\boldsymbol{a}_{[l]}(t-1))\|+\eta\delta\\
&\le\|\boldsymbol{a}(t-1)-\boldsymbol{a}_{[l]}(t-1)\|+\frac{\eta}{L}\sum_{m=1}^{K}\sum_{(p_j,\boldsymbol{x}_j)\in\mathcal{B}_m}\|\bigtriangledown v_i(\boldsymbol{a}_m(t-1))\\
&\quad-\bigtriangledown v_j((\boldsymbol{a}_m(t-1))+\bigtriangledown v_j((\boldsymbol{a}_m(t-1))-\bigtriangledown v_j(\boldsymbol{a}_{[l]}(t-1))\|\\
&\quad+\eta\delta\\
&\le\|\boldsymbol{a}(t-1)-\boldsymbol{a}_{[l]}(t-1)\|+\frac{\eta\beta}{L}\sum_{m=1}^{K}L_m\|\boldsymbol{a}_m(t-1)-\boldsymbol{a}_{[l]}(t-1)\|\\
&\quad+2\eta\delta\\
&\le\|\boldsymbol{a}(t-1)-\boldsymbol{a}_{[l]}(t-1)\|+\frac{\eta\beta}{L}\sum_{m=1}^{K}L_m g(t-1-(l-1)\tau)+2\eta\delta\\
&=\|\boldsymbol{a}(t-1)-\boldsymbol{a}_{[l]}(t-1)\|+2\eta\delta(\eta\beta+1)^{t-1-(l-1)\tau}.
\end{aligned}\tag{29}$$

As a consequence,

$$\begin{aligned}
&\|\boldsymbol{a}(t)-\boldsymbol{a}_{[l]}(t)||-||\boldsymbol{a}(t-1)-\boldsymbol{a}_{[l]}(t-1)\|\\
&\qquad\le 2\eta\delta(\eta\beta+1)^{t-1-(l-1)\tau}.
\end{aligned}\tag{30}$$

When $t=(l-1)\tau$, $\boldsymbol{a}(t)=\boldsymbol{a}_{[l]}(t)$, so $\boldsymbol{a}(t)-\boldsymbol{a}_{[l]}(t)=0$. By summing up (30) over different values of $t\in((l-1)\tau,l\tau)$, we have:

$$\begin{aligned}
\|\boldsymbol{a}(t)-\boldsymbol{a}_{[l]}(t)\|=&\sum_{x=(l-1)\tau+1}^{t}\|\boldsymbol{a}(x)\\
&-\boldsymbol{a}_{[l]}(x)\|-\|\boldsymbol{a}(x-1)-\boldsymbol{a}_{[l]}(x-1)\|\\
\le&\,2\eta\delta\sum_{x=(l-1)\tau+1}^{t}(\eta\beta+1)^{x-1-(l-1)\tau}\\
=&\,\frac{2\delta}{\beta}((\eta\beta+1)^{t-(l-1)\tau}-1)\\
=&\,g(t-(l-1)\tau).
\end{aligned}\tag{31}$$

This completes the proof.

## Appendix D. Proof of Theorem 3

Intuitively, we have $V(\boldsymbol{a}_{[l]}(t))-V(\boldsymbol{a}^*)>0$. Then, we introduce the following lemmas.

*Lemma 2:* When $\eta\beta<1$, for any $l$ and $t\in[l]$, we have that $\|\boldsymbol{a}_{[l]}(t)-\boldsymbol{a}^*\|$ does not increase with $t$.

*Proof:* According to Lemma 3.5 in [34], we have:

$$\begin{aligned}
&\bigtriangledown V(\boldsymbol{a}_{[l]}(t))^T(\boldsymbol{a}_{[l]}(t)-\boldsymbol{a}^*)-\frac{\|\bigtriangledown V(\boldsymbol{a}_{[l]}(t))\|^2}{2\beta}\\
&\qquad\ge V(\boldsymbol{a}_{[l]}(t))-V(\boldsymbol{a}^*)>0,
\end{aligned}\tag{32}$$

i.e., $\bigtriangledown V(\boldsymbol{a}_{[l]}(t))^T(\boldsymbol{a}_{[l]}(t)-\boldsymbol{a}^*)>\frac{\|\bigtriangledown V(\boldsymbol{a}_{[l]}(t))\|^2}{2\beta}$. Therefore, we have:

$$\begin{aligned}
&\|\boldsymbol{a}_{[l]}(t+1)-\boldsymbol{a}^*\|^2=\|\boldsymbol{a}_{[l]}(t)-\eta\bigtriangledown V(\boldsymbol{a}_{[l]}(t))-\boldsymbol{a}^*\|^2\\
&=\|\boldsymbol{a}_{[l]}(t)-\boldsymbol{a}^*\|^2-2\eta\bigtriangledown V(\boldsymbol{a}_{[l]}(t))^T(\boldsymbol{a}_{[l]}(t)-\boldsymbol{a}^*)\\
&\quad+\eta^2\|\bigtriangledown V(\boldsymbol{a}_{[l]}(t))\|^2\\
&<\|\boldsymbol{a}_{[l]}(t)-\boldsymbol{a}^*\|^2-\eta\frac{\|\bigtriangledown V(\boldsymbol{a}_{[l]}(t))\|^2}{\beta}+\eta^2\|\bigtriangledown V(\boldsymbol{a}_{[l]}(t))\|^2\\
&=\|\boldsymbol{a}_{[l]}(t)-\boldsymbol{a}^*\|^2+\eta(\frac{\eta\beta-1}{\beta})\|V(\boldsymbol{a}_{[l]}(t))\|^2.
\end{aligned}\tag{33}$$

When $\eta\beta<1$, we have:

$$\|\boldsymbol{a}_{[l]}(t+1)-\boldsymbol{a}^*\|^2\le\|\boldsymbol{a}_{[l]}(t)-\boldsymbol{a}^*\|^2. \tag{34}$$

■

*Lemma 3:* For any $l$, when $\eta\beta<1$ and $t\in[l]$, we have:

$$V(\boldsymbol{a}_{[l]}(t+1))-V(\boldsymbol{a}_{[l]}(t))\le\eta(\frac{\beta\eta}{2}-1)\|\bigtriangledown V(\boldsymbol{a}_{[l]}(t))\|^2. \tag{35}$$

*Proof:* According to Lemma 3.4 in [34], we have:

$$V(\boldsymbol{x}_1)-V(\boldsymbol{x}_2)\le\bigtriangledown V(\boldsymbol{x}_2)^T(\boldsymbol{x}_1-\boldsymbol{x}_2)+\frac{\beta}{2}||\boldsymbol{x}_1-\boldsymbol{x}_2||^2. \tag{36}$$

Then, we can derive that

$$\begin{aligned}
&V(\boldsymbol{a}_{[l]}(t+1))-V(\boldsymbol{a}_{[l]}(t))\le\bigtriangledown V(\boldsymbol{a}_{[l]}(t))^T(\boldsymbol{a}_{[l]}(t+1)-\boldsymbol{a}_{[l]}(t))\\
&\quad+\frac{\beta}{2}\|\boldsymbol{a}_{[l]}(t+1)-\boldsymbol{a}_{[l]}(t)\|^2\\
&=\bigtriangledown V(\boldsymbol{a}_{[l]}(t+1))^T(-\eta\bigtriangledown V(\boldsymbol{a}_{[l]}(t)))+\frac{\beta}{2}\|-\eta\bigtriangledown V(\boldsymbol{a}_{[l]}(t))\|^2\\
&=\eta(\frac{\beta\eta}{2}-1)\|\bigtriangledown V(\boldsymbol{a}_{[l]}(t))\|^2.
\end{aligned}\tag{37}$$

■

*Lemma 4:* For any $l$, when $\eta\beta<1$ and $t\in[l]$, we have:

$$\begin{aligned}
&\frac{1}{V(\boldsymbol{a}_{[l]}(t+1))-V(\boldsymbol{a}^*)}-\frac{1}{V(\boldsymbol{a}_{[l]}(t))-V(\boldsymbol{a}^*)}\\
&\qquad\ge\sigma\eta(1-\frac{\beta\eta}{2}).
\end{aligned}\tag{38}$$

*Proof:* According to (35), we have:

$$\begin{aligned}
&V(\boldsymbol{a}_{[l]}(t+1))-V(\boldsymbol{a}^*)\le V(\boldsymbol{a}_{[l]}(t))\\
&\qquad-V(\boldsymbol{a}^*)+\eta(\frac{\beta\eta}{2}-1)\|\bigtriangledown V(\boldsymbol{a}_{[l]}(t))\|^2.
\end{aligned}\tag{39}$$

Since function $V(\boldsymbol{a})$ is convex, we have:

$$\begin{aligned}
V(\boldsymbol{a}_{[l]}(t))-V(\boldsymbol{a}^*)&\le V(\boldsymbol{a}_{[l]}(t))^T(\boldsymbol{a}_{[l]}(t)-\boldsymbol{a}^*)\\
&\le\|V(\boldsymbol{a}_{[l]}(t))\|\|\boldsymbol{a}_{[l]}(t)-\boldsymbol{a}^*\|.
\end{aligned}\tag{40}$$

Therefore, we can derive that

$$\frac{V(\boldsymbol{a}_{[l]}(t))-V(\boldsymbol{a}^*)}{\|\boldsymbol{a}_{[l]}(t)-\boldsymbol{a}^*\|}\le\|V(\boldsymbol{a}_{[l]}(t))\|. \tag{41}$$

Substituting (41) into (39), we have

$$\begin{aligned}
&V(\boldsymbol{a}_{[l]}(t+1)) - V(\boldsymbol{a}^*) \le V(\boldsymbol{a}_{[l]}(t)) - V(\boldsymbol{a}^*) \\
&\quad - \eta(1-\frac{\beta\eta}{2})\frac{(V(\boldsymbol{a}_{[l]}(t)) - V(\boldsymbol{a}^*))^2}{\|\boldsymbol{a}_{[l]}(t) - \boldsymbol{a}^*\|^2} \\
&\le V(\boldsymbol{a}_{[l]}(t)) - V(\boldsymbol{a}^*) - \sigma\eta(1-\frac{\beta\eta}{2})(V(\boldsymbol{a}_{[l]}(t)) - V(\boldsymbol{a}^*))^2.
\end{aligned} \tag{42}$$

Since $(V(\boldsymbol{a}_{[l]}(t+1)) - V(\boldsymbol{a}^*))(V(\boldsymbol{a}_{[l]}(t)) - V(\boldsymbol{a}^*)) > 0$ (proved in Lemma 2), we can use it to divide both sides of (42) and have

$$\begin{aligned}
\frac{1}{V(\boldsymbol{a}_{[l]}(t)) - V(\boldsymbol{a}^*)} &\le \frac{1}{V(\boldsymbol{a}_{[l]}(t+1)) - V(\boldsymbol{a}^*)} \\
&\quad - \sigma\eta(1-\frac{\beta\eta}{2})\frac{V(\boldsymbol{a}_{[l]}(t)) - V(\boldsymbol{a}^*)}{V(\boldsymbol{a}_{[l]}(t+1)) - V(\boldsymbol{a}^*)}.
\end{aligned} \tag{43}$$

From (39), we know that $V(\boldsymbol{a}_{[l]}(t)) - V(\boldsymbol{a}^*) \ge V(\boldsymbol{a}_{[l]}(t+1)) - V(\boldsymbol{a}^*)$. Thus, $\frac{V(\boldsymbol{a}_{[l]}(t)) - V(\boldsymbol{a}^*)}{V(\boldsymbol{a}_{[l]}(t+1)) - V(\boldsymbol{a}^*)} \ge 1$. Substituting it into (43), we have:

$$\begin{aligned}
\frac{1}{V(\boldsymbol{a}_{[l]}(t+1)) - V(\boldsymbol{a}^*)} - \frac{1}{V(\boldsymbol{a}_{[l]}(t)) - V(\boldsymbol{a}^*)} \\
\ge \sigma\eta(1-\frac{\beta\eta}{2}).
\end{aligned} \tag{44}$$

■

Now we can prove Theorem 3.

Based on Lemma 4, we can infer that

$$\begin{aligned}
&\frac{1}{V(\boldsymbol{a}_{[l]}(l\tau)) - V(\boldsymbol{a}^*)} - \frac{1}{V(\boldsymbol{a}_{[l]}((l-1)\tau)) - V(\boldsymbol{a}^*)} \\
&= \sum_{t=(l-1)\tau}^{l\tau-1}(\frac{1}{V(\boldsymbol{a}_{[l]}(t+1)) - V(\boldsymbol{a}^*)} - \frac{1}{V(\boldsymbol{a}_{[l]}(t)) - V(\boldsymbol{a}^*)}) \\
&\ge \tau\sigma\eta(1-\frac{\beta\eta}{2}).
\end{aligned} \tag{45}$$

Summing up (45) for all $l = 1, 2, ..., Z$ yields

$$\begin{aligned}
\sum_{l=1}^{Z}[\frac{1}{V(\boldsymbol{a}_{[l]}(l\tau)) - V(\boldsymbol{a}^*)} - \frac{1}{V(\boldsymbol{a}_{[l]}((l-1)\tau)) - V(\boldsymbol{a}^*)}] \\
\ge \sum_{l=1}^{Z}\tau\sigma\eta(1-\frac{\beta\eta}{2}) = Z\tau\sigma\eta(1-\frac{\beta\eta}{2}).
\end{aligned} \tag{46}$$

Since $T = Z\tau$, we can rewrite (46) as follows:

$$\begin{aligned}
&\frac{1}{V(\boldsymbol{a}_{[Z]}(T)) - V(\boldsymbol{a}^*)} - \frac{1}{V(\boldsymbol{a}_{[1]}(0)) - V(\boldsymbol{a}^*)} \\
&\quad - \sum_{l=1}^{Z-1}(\frac{1}{V(\boldsymbol{a}_{[l+1]}(l\tau)) - V(\boldsymbol{a}^*)} - \frac{1}{V(\boldsymbol{a}_{[l]}(l\tau)) - V(\boldsymbol{a}^*)}) \\
&\ge T\sigma\eta(1-\frac{\beta\eta}{2}).
\end{aligned} \tag{47}$$

We also have:

$$\begin{aligned}
&\frac{1}{V(\boldsymbol{a}_{[l+1]}(l\tau)) - V(\boldsymbol{a}^*)} - \frac{1}{V(\boldsymbol{a}_{[l]}(l\tau)) - V(\boldsymbol{a}^*)} \\
&= \frac{V(\boldsymbol{a}_{[l]}(l\tau)) - V(\boldsymbol{a}^*) - V(\boldsymbol{a}_{[l+1]}(l\tau)) + V(\boldsymbol{a}^*)}{(V(\boldsymbol{a}_{[l+1]}(l\tau)) - V(\boldsymbol{a}^*))(V(\boldsymbol{a}_{[l]}(l\tau)) - V(\boldsymbol{a}^*))} \\
&= -\frac{V(\boldsymbol{a}_{[l+1]}(l\tau)) - V(\boldsymbol{a}_{[l]}(l\tau))}{(V(\boldsymbol{a}_{[l+1]}(l\tau)) - V(\boldsymbol{a}^*))(V(\boldsymbol{a}_{[l]}(l\tau)) - V(\boldsymbol{a}^*))} \\
&\ge -\frac{\rho\|\boldsymbol{a}_{[l+1]}(l\tau) - \boldsymbol{a}_{[l]}(l\tau)\|}{(V(\boldsymbol{a}_{[l+1]}(l\tau)) - V(\boldsymbol{a}^*))(V(\boldsymbol{a}_{[l]}(l\tau)) - V(\boldsymbol{a}^*))} \\
&= -\frac{\rho\|\boldsymbol{a}(l\tau) - \boldsymbol{a}_{[l]}(l\tau)\|}{(V(\boldsymbol{a}_{[l+1]}(l\tau)) - V(\boldsymbol{a}^*))(V(\boldsymbol{a}_{[l]}(l\tau)) - V(\boldsymbol{a}^*))} \\
&\ge -\frac{\rho g(\tau)}{(V(\boldsymbol{a}_{[l+1]}(l\tau)) - V(\boldsymbol{a}^*))(V(\boldsymbol{a}_{[l]}(l\tau)) - V(\boldsymbol{a}^*))}.
\end{aligned} \tag{48}$$

Since $V(\boldsymbol{a}^f) - V(\boldsymbol{a}^*) \ge \epsilon$, we have $V(\boldsymbol{a}_{[l]}(l\tau)) - V(\boldsymbol{a}^*) \ge \epsilon$ for all $l$. We have proved that $V(\boldsymbol{a}_{[l]}(t)) \ge V(\boldsymbol{a}_{[l]}(t+1))$. Therefore,

$$\begin{aligned}
\frac{-1}{(V(\boldsymbol{a}_{[l+1]}(l\tau)) - V(\boldsymbol{a}^*))(V(\boldsymbol{a}_{[l]}(l\tau)) - V(\boldsymbol{a}^*))} \\
\ge -\frac{1}{\epsilon^2}.
\end{aligned} \tag{49}$$

Substituting (49) into (48), we have:

$$\begin{aligned}
\frac{1}{V(\boldsymbol{a}_{[l+1]}(l\tau)) - V(\boldsymbol{a}^*)} - \frac{1}{V(\boldsymbol{a}_{[l]}(l\tau)) - V(\boldsymbol{a}^*)} \\
\ge -\frac{\rho g(\tau)}{\epsilon^2}.
\end{aligned} \tag{50}$$

Substituting (50) into (47), we have:

$$\begin{aligned}
&\frac{1}{V(\boldsymbol{a}_{[Z]}(T)) - V(\boldsymbol{a}^*)} - \frac{1}{V(\boldsymbol{a}_{[1]}(0)) - V(\boldsymbol{a}^*)} \\
&\ge T\sigma\eta(1-\frac{\beta\eta}{2}) + \sum_{l=1}^{Z-1}(-\frac{\rho g(\tau)}{\epsilon^2}) \\
&= T\sigma\eta(1-\frac{\beta\eta}{2}) - (Z-1)\frac{\rho g(\tau)}{\epsilon^2}.
\end{aligned} \tag{51}$$

According to (49), we also have:

$$\frac{-1}{(V(\boldsymbol{a}(T)) - V(\boldsymbol{a}^*))(V(\boldsymbol{a}_{[Z]}(T)) - V(\boldsymbol{a}^*))} \ge -\frac{1}{\epsilon^2}. \tag{52}$$

Then, we have:

$$\begin{aligned}
&\frac{1}{V(\boldsymbol{a}(T)) - V(\boldsymbol{a}^*)} - \frac{1}{V(\boldsymbol{a}_{[Z]}(T)) - V(\boldsymbol{a}^*)} \\
&= \frac{V(\boldsymbol{a}_{[Z]}(T)) - V(\boldsymbol{a}^*) - V(\boldsymbol{a}(T)) + V(\boldsymbol{a}^*)}{(V(\boldsymbol{a}(T)) - V(\boldsymbol{a}^*))(V(\boldsymbol{a}_{[Z]}(T)) - V(\boldsymbol{a}^*))} \\
&= \frac{V(\boldsymbol{a}_{[Z]}(T)) - V(\boldsymbol{a}(T))}{(V(\boldsymbol{a}(T)) - V(\boldsymbol{a}^*))(V(\boldsymbol{a}_{[Z]}(T)) - V(\boldsymbol{a}^*))} \\
&\ge -\frac{\rho g(\tau)}{(V(\boldsymbol{a}(T)) - V(\boldsymbol{a}^*))(V(\boldsymbol{a}_{[Z]}(T)) - V(\boldsymbol{a}^*))} \\
&\ge -\frac{\rho g(\tau)}{\epsilon^2}.
\end{aligned} \tag{53}$$

Summing up (51) and (53), we have:

$$\begin{aligned}
&\frac{1}{V(\boldsymbol{a}(T)) - V(\boldsymbol{a}^*)} - \frac{1}{V(\boldsymbol{a}_{[1]}(0)) - V(\boldsymbol{a}^*)} \\
&\ge T\sigma\eta(1-\frac{\beta\eta}{2}) - Z\frac{\rho g(\tau)}{\epsilon^2} \\
&= T\sigma\eta(1-\frac{\beta\eta}{2}) - \frac{T}{\tau}\frac{\rho g(\tau)}{\epsilon^2} \\
&= T[\sigma\eta(1-\frac{\beta\eta}{2}) - \frac{\rho g(\tau)}{\tau\epsilon^2}].
\end{aligned} \tag{54}$$

Thus, we have:

$$\begin{aligned}\frac{1}{V(\boldsymbol{a}(T))-V(\boldsymbol{a}^*)} &\geq \frac{1}{V(\boldsymbol{a}(T))-V(\boldsymbol{a}^*)}\\ &- \frac{1}{V(\boldsymbol{a}_{[1]}(0))-V(\boldsymbol{a}^*)}\\ &\geq T[\sigma\eta(1-\frac{\beta\eta}{2})-\frac{\rho g(\tau)}{\tau\epsilon^2}]>0.\end{aligned} \tag{55}$$

When the condition that $(1-\frac{\beta\eta}{2})\sigma\eta > \frac{\rho g(\tau)}{\tau\epsilon}$ is satisfied, we can derive that

$$V(\boldsymbol{a}(T))-V(\boldsymbol{a}^*) \leq \frac{1}{T(\sigma\eta(1-\frac{\beta\eta}{2})-\frac{\rho g(\tau)}{\tau\epsilon^2})}, \tag{56}$$

i.e.,

$$V(\boldsymbol{a}^f)-V(\boldsymbol{a}^*) \leq \frac{1}{T(\eta\sigma(1-\frac{\beta\eta}{2})-\frac{\rho g(\tau)}{\tau\epsilon^2})}.$$

This completes the proof.

## References

[1] Y. Jiang, W. Huang, M. Bennis, and F. Zheng, "Decentralized asynchronous coded caching design and performance analysis in fog radio access networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 3, pp. 540–551, Mar. 2020.

[2] Y. Jiang, C. Wan, M. Tao, F. Zheng, P. Zhu, X. Gao, and X. You, "Analysis and optimization of fog radio access networks with hybrid caching: Delay and energy efficiency," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 69–82, Jan. 2021.

[3] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. M. Leung, "Distributed resource allocation and computation offloading in fog and cloud networks with non-orthogonal multiple access," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 12 137–12 151, Dec. 2018.

[4] Y. Jiang, A. Peng, C. Wan, Y. Cui, X. You, F. Zheng, and S. Jin, "Analysis and optimization of cache-enabled fog radio access networks: Successful transmission probability, fractional offloaded traffic and delay," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5219–5231, May 2020.

[5] Y. Jiang, Y. Hu, M. Bennis, F. Zheng, and X. You, "A mean field game-based distributed edge caching in fog radio access networks," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1567–1580, Mar. 2020.

[6] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5G systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, Feb. 2014.

[7] J. Lee and J. Lee, "Music popularity: Metrics, characteristics, and audio-based prediction," *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 3173–3182, Nov. 2018.

[8] Y. Liu, T. Zhi, H. Xi, X. Duan, and H. Zhang, "A novel content popularity prediction algorithm based on auto regressive model in information-centric IoT," *IEEE Access*, vol. 7, pp. 27 555–27 564, 2019.

[9] N. B. Hassine, R. Milocco, and P. Minet, "ARMA based popularity prediction for caching in content delivery networks," in *2017 Wireless Days*, Mar. 2017, pp. 113–120.

[10] W. Liu, J. Zhang, Z. Liang, L. Peng, and J. Cai, "Content popularity prediction and caching for ICN: A deep learning approach with SDN," *IEEE Access*, vol. 6, pp. 5075–5089, 2018.

[11] H. Feng, Y. Jiang, D. Niyato, F. Zheng, and X. You, "Content popularity prediction via deep learning in cache-enabled fog radio access networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2019, pp. 1–6.

[12] Y. Jiang, H. Feng, F. C. Zheng, D. Niyato, and X. You, "Deep learning-based edge caching in fog radio access networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 12, pp. 8442–8454, Dec. 2020.

[13] S. Mehrizi, A. Tsakmalis, S. Chatzinotas, and B. Ottersten, "A Bayesian Poisson Gaussian Process model for popularity learning in edge-caching networks," *IEEE Access*, vol. 7, pp. 92 341–92 354, 2019.

[14] Y. Jiang, M. Ma, M. Bennis, F. Zheng, and X. You, "User preference learning-based edge caching for fog radio access network," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1268–1283, Feb. 2019.

[15] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y. C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[16] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, Apr. 2019, pp. 1387–1395.

[17] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[18] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.

[19] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Differentially private asynchronous federated learning for mobile edge computing in urban informatics," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2134–2143, Mar. 2020.

[20] W. Sun, N. Xu, L. Wang, H. Zhang, and Y. Zhang, "Dynamic digital twin and federated learning with incentives for air-ground networks," *IEEE Transactions on Network Science and Engineering*, pp. 1–13, 2020.

[21] W. Sun, S. Lei, L. Wang, Z. Liu, and Y. Zhang, "Adaptive federated learning and digital twin for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5605–5614, Aug. 2021.

[22] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 700–10 714, Dec. 2019.

[23] A. Zohourian, H. Sajedi, and A. Yavary, "Popularity prediction of images and videos on instagram," in *2018 4th International Conference on Web Research (ICWR)*, Apr. 2018, pp. 111–117.

[24] Q. Chen, W. Wang, and Z. Zhang, "Clustered popularity prediction for content caching," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, May 2019, pp. 1–6.

[25] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *CoRR*, vol. abs/1812.02858, 2018. [Online]. Available: http://arxiv.org/abs/1812.02858

[26] Y. Wu, Y. Jiang, M. Bennis, F. Zheng, X. Gao, and X. You, "Content popularity prediction in fog radio access networks: A federated learning based approach," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, June 2020, pp. 1–6.

[27] Y. Jiang, M. Ma, M. Bennis, F. Zheng, and X. You, "A novel caching policy with content popularity prediction and user preference learning in Fog-RAN," in *2017 IEEE Globecom Workshops (GC Wkshps)*, Dec. 2017, pp. 1–6.

[28] 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.

[29] S. Li, X. Jie, M. V. D. Schaar, and W. Li, "Trend-aware video caching through online learning," *IEEE Transactions on Multimedia*, vol. 18, no. 12, pp. 2503–2516, Dec. 2016.

[30] W. Jing, X. Wen, Z. Lu, and H. Zhang, "User-centric delay-aware joint caching and user association optimization in cache-enabled wireless networks," *IEEE Access*, vol. 7, pp. 74 961–74 972, 2019.

[31] J. Konecný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *CoRR*, vol. abs/1610.02527, 2016. [Online]. Available: http://arxiv.org/abs/1610.02527

[32] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," *CoRR*, vol. abs/1804.05271, 2018. [Online]. Available: http://arxiv.org/abs/1804.05271

[33] F. M. Harper and J. A. Konstan, *The MovieLens Datasets: History and Context*, 2015.

[34] S. Bubeck, "Convex optimization: Algorithms and complexity," *Foundations and trends in Machine Learning*, vol. 8, no. 3-4, pp. 231–357, 2015.

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/TWC.2021.3124586, IEEE Transactions on Wireless Communications

**Yanxiang Jiang (S'03-M'07-SM'18)** received the B.S. degree in electrical engineering from Nanjing University, Nanjing, China, in 1999 and the M.S. and Ph.D. degrees in communications and information systems from Southeast University, Nanjing, China, in 2003 and 2007, respectively.

Dr. Jiang was a Visiting Scholar with the Signals and Information Group, Department of Electrical and Computer Engineering, University of Maryland at College Park, College Park, MD, USA, in 2014. He is currently an Associate Professor with the National Mobile Communications Research Laboratory, Southeast University, Nanjing, China. His research interests are in the area of broadband wireless mobile communications, covering topics such as machine learning for wireless communications, edge caching, radio resource allocation and management, fog radio access networks, small cells and heterogeneous networks, cooperative communications, green communications, device to device communications, and massive MIMO.

**Yuting Wu** received the M.S. degree in communications and information systems from Southeast University, Nanjing, China.

Her research interests include radio resource management and edge caching.

**Fu-Chun Zheng** obtained the BEng (1985) and MEng (1988) degrees in radio engineering from Harbin Institute of Technology, China, and the PhD degree in Electrical Engineering from the University of Edinburgh, UK, in 1992.

From 1992 to 1995, he was a post-doctoral research associate with the University of Bradford, UK, Between May 1995 and August 2007, he was with Victoria University, Melbourne, Australia, first as a lecturer and then as an associate professor in mobile communications. He was with the University of Reading, UK, from September 2007 to July 2016 as a Professor (Chair) of Signal Processing. He has also been a distinguished adjunct professor with Southeast University, China, since 2010. Since August 2016, he has been with Harbin Institute of Technology (Shenzhen), China, as a distinguished professor, and the University of York, UK. He has been awarded two UK EPSRC Visiting Fellowships - both hosted by the University of York (UK): first in August 2002 and then again in August 2006. Over the past two decades, Dr Zheng has also carried out many government and industry sponsored research projects - in Australia, the UK, and China. He has been both a short term visiting fellow and a long term visiting research fellow with British Telecom, UK. Dr Zheng's current research interests include multiple antenna systems, green communications, ultra-dense networks, and ultra-reliable low latency communications (URLLC).

He has been an active IEEE member since 1995. He was an editor (2001-2004) of IEEE Transactions on Wireless Communications. In 2006, Dr Zheng served as the general chair of IEEE VTC 2006-S, Melbourne, Australia (www.ieeevtc.org/vtc2006spring) - the first ever VTC held in the southern hemisphere in VTCs history of six decades. More recently he was the executive TPC Chair for VTC 2016-S, Nanjing, China (the first ever VTC held in mainland China: www.ieeevtc.org/vtc2016spring).

**Mehdi Bennis (S'07-AM'08-SM'15)** received his M.Sc. degree in electrical engineering jointly from EPFL, Switzerland, and the Eurecom Institute, France, in 2002. He obtained his Ph.D. from the University of Oulu in December 2009 on spectrum sharing for future mobile cellular systems. Currently he is a professor at the University of Oulu and an Academy of Finland research fellow. His main research interests are in radio resource management, heterogeneous networks, game theory, and machine learning in 5G networks and beyond. He has co-authored one book and published more than 200 research papers in international conferences, journals, and book chapters. He was the recipient of the prestigious 2015 Fred W. Ellersick Prize from the IEEE Communications Society, the 2016 Best Tutorial Prize from the IEEE Communications Society, the 2017 EURASIP Best Paper Award for the Journal of Wireless Communications and Networks, and the 2017 all-University of Oulu Award for Research.

**Xiaohu You (SM'11-F'12)** received the B.S., M.S., and Ph.D. degrees in electrical engineering from Southeast University, Nanjing, China, in 1982, 1985, and 1988, respectively. Since 1990, he has been working with National Mobile Communications Research Laboratory at Southeast University, where he holds the ranks of professor and director. He is the Chief of the Technical Group of China 3G/B3G Mobile Communication R&D Project. His research interests include mobile communications, adaptive signal processing, and artificial neural networks, with applications to communications and biomedical engineering.

Dr. You was a recipient of the Excellent Paper Prize from the China Institute of Communications in 1987; the Elite Outstanding Young Teacher award from Southeast University in 1990, 1991, and 1993; and the National Technological Invention Award of China in 2011. He was also a recipient of the 1989 Young Teacher Award of Fok Ying Tung Education Foundation, State Education Commission of China.