







Intelligent Content Caching and User Association in Mobile Edge Computing Networks for Smart Cities

Hui Li , *Student Member, IEEE*, Xiuhua Li , *Member, IEEE*, Chuan Sun , *Student Member, IEEE*, Fang Fang , *Senior Member, IEEE*, Qilin Fan , *Member, IEEE*, Xiaofei Wang , *Senior Member, IEEE*, and Victor C. M. Leung , *Life Fellow, IEEE*

Abstract—To support rapidly increasing multimedia services of smart cities, mobile edge computing (MEC) networks can significantly reduce content acquisition latency. However, due to user mobility and the possibility of re-association, it is challenging to obtain a proper content caching and user association policy. In this article, we investigate the issue of joint content caching and user association with high user mobility in MEC networks by minimizing content acquisition latency and handover latency. To address the problem, we optimize the original mixed time-scale problem in two stages: long-time scale content caching and short-time scale user association. We propose an intelligent content caching framework based on a weighted latent factor model to determine content caching policy at each time frame. Then we design a matching theory-based lazy re-association strategy at each time slot. Simulation results

on real-world MEC networks demonstrate the effectiveness of the proposed framework.

Index Terms—Mobile edge computing networks, content caching, user association, weighted latent factor model.

I. INTRODUCTION

RECENTLY, with the rapid development of communication technology and the acceleration of smart cities, massive applications such as ultra-high-definition video and virtual reality are showing an explosive growth tendency [1], [2], [3], [4], [5]. The data generated by intelligent terminals have reached the zettabyte scale, and mobile network operators (MNOs) are facing unprecedented traffic pressure [6], [7]. Mobile edge computing (MEC) emerges as a promising technology that integrates distributed content caching and data prediction technology, by partially migrating storage and computing resources to network edges (e.g., base stations (BSs)) that users associate with in advance, can provide content acquisition services conveniently [8], [9]. As a result, mobile devices (MDs) can directly download content locally, which significantly reduces core network congestion and improves quality of service (QoS) [10], [11], [12], [13].

Although MEC networks have advantages in reducing content acquisition latency, there still exist several technical bottlenecks in performing content caching or user association in heterogeneous MEC environments, which can be summarized as follows: i) considering the limited storage of BSs in smart cities, it is impractical to cache all contents. Traditional content caching policies generally cache the most popular content but lack analysis about the regularity of historical knowledge of users' requests; ii) there are massively available BSs with different storage capabilities and service coverage in MEC networks. MD usually chooses the BS with faster transmission rates for association and neglects the impacts of content cache status on making user association decisions. It results in higher re-association rates and workload imbalance [14], [15].

Recently, extensive studies have been conducted on the issue of content caching or user association, and optimizing service capabilities in MEC networks. The studies in [16], [17] investigated the user association issue to maximize the total rate of coverage area in ultra-dense networks. However, these studies only discussed the issue of user association and content caching separately, which are not considered jointly. Actually, content

Manuscript received 25 April 2023; revised 25 July 2023; accepted 31 August 2023. Date of publication 6 September 2023; date of current version 8 January 2024. This work was supported in part by the National Key R&D Program of China under Grant 2022YFE0125400, in part by the National NSFC under Grants 62372072, 62102053, and 62072060, in part by the Chongqing Research Program of Basic Research and Frontier Technology under Grant cstc2022ycjhbzxm0058, in part by the Key Research Program of Chongqing Science and Technology Commission under Grant cstc2021jsex-dxwtBX0019, in part by the Science and Technology Plan Project of Chongqing Economic and Information Commission under Grant 2211R49R03, in part by the Haihe Lab of ITAI under Grant 22HHXCJC00002, in part by the Natural Science Foundation of Chongqing, China under Grant CSTB2022NSCQ-MSX1104, in part by the General Program of Chongqing Science and Technology Commission under Grants CSTB2022TIAD-GPX0017 and CSTB2022TIAD-STX0006, in part by the Regional Innovation Cooperation Project of Sichuan Province under Grant 2023YFQ0028, in part by Regional Science and Technology Innovation Cooperation Project of Chengdu City under Grant 2023-YF11-00023-HZ, and in part by the Guangdong Pearl River Talent Recruitment Program under Grants 2019ZT08X603 and 2019JC01X235. A preliminary version of the article appears as "Mobility-Aware Content Caching and User Association for Ultra-Dense Mobile Edge Computing Networks" in IEEE Global Communications Conference (GLOBECOM), Taibei, China, December 2020, [DOI: 10.1109/GLOBECOM42002.2020.9348257]. This article has made a significant extension on system modeling, scheme design and evaluation results. Recommended for acceptance by Dr. Pan Li. (*Corresponding author: Xiuhua Li.*)

Hui Li, Xiuhua Li, Chuan Sun, and Qilin Fan are with the School of Big Data & Software Engineering, Chongqing University, Chongqing 400000, China (e-mail: h.li@cqu.edu.cn; lixiuhua@cqu.edu.cn; c.sun@cqu.edu.cn; fan-qilin@cqu.edu.cn).

Fang Fang is with the Department of Electrical and Computer Engineering, Western University, London, ON N6A3K7, Canada, and also with the Department of Computer Science, Western University, London, ON N6A3K7, Canada (e-mail: fang.fang@uwo.ca).

Xiaofei Wang is with the Tianjin Key Laboratory of Advanced Networking, School of Computer Science and Technology, Tianjin University, Tianjin 300072, China (e-mail: xiaofeiwang@tju.edu.cn).

Victor C. M. Leung is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China, and also with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T1Z4, Canada (e-mail: vleung@ieee.org).

Digital Object Identifier 10.1109/TNSE.2023.3312369

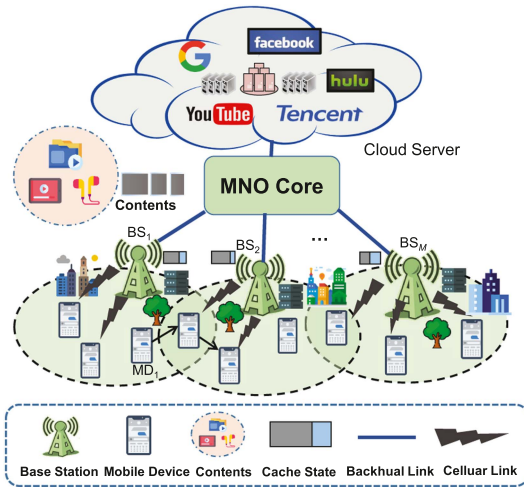


Fig. 1. Illustration of system model of MEC networks in smart cities.

caching and user association are interdependent and a reasonable BS association policy can satisfy content requests locally as well as provide MDs with sufficient resources. The study in [18] investigated the content placement issue in a small cell network to offload network traffic and decomposed the delay minimization problem into the user association subproblem and content placement subproblem. The authors in [19], [20] discussed a joint content caching and user association problem in heterogeneous wireless networks to minimize total delay and request miss ratio, respectively. However, these studies only considered static cases and ignored the possibility of re-association caused by user mobility. In a real-world MEC environment, users usually have high mobility, MDs' location information and available BSs may change over time. Hence, it is reasonable to design a content caching policy while considering user association information. We consider a practical motivated scenario as shown in Fig. 1. It can be observed that MD₁ is associated with BS₁ at the previous time slot. At the next time slot, MD₁ moves into the overlapping area served by BS₁ and BS₂. If BS₁ and BS₂ caches MD₁'s request content simultaneously, MD₁ should choose to associate with BS₁ rather than BS₂ to avoid re-association.

In this article, we are motivated to investigate the issue of joint content caching and user association with high user mobility in MEC networks of smart cities. In particular, we propose a MEC architecture that supports cooperative edge caching between the cloud server and BSs, where BSs are densely deployed. Our goal is to minimize the weighted sum of content acquisition latency caused by downloading contents from BSs, and handover latency caused by re-association. The main contributions are stated as follows:

- We present a three-layer MEC network in smart cities that support edge-cloud caching, for reducing content acquisition and handover latency as our major objective, towards future MEC networks. Besides, to exploit the issue of long-time scale content caching and short-time scale user association, we formulate the problem as a mixed time-scale non-convex binary integer linear programming (BILP) problem.

- To address the formulated problem, we propose a weighted LFM based intelligent content caching framework by introducing a weight matrix to associate different latent factors for learning intelligent long-time scale content caching policy at each time frame. Then, we design a short-time scale matching theory-based lazy re-association strategy for determining user association policy at each time slot.
- Together with theoretical analysis, we evaluate the performance of our proposed framework through extensive simulations on a real-world EUA dataset. Evaluation results demonstrate that our proposed framework has superior performance on reducing content acquisition latency and handover latency in the considered MEC network.

The remainder of this article is organized as follows. We discuss related work in Section II. Section III describes the system model and problem formulation. Then, we investigate our proposed framework in Section IV. Finally, evaluation results are presented in Section V, followed by concluding remarks in Section VI.

II. RELATED WORK

A. Content Caching for MEC

In terms of caching contents in MEC networks, several pioneering studies have been presented and achieved remarkable performance in reducing latency under different environment settings [21], [22], [23]. Li et al. [21] formulated a joint content placement and delivery problem for optimizing traffic offloading, and proposed a recurrent neural network and deep reinforcement learning algorithm to improve caching performance. The authors in [22] investigated the impact of content caching to improve the QoS in ultra-dense MEC networks and proposed a weighted double Q-learning method to deal with the decision problem. Zhang et al. [23] intended to explore the wireless coded caching policy to minimize transmission delay and cache replacement cost, and proposed a supervised deep learning-based method in continuous action space. However, these content caching strategies only considered the resource availability of BSs or the status of user equipment, and did not fully consider the historical knowledge of content requests. Recently, some studies began to investigate the impact of historical requests on content caching policy. Jiang et al. [24] considered a typical content caching scenario in MEC networks with user historical demand information to maximize the total expected caching reward and proposed a multi-agent reinforcement learning-based algorithm to address the formulated problem. Wu et al. [25] optimized the content caching policy to reduce the duplicated content transmissions in MEC networks with the consideration of user history request information. These studies have been evidenced to reduce content acquisition latency effectively based on historical knowledge of content requests.

B. User Association for MEC

User association is another crucial factor that affects the performance of MEC networks. Recently, extensive studies have been conducted on the issue of user association in MEC

networks [26], [27], [28], [29]. Teng et al. [26] investigated the user association problem in MEC networks under uncertain traffic demands and decomposed the original problem into a set of subproblems by formulating the BS states as a Markov chain. The authors in [27] investigated joint user association and resource allocation in the channel interference ultra-dense MEC network and proposed a multi-agent Q-learning based method to solve the joint optimization problem. The authors in [28] proposed a multi-agent Q-learning algorithm to jointly optimize user association and power allocation in heterogeneous networks. Simulation results demonstrate that the proposed framework achieves load balancing and improves energy efficiency. Zhang et al. [29] investigated user association and power allocation in mmWave-based energy harvesting MEC network under the constraints of load balance and QoS requirements, and proposed an iterative gradient user association and power allocation algorithm to address the formulated problem. However, these studies neglect the impacts of BSs' content cache status on making user association decisions, which is more likely to cause re-association in multiple available BSs association scenarios.

C. Joint Content Caching and User Association

In terms of joint content caching and user association in MEC networks. Dai et al. [30] proposed a reinforcement learning-based user association method and formulated the association process as a contextual multi-armed bandit problem in a two-tier full-duplex ultra-dense network, aiming at improving network utilization efficiency and alleviating backhauling traffic. Li et al. [31] investigated the issue of joint content caching and user association for densely deployed heterogeneous networks, and proposed a rapid association and delayed association method to reduce the content download latency. Yang et al. [32] established a joint user association and content caching framework in mobile networks to minimize the content transmission delay of MDs, and proposed an alternative algorithm for solving the corresponding optimization problem. However, these studies only considered user association cases in a static scenario, and ignored the possibility of re-association with high user mobility. In contrast, our work focuses on how to design joint content caching and user association policy in dynamic MEC networks with high user mobility under practical considerations of network constraints, which can be applied in realistic environments.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. MEC Network Architecture

As illustrated in Fig. 1, we consider a scenario of a three-layer MEC network in smart cities with a cloud server (CS), M densely deployed BSs (denoted by $\mathcal{M} = \{1, 2, \dots, M\}$) and U MDs (denoted by $\mathcal{U} = \{1, 2, \dots, U\}$), in which each BS is equipped with an edge server and can provide high-speed content delivery services for the MDs, such as smartphones, smartpads and wearable devices. Considering comprehensive coverage of hotspot areas and improvement of service capabilities, the service areas of BSs in the considered MEC system are overlapped. Each BS has limited cache capacities and is connected to MNO

TABLE I
SUMMARY OF IMPORTANT NOTATIONS

Notation	Definition
$\mathcal{M}(M)$	The set (number) of base stations
$\mathcal{U}(U)$	The set (number) of mobile devices
$\mathcal{F}(F)$	The set (number) of contents
$\mathcal{T}, \mathcal{T}'$	The set of timeline and time frame
\mathcal{C}	The set of content caching policy
\mathcal{A}	The set of user association policy
\mathcal{R}	The matrix of user requests
$c_{m,f}^t$	The variable for whether BS m caches content f
$a_{u,m}^t$	The variable for whether MD u associates BS m
$r_{u,f}^t$	The indicator for whether MD u requests content f at time slot t
s_f	The storage size of the content f
S_m	The storage capacity of BS m
$D_{m,f}$	The latency for obtaining the content f at BS m
$d_{m,f}^L$	The latency for obtaining content f at BS m locally
$d_{m,f}^{C \rightarrow B}$	The latency for fetching content f via backhaul link
$D_{m,f}^C$	The content acquisition latency at time slot t
$D_{m,f}^H$	The handover latency at time slot t
σ_u^t	The BS that MD u associates with at time slot t
$\delta_{m,n}$	The handover latency between BS m and BS n
P, Q	The user-feature and content-feature matrix
W	The BS-content weight matrix
G	The content requests matrix
p_m, q_f	The feature vector of BS m and content f
α	The coefficient to balances the content acquisition latency and the handover latency
$\lambda_m, \lambda_f, \lambda_w$	The penalty coefficient
L, μ	The loss function and learning rate
$s_{m,u}$	The cosine similarity between BS m and MD u

through high-capacity cables or optical fibers. The CS and BSs are cooperative to build a stable edge-cloud cache network, and the MDs can directly obtain the desired contents from the CS when the BS cache misses. Besides, the MDs are randomly distributed in the service area of BSs and the location of MDs may change over time. We assume that the user's mobility information is already known in this article. The MEC network operates in a slotted manner and the timeline is equally divided into a series of slots (denoted by $\mathcal{T} = \{1, 2, \dots, t, \dots, T\}$). Each time slot $t \in \mathcal{T}$ has a reasonable time span (e.g., several seconds or minutes), which is determined by user request frequency and system measurement. To describe content caching status more effectively, we divide the timeline \mathcal{T} into multiple equal time frames (denoted by $\mathcal{T}' = \{T_1, T_2, \dots, T_j, \dots, T_J\}$). Each time frame T_j contains multiple time slots and the MNO will perform a content caching decision according to the content request at the beginning of each time frame. As a result, we have $\mathcal{T} = T_1 \cup T_2 \cup \dots \cup T_J$, and $T_i \cap T_j = \emptyset, \forall i \neq j$. Some key parameters are listed in Table I.

B. Content Caching Model

There is a catalog of F contents in the considered MEC network (denoted by $\mathcal{F} = \{1, 2, \dots, F\}$), which are indivisible. We assume that the CS has enough storage capability to cache all contents but edge servers can only cache partial contents. Denote $c_{m,f}^j \in \{0, 1\}$ as the content caching policy of BS m ,

where $c_{m,f}^j = 1$ indicates that BS m has cached content f at time frame j , otherwise $c_{m,f}^j = 0$. Thus, we have the constrain about limited storage of edge servers

$$\sum_{f=1}^{\mathcal{F}} c_{m,f}^j s_f \leq S_m, \forall m \in \mathcal{M}, \forall j \in \mathcal{T}', \quad (1)$$

where s_f is the storage size of content f , S_m is the storage capacity of BS m . Moreover, we assume that content popularity changes slowly and can be determined in advance by using machine learning methods or analyzing the relationship between users' behavior and preference [33], [34].

Furthermore, we calculate content acquisition latency in terms of cache hits and misses. When the requested content is not cached in the BSs, we need to get the contents from the CS through the MNO core. Thus, the latency for obtaining content f at BS m at time slot t of MD u can be expressed as

$$D_{u,m,f}^t = c_{m,f}^j d_{u,m,f}^{t,L} + (1 - c_{m,f}^j) (d_{u,m,f}^{t,L} + d_{m,f}^{t,C \rightarrow B}), \quad (2)$$

where $d_{u,m,f}^{t,L} = \frac{s_f}{R_{u,m}^t}$ is the latency for obtaining content f locally at BS m , $R_{u,m}^t$ is the achievable transmission rate. $d_{m,f}^{t,C \rightarrow B}$ is the latency for fetching content f from CS to BS m . The first term on the right refers to the latency when the cache hits, and the second term means the cache misses.

In addition, we assume that each MD sends a content request at each time slot. Denote $\mathcal{R} = \{r_{u,f}^t | \forall u \in \mathcal{U}, \forall f \in \mathcal{F}, \forall t \in \mathcal{T}\}$ as the matrix of all MDs' requests, where $r_{u,f}^t = 1$ denotes that MD u requests content f at time slot t , otherwise $r_{u,f}^t = 0$. Furthermore, the latency for content acquisition at time slot t can be calculated as

$$D_C^t = \sum_{m=1}^{\mathcal{M}} \sum_{u=1}^{\mathcal{U}} \sum_{f=1}^{\mathcal{F}} r_{u,f}^t a_{u,m}^t D_{u,m,f}^t, \forall t \in \mathcal{T}, \quad (3)$$

where $a_{u,m}^t \in \{0, 1\}$ is the user association policy, and we will introduce the detail in the next section.

C. User Association Model

Moreover, to get the desired contents, each MD needs to associate with a suitable BS at each time slot. An excellent user association strategy can not only speed up content acquisition, but also meet users' requests locally. Denote a binary indicator $a_{u,m}^t \in \{0, 1\}$ for whether MD u is associated with BS m or not at time slot t . Note that the number of associated BS at the same time slot should be equal to 1, which can be constrained as

$$\sum_{m=1}^{\mathcal{M}} a_{u,m}^t = 1, \forall u \in \mathcal{U}, \forall t \in \mathcal{T}. \quad (4)$$

However, in dynamic scenarios, due to the diversity of requested contents and user mobility, the case of user re-association is inevitable. When re-association occurs, the MDs may suffer from service interruptions and a loss of QoS. The BSs should update the location of service profile data, which records MDs' runtime environment (e.g., the latest running status, operating

system, and software information), and it will cause handover latency. In this article, the handover latency between BS m and BS n can be calculated as a time consumption for transmitting the MDs' service profile data from BS m to BS n , which can be quantified as $\delta_{m,n}$. We assume that the user profile data size and the bandwidth between BSs are almost the same, thus, $\delta_{m,n}$ can be determined as a constant in advance. Define $\Gamma = \{\sigma_u^t | \sigma_u^t \in \mathcal{M}, u \in \mathcal{U}, \forall t \in \mathcal{T}\}$ as the set of corresponding associated BS, and the element σ_u^t represents the BS that user u associates with at time slot t . Thus, the total handover latency at time slot t can be expressed as

$$D_H^t = \sum_{u=1}^{\mathcal{U}} \delta_{\sigma_{u,t-1}^t, \sigma_u^t}, \forall t \in \mathcal{T}. \quad (5)$$

D. Problem Formulation

In the considered MEC network, to improve QoS and satisfy MDs' content requests locally, our objective is to minimize the weighted sum of content acquisition latency caused by downloading contents from the BSs, and handover latency caused by re-association. By joint optimizing the content caching policy $\mathcal{C} = \{c_{m,f}^j | \forall m \in \mathcal{M}, \forall f \in \mathcal{F}, \forall j \in \mathcal{T}'\}$ and the user association policy $\mathcal{A} = \{a_{u,m}^t | \forall m \in \mathcal{M}, \forall u \in \mathcal{U}, \forall t \in \mathcal{T}\}$, the corresponding weighted sum latency minimization problem can be formulated as

$$\mathcal{P} : \min_{\mathcal{A}, \mathcal{C}} \sum_{t=1}^{\mathcal{T}} (\alpha D_C^t + (1 - \alpha) D_H^t) \quad (6a)$$

$$\text{s.t. C1 : } \sum_{f=1}^{\mathcal{F}} c_{m,f}^j d_f \leq S_m, \forall m \in \mathcal{M}, \forall j \in \mathcal{T}', \quad (6b)$$

$$\text{C2 : } \sum_{m=1}^{\mathcal{M}} a_{u,m}^t = 1, \forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \quad (6c)$$

$$\text{C3 : } c_{m,f}^j \in \{0, 1\}, \forall m \in \mathcal{M}, \forall f \in \mathcal{F}, \forall j \in \mathcal{T}', \quad (6d)$$

$$\text{C4 : } a_{u,m}^t \in \{0, 1\}, \forall m \in \mathcal{M}, \forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \quad (6e)$$

where coefficient α balances content acquisition latency and handover latency. The constraints can be explained as follows: constraint C1 states that the contents' size can not exceed the total available storage capacity at each BS; constraint C2 restricts each MD can establish at most one association with the BS; Constraints C3 and C4 show that the content caching and user association policy is 0-1 variable. The origin problem is a mixed time-scale non-convex BILP problem, which is NP-hard. We focus on designing a low-complexity intelligent content caching and user association framework to obtain a competitive suboptimal solution.

IV. PROPOSED FRAMEWORK DESIGN

In this section, we first analyze the formulated problem and decompose it into two subproblems. Then, we present the corresponding algorithm for solving. Finally, we give a detailed analysis of computational complexity.

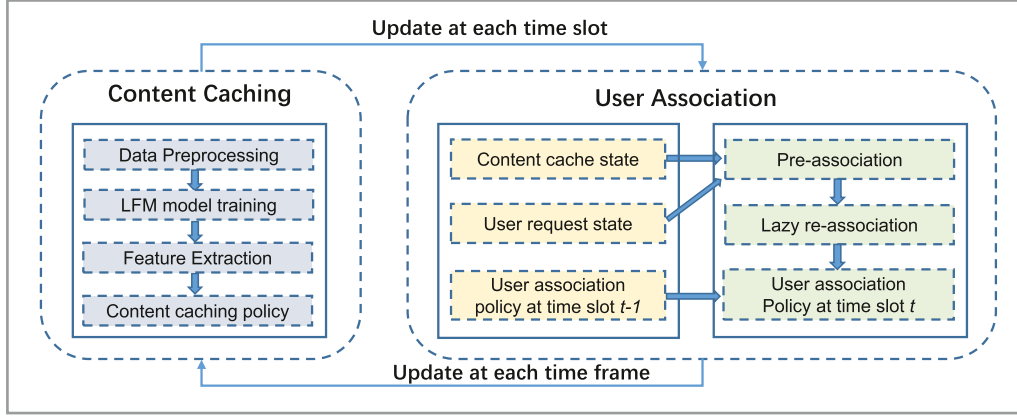


Fig. 2. Intelligent content caching and user association framework in the considered MEC network.

A. Problem Analysis

Content caching and user association have different time scales. In practice, a short-time scale user association policy can affect content caching policy for next time frame. Long-time scale content caching decision also affects short-time scale user association decision in the current time frame. Besides, it exists coupling between content caching policy \mathcal{C} and user association policy \mathcal{A} in term D_C^t , and the internal correlation of user association policy of two consecutive adjacent time slots in term D_H^t in the objective function. It is generally challenging to obtain the optimal solution through traditional optimization methods. To solve it, we optimize the original mixed time-scale problem in two stages:

- i) Long-time scale content caching. Content requests are generally relevant in a long-time scale. Hence, we can learn from the historical request information, and then determine the content caching strategy. Specifically, at the beginning of each time frame j , we can determine content caching policy according to the user association information and content request at the previous time frame $j - 1$.
- ii) Short-time scale user association. The handover latency is related to the user association policy at two consecutive adjacent time slots, and is affected by the content caching strategy of BSs to some extent. At each time slot t , we associate MDs with the BSs at the time slot $t - 1$ to reduce handover latency and match the MDs to BSs that can satisfy user content requests locally as much as possible.

Motivated by this, we propose an intelligent content caching framework based on the weighted LFM, which can learn from historical requests information. After that, with the obtained content caching policy just obtained, we design the matching theory-based lazy re-association strategy for determining a user association policy at each time slot. The detailed process is shown in Fig. 2.

B. Weighted LFM Based Intelligent Content Caching

At this stage, we intend to determine the content caching policy by using the weighted LFM method. LFM is a kind of

implicit semantic model and uses matrix factorization technology to process the content scoring matrix according to machine learning and optimization theory, thereby obtaining the potential feature of the users and contents. It is currently widely used in the field of recommendation systems [35], [36]. In this article, we have made certain improvements based on the basic LFM model to make it applicable for the field of the considered MEC scenarios.

Specifically, at each time frame j , we use the historical request information at the previous time frame $j - 1$ as the training sample. Then we construct the content request matrix G at each BS based on user association policy at time frame $j - 1$. User association policy at time frame $j - 1$ can be used to determine which BSs should cache content and MDs should associate with. We normalize the user content request matrix to approximate the user-content score matrix. After that, we propose a weighted LFM algorithm based on the basic LFM model, which introduces a BS-content weight matrix to associate the latent factors (i.e. BS-feature matrix and content-feature matrix), such that the latent factors could be used to estimate content request properly. After modeling by weighted LFM, the content request matrix G can be decomposed into three low-dimensional matrices (i.e., BS-feature matrix \mathcal{P} , content-feature matrix \mathcal{Q} and BS-content weight matrix \mathcal{W}) and the BS-content weight matrix \mathcal{W} can be interpreted as BS's attention to different contents. By multiplying these latent factors, the predicted content request probability can be calculated (as shown in Fig. 3). Finally, we determine the content caching policy based on the generated content request matrix. The detailed process can be shown as follows.

1) *Data Preprocessing*: In this phase, to improve the accuracy and convergence speed of the model, we need to normalize the user history request data at time frame $j - 1$ as the content request probability at each BS to approximate the user-content score matrix, which can be processed as

$$g_{m,f} = \frac{g'_{m,f}{}^{j-1}}{\sum_{f=1}^{\mathcal{F}} g'_{m,f}{}^{j-1}}, \forall m \in \mathcal{M}, \forall f \in \mathcal{F}, \quad (7)$$

where $g'_{m,f}{}^{j-1}$ is the total request number for content f at BS m at time frame $j - 1$ and $g_{m,f}$ is the normalized content request probability. After the normalization operation, the users'

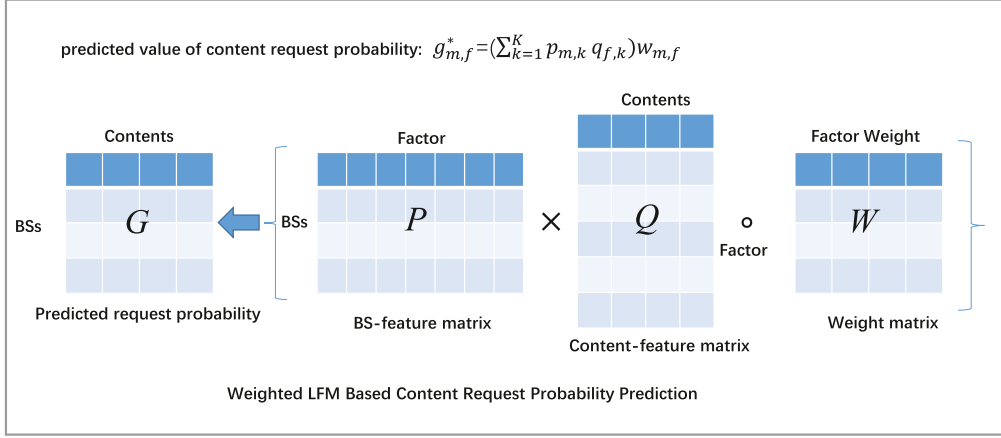


Fig. 3. Overview of the proposed weighted LFM algorithm for content caching.

requested data can be normalized in the range of [0, 1] for subsequent model training.

2) *Weighted LFM Model Training*: Denote K as the number of hidden features. Denote the set $P = \{p_{m,k} | \forall m \in \mathcal{M}, \forall k \leq K\}$, $Q = \{q_{f,k} | \forall f \in \mathcal{F}, \forall k \leq K\}$ and $W = \{w_{m,f} | \forall m \in \mathcal{M}, \forall f \in \mathcal{F}\}$ as the BS-feature matrix, content-feature matrix and BS-content weight matrix, respectively, where k is the index of hidden features. Hence, we can get the decomposition formula as

$$G = (PQ^T) \circ W, \quad (8)$$

where G is the content requests probability matrix and has many missing values in the beginning state, the symbol \circ is the Hadamard product. Each element in matrix G can also be calculated as

$$g_{m,f}^* = \left(\sum_{k=1}^K p_{m,k} q_{f,k} \right) w_{m,f}, \forall m \in \mathcal{M}, \forall f \in \mathcal{F}, \quad (9)$$

where the elements $g_{m,f}^*$ is the predicted value of content request probability. Our goal is to minimize the gap between the true value and the predicted value. Hence, the loss function can be defined as

$$\begin{aligned} L(p, q) &= \sum_{g_{m,f} \in Tra} (g_{m,f} - g_{m,f}^*)^2 \\ &= \sum_{g_{m,f} \in Tra} \left(g_{m,f} - \left(\sum_{k=1}^K p_{m,k} q_{f,k} \right) w_{m,f} \right)^2, \end{aligned} \quad (10)$$

where Tra is the training data on the training dataset.

3) *Feature Extraction*: To avoid over fitting in the training process, we add the L_2 penalty term $\lambda_m \|p_m\|^2 + \lambda_f \|q_f\|^2 + \lambda_w \|w_{m,f}\|^2$ into the objective function $L(p, q)$, where $p_m = [p_{m,1}, p_{m,2}, \dots, p_{m,k}, \dots, p_{m,K}]$ and $q_f = [q_{f,1}, q_{f,2}, \dots, q_{f,k}, \dots, q_{f,K}]$ is the feature vector of BS m and content f , respectively. λ_m, λ_f and λ_w is the penalty

coefficient. Thus, the objective function can be transformed into

$$\begin{aligned} L(p, q) &= \sum_{g_{m,f} \in Tra} \left(g_{m,f} - \left(\sum_{k=1}^K p_{m,k} q_{f,k} \right) w_{m,f} \right)^2 \\ &\quad + \lambda_m \|p_m\|^2 + \lambda_f \|q_f\|^2 + \lambda_w \|w_{m,f}\|^2. \end{aligned} \quad (11)$$

To minimize the loss function, we first need to find the partial derivative of the objective function $L(p, q)$ with respect to the parameters $p_{m,k}$, $q_{f,k}$ and $w_{m,f}$, respectively. Define $\Delta g_{m,f} = g_{m,f} - (\sum_{k=1}^K p_{m,k} q_{f,k}) w_{m,f}$, the partial derivative of the objective function can be expressed as

$$\begin{cases} \frac{\partial L(p, q)}{\partial p_{m,k}} = -2\Delta g_{m,f} q_{f,k} w_{m,f} + 2\lambda_m p_{m,k}, \forall m \in \mathcal{M}, \forall k, \\ \frac{\partial L(p, q)}{\partial q_{f,k}} = -2\Delta g_{m,f} p_{m,k} w_{m,f} + 2\lambda_f q_{f,k}, \forall f \in \mathcal{F}, \forall k, \\ \frac{\partial L(p, q)}{\partial w_{m,f}} = -2\Delta g_{m,f} \left(\sum_{k=1}^K p_{m,k} q_{f,k} \right) + 2\lambda_w w_{m,f}, \\ \quad \forall m \in \mathcal{M}, \forall f \in \mathcal{F}. \end{cases} \quad (12)$$

By using the stochastic gradient descent method, parameters $p_{m,k}$, $q_{f,k}$ and $w_{m,f}$ can be updated by continuous iteration until convergence. The form can be shown as

$$\begin{cases} p_{m,k} = p_{m,k} + \mu (\Delta g_{m,f} q_{f,k} w_{m,f} - \lambda_m p_{m,k}), \\ \quad \forall m \in \mathcal{M}, \forall k \leq K, \\ q_{f,k} = q_{f,k} + \mu (\Delta g_{m,f} p_{m,k} w_{m,f} - \lambda_f q_{f,k}), \\ \quad \forall f \in \mathcal{F}, \forall k \leq K, \\ w_{m,f} = w_{m,f} + \mu (\Delta g_{m,f} (\sum_{k=1}^K p_{m,k} q_{f,k}) - \lambda_w w_{m,f}), \\ \quad \forall m \in \mathcal{M}, f \in \mathcal{F}, \end{cases} \quad (13)$$

where μ denotes the learning rate. In order to ensure the convergence of the proposed algorithm and increase the iteration speed, the learning rate μ needs to be carefully designed and dynamically adjusted. Specifically, we update the learning rate $\mu = \mu * 0.95$ at each episode. The proposed algorithm uses a large learning rate μ at the initial stage to quickly approach the optimal value and gradually reduces the value of the learning rate μ in the subsequent stages to ensure that the model converges stably.

Algorithm 1: Weighted LFM Based Intelligent Content Caching Algorithm at Each Time Frame.

- 1 **Input:** All BSs information, user history request \mathcal{H} .
 - 2 **Initialize:** loss function $L(p, q)$, learning rate μ , where μ is a positive number no larger than 1, feature vector p_m and q_f , and weight matrix \mathcal{W} .
 - 3 Normalize users' request data according to (7).
 - 4 Define the loss function according to (10) and take the partial derivative of the objective function $L(p, q)$ with respect to the parameters $p_{m,k}$, $q_{f,k}$ and $w_{m,f}$ according to (12).
 - 5 **while** loss function $L(p, q)$ is not convergence **do**
 - 6 Update the parameters $p_{m,k}$, $q_{f,k}$ and $w_{m,f}$ according to (13).
 - 7 Adjust the learning rate μ .
 - 8 **end**
 - 9 Determine the content caching policy based on the generated content request probability matrix G .
 - 10 **Output:** Content caching policy \mathcal{C} .
-

4) *Content Caching Policy:* Through the above steps, we can reconstruct the new content request probability matrix G . The matrix G can be reconstructed in a parallel manner to speed up algorithm convergence and the rows of matrix G represent the probabilities of different content requests. Then, according to different cache sizes of BSs, we cache the contents in turn according to the probability, until the available cache capability of BS is full. The detailed process of the proposed algorithm is given in Algorithm 1.

C. Matching Theory-Based User Association

In the considered MEC network, a reasonable user association policy can ensure MDs associate with an appropriate BS, which not only satisfies content requests locally but also provides sufficient communication resources. Considering the internal correlation of user association policy at two adjacent time slots, in this section, we design a matching theory-based lazy re-association strategy to address the challenges at each time slot to reduce the handover latency.

The problem to determine user association policy can be modeled as a bipartite graph matching problem. Specifically, we define the set of MDs and BSs as two disjoint sets in a bipartite graph. Then, our goal is to match the set of MDs to the set of BSs so that the content acquisition latency D_C^t at each time slot can be minimized. Mathematically, the considered matching process can be stated as follow.

Definition 1: Given two disjoint sets, the BSs set \mathcal{M} and the MDs set \mathcal{U} , an one-to-many matching φ can be defined as a mapping from the set $\mathcal{M} \cup \mathcal{U}$ into the set $\mathcal{M} \cup \mathcal{U}$ so that for each $m \in \mathcal{M}$ and $u \in \mathcal{U}$ satisfy the conditions:

- 1) $\varphi(u) \in \mathcal{M}$ and $|\varphi(u)| = 1$;
- 2) $\varphi(m) \subseteq \mathcal{U}$;
- 3) $m = \varphi(u) \Leftrightarrow u \in \varphi(m)$.

where $|\varphi(\cdot)|$ is the size of the matching outcome. The first condition states that each MD only can associate with one BS.

The second condition indicates that the MDs that are associated with BS m must be a subset of the set \mathcal{U} .

The detailed process of the proposed strategy at each time slot can be shown in Algorithm 2, which includes pre-association, lazy re-association, swap and update matching three parts. Among them, pre-association and lazy re-association respectively correspond to the two stages of user association in the green rectangle in Fig. 2. The swap and update matching process corresponds to the final step of determining the user association policy at time slot t .

1) *Pre-Association:* In this step, we first perform the pre-association operation based on user historical request data \mathcal{H} . We intend to calculate the cosine similarity between the MDs and the BSs, which reflects the *distance* between the user request and the BS cache state to a certain extent. The cosine similarity between BS m and MD u can be defined as

$$s_{m,u} = \frac{\sum_{f=1}^F c_{m,f} e_{u,f}}{\sqrt{\sum_{f=1}^F c_{m,f}^2} \sqrt{\sum_{f=1}^F e_{u,f}^2}}, \forall m \in \mathcal{M}, \forall u \in \mathcal{U}, \quad (14)$$

where $e_{u,f} \in \{0, 1\}$ indicates that MD u request content f or not. The higher similarity means that the cache status of the BS can better satisfy user requests, and it is more likely to be associated with this BS in the future time slot. In the initial phase, we associate the MDs with the BSs with the highest degree of cosine similarity.

2) *Lazy Re-Association:* In this step, we determine the user association policy at each time slot t . We jointly consider the communication quality and content caching to minimize the re-association as much as possible. Specifically, we set a tolerable minimum transmission rate threshold R_u^{\min} . When the current transmission rate $R_{u,m}^t$ is greater than the tolerable minimum transmission rate R_u^{\min} , the communication quality is reliable. Meanwhile, based on the user association policy at previous time slot $t-1$, when requested content f at time slot t can be satisfied by the BS which is associated with at previous time slot $t-1$, i.e., the condition $R_{u,m}^t \geq R_u^{\min}$ and $c_{u^{t-1},f}^j = 1$ holds, we keep the same association unchanged. That is, we try to reduce the occurrence of re-association as much as possible. As a result, the handover latency caused by re-association can be significantly decreased.

3) *Swap and Update Matching:* Finally, we use the matching theory-based methods to swap two different MDs reasonably so that they can be associated with a reasonable BS simultaneously. Denote the set $\Lambda = [\varphi^1, \varphi^2, \dots, \varphi^t, \dots, \varphi^T]$ as the matching sequence of all swap processes at all time slots, and each element $\varphi^t = [\varphi_0^t, \varphi_1^t, \varphi_2^t, \dots, \varphi_i^t, \dots]$ is the set of all matching processes at each time slot t , where i is the index of swapping operation. After a series of swap and update matching operations between the MDs, the content acquisition latency D_C^t at time slot t will eventually converge. The matching process changes as below

$$\varphi_0^t \rightarrow \varphi_1^t \rightarrow \varphi_2^t \rightarrow \dots \rightarrow \varphi_i^t \rightarrow \varphi_{i+1}^t \rightarrow \dots \rightarrow \quad (15)$$

Note that the swap and update matching operations are permitted if and only if the content acquisition latency D_C^t at each time slot is decreased after the swapping operation. If the condition

holds, we will update the user association policy between the two different MDs.

Theorem 1: The proposed swap and update matching procedure can converge after the finite swap and update matching operations. Meanwhile, the swap and update matching operations converge at most $|U|^2$ steps.

Proof 1: What we need to do is to prove that each element $\varphi^t = [\varphi_0^t, \varphi_1^t, \varphi_2^t, \dots, \varphi_i^t, \dots]$ at each time slot in the set Λ is monotonically decreasing and has a lower bound, where φ_0^t is the matching result at the initial phase at time slot t . Denote the $(D_C^t)_i$ as the corresponding content acquisition latency at swapping operation i . As mentioned above, the swap operations are permitted if and only if the content acquisition latency D_C^t is decreased after swapping. That is, for each swapping operation i , the condition $(D_C^t)_i > (D_C^t)_{i+1}, \forall i$ holds. So, the corresponding content acquisition latency of set $\varphi^t = [\varphi_0^t, \varphi_1^t, \varphi_2^t, \dots, \varphi_i^t, \dots]$ is monotonically decreasing obviously. Meanwhile, considering the limited communication and computing resources, the total content acquisition latency is always greater than 0, so the condition $(D_C^t)_i \geq 0, \forall i$ holds. Hence, the procedure can converge after the finite swap and update matching operations. Meanwhile, since we only consider exchanges between two different MDs, the total number of swap and update matching operations is less than $|U|^2$. Thus, the swap and update matching operations converge at most $|U|^2$ steps.

D. Time and Space Complexity Analysis

We analyze the complexity of the proposed framework from time and space aspect: i) in terms of time complexity, for Algorithm 1, the time complexity comes down to the parameters update of BS-feature matrix and content-feature matrix, hence the time complexity with gradient descent is $O(D_1 \cdot N \cdot K)$, where D_1 is the size of training sample; N is the number of iteration. Compared with the pre-association and lazy re-association procedure, since the number of MDs is much larger than BSs. The main computation of Algorithm 2 comes from the swap and update matching procedure. Hence, the time complexity of Algorithm 2 is $O(|U|^2)$; ii) in terms of space complexity, Algorithm 1 needs to store the BS-feature matrix and content-feature matrix. The required memory space is the sum of the BSs-feature matrix, content-feature matrix and BS-content weight matrix. Hence, the space complexity is $O((M + F) \cdot K + M \cdot F)$. Algorithm 2 needs to store the request matrix, user association policy at the previous time $t - 1$, and user history request data. Thus, the total space complexity is $O(U \cdot F \cdot T + M \cdot U + D_2)$, where D_2 is the size of user history request information.

V. PERFORMANCE EVALUATION

A. Parameter Settings

In this section, we evaluate the performance of our proposed framework through extensive simulations in real-world MEC environments. The experiments are conducted on a real dataset that covers a 6.2 km² central business district region located in Melbourne, Australia. Specifically, the dataset contains the

Algorithm 2: Matching Theory-based Lazy Re-association Strategy at Each Time Slot.

```

1 Input: All the MDs and BSs information, content
   caching policy  $\mathcal{C}$ , request matrix  $\mathcal{R}$  and user history
   request data  $\mathcal{H}$ , the user association policy at
   previous time slot  $t - 1$ .
2 Initialize: The content acquisition latency  $D_C^t \leftarrow 0$ ,
   the handover latency  $D_H^t \leftarrow 0, i = 0$ .
3 —Step 1. [Pre-association]
4 if the time slot  $t = 0$  then
5     Calculate the cosine similarity  $s_{m,u}$  between BS
        $m$  and MD  $u$  according to (14) and perform
       pre-association operation with the highest
       degree of cosine similarity.
6 end
7 —Step 2. [Lazy re-association]
8 for each  $u \in \mathcal{U}$  do
9     if MD  $u$ 's request can be satisfied by the BS
       associated at time slot  $t - 1$  and  $\sigma_u^t = \sigma_u^{t-1}$  then
10        Keep the same association.
11    end
12 end
13 —Step 3. [Swap and update matching]
14 while value  $D_C^t$  does not converges or  $i < I_{max}$  do
15     for each  $u', u'' \in \mathcal{U}$  and  $u' \neq u''$  do
16         if  $\varphi(u') \neq \varphi(u'')$  then
17             Calculate latency  $(D_C^t)_i$  after swapping.
18             if the latency  $(D_C^t)_i$  decreases then
19                 Update the user association policy for
20                 MD  $u'$  and MD  $u''$ .
21             end
22         end
23     end
24 end
25 Calculate latency  $D_C^t$  and  $D_H^t$  at time slot  $t$ .
26 Output: User association policy  $a_{u,m}^t$  at time slot  $t$ .
```

geographic distribution information of real-world 125 edge BSs, 816 edge MDs and their service range, which is obtained by Australian Communications and Media Authority [37], [38]. Fig. 4(a) from Google Map¹ shows the location of the BSs, MDs and their service range. We choose a MEC network composed of $M = 10$ BSs and $U = 200$ MDs (default) in the region. For simulation purposes, we set the size of content catalog F as 80 and generate a certain number of user requests as a content dataset. Meanwhile, the content dataset is divided into the training dataset and test dataset based on cross-validation rules. Fig. 4(b) shows the relationship between the number of content requests and the content index in the test dataset, which includes 4000 content requests totally. The storage size of each BS as the percentage of the total content size. We use the Random-Waypoint mobility model to generate the MDs' moving path. The sizes of content are randomly set in the range

¹<https://github.com/swinedge/eua-dataset>.

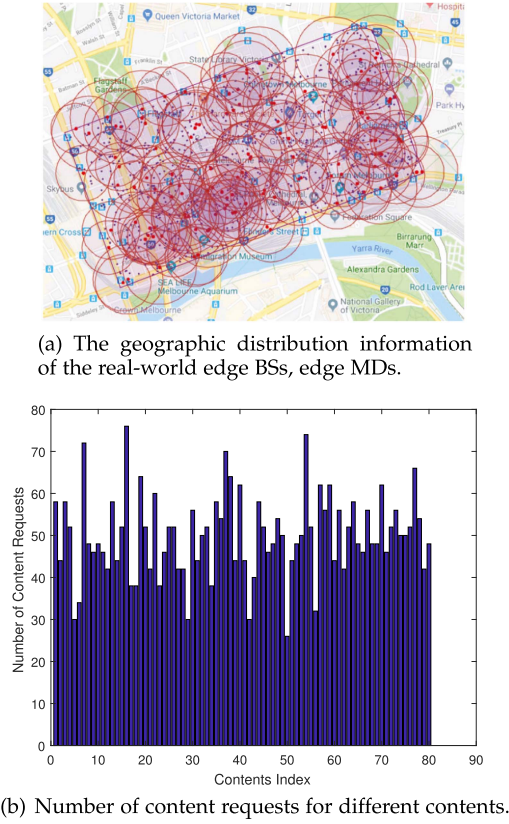


Fig. 4. Geographic distribution information of real-world dataset and the number of requests for different contents.

of [5, 20] Mbits and the storage space of each BS is set as the percentage of the total content size. We set tolerable minimum transmission rate R_u^{\min} as one-fifth of the transmission rate achieved under favorable conditions. Besides, we simulate $T = 20$ time slots, and the interval of each time slot length is within a reasonable range and equal.

B. Performance Metrics and Baseline Schemes

To evaluate the effectiveness of our proposed framework, we use four common performance metrics in the MEC network as: 1) *Total latency*, including content acquisition latency and handover latency two parts; 2) *Cache hit ratio*, denoting the percentage of supported requests by the MDs; 3) *Handover latency*, denoting the latency of transmitting the MDs' service profile data between BSs; 4) *Percentage of traffic offload*, denoting the reduced traffic of obtaining content through the backhaul link;

In addition, we compare our proposed framework with the following baseline schemes as: Deep learning methods (including *Multi-Armed Bandit (MAB)* [30] and *Multi-agent reinforcement learning (MARL)* [24]), traditional optimization methods (including *Joint Content Caching and User Association (JCC-UA)* [31] and *Constraints and Convex-Concave Procedure (CCCP)* [25]), and classical methods (including *Most Popular Caching and the Proposed User Association strategy (MPCPUA)* [39], *Least Recently Used caching and Meet Current Request association strategy (LRUMCR)* [40], *Most Recently Used caching and the Proposed User Association*

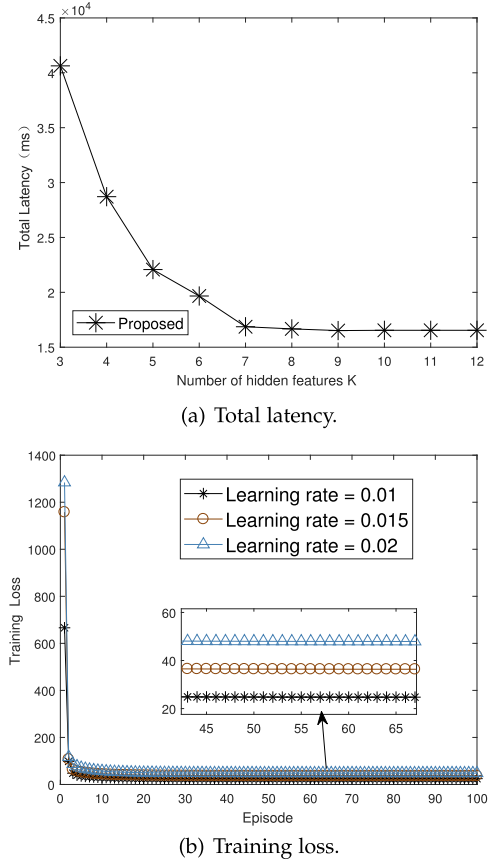


Fig. 5. Total latency under different numbers of hidden features K and the training loss at each episode with different learning rates.

strategy (*MRUPUA*) and *No Caching and Random Association strategy (NCRA)*). To explain it, *MAB* and *MARL* as deep learning approaches for solving the cache-aware user association problem; *JCC-UA* and *CCCP* used as traditional optimization methods to address the similar case in the article; *MPCPUA*: Each BS caches the most popular contents and the proposed user association policy used in this article; *LRUMCR*: derived from [40] to address the same problem; *MRUPUA*: Each BS first clears the most commonly used content, which may be outdated or out of time; *NCRA*: Each BS not caches any contents and the MD associate with the BS randomly to fetch the desired content from the CS directly.

C. Hyperparameters Setting and Convergence

In this subsection, we investigate the impact of the different numbers of hidden features and the convergence of the proposed algorithms under different learning rates in Fig. 5. Specifically, we analyze the total latency difference under different hyperparameters K and present the convergence performance in terms of the loss function at each episode.

Fig. 5(a) compares the performance of the total latency versus the hyperparameters K of the hidden features. It can be observed that the total latency decreases as the number of hidden features increases. Specifically, we find that when the number of hidden features is less than 7, the total delay decreases faster, and when

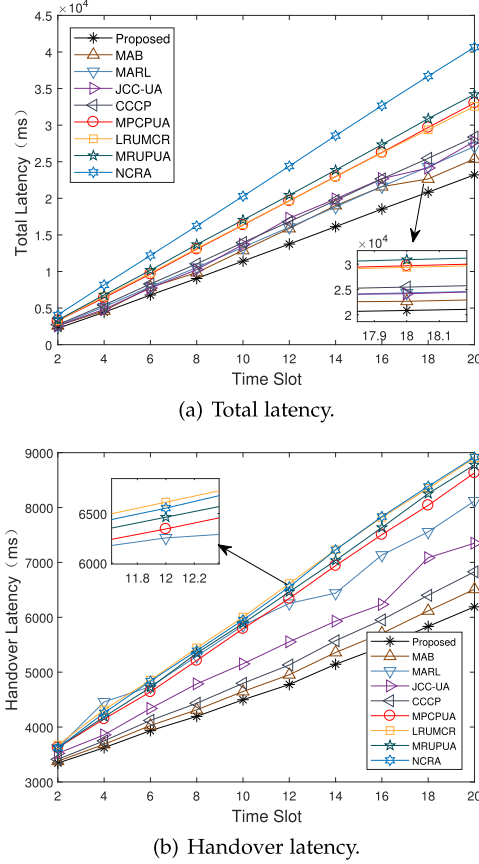


Fig. 6. Total latency and the handover latency at different time slots.

it is greater than or equal to 7, the total delay remains almost unchanged. Hence, we can conclude that when the number of hidden features $K = 7$, the algorithm efficiency can be optimal and we will adopt this setting in subsequent experiments. Fig. 5(b) shows the convergence curve in terms of the training loss on each episode. Clearly, as the episode increases, the training loss under different learning rates decreases continuously and finally converges. Specifically, when the initial learning rate is 0.01, the convergence of the proposed algorithms is faster compared with other learning rates. Besides, we can easily find that the training loss curve drops quickly at the beginning, and then it drops slowly. Hence, under different initial learning rate settings, the proposed framework with the initial learning rate $\alpha = 0.01$ has better training efficiency.

D. Total and Handover Latency of Different Time Slots

In this subsection, we calculate the total latency and handover latency at different time slots in Fig. 6. Specifically, we conduct the experiments on 200 MDs and the cache ratio of each BS is set as 25% of total contents.

Fig. 6(a) evaluates the performance of our proposed framework in reducing the total latency compared with these baseline schemes at each time slot. We can observe that the total latency increases monotonically and the gradients of the total latency curves remain almost unchanged. As the time slot increases, the effectiveness of the proposed algorithm in reducing the total

latency is more significant. Specifically, the proposed framework can achieve 22.4%, 22.9% and 24.2% performance improvement at a quarter, half, and all of the total time slots compared with the CCCP strategy. Meanwhile, the proposed framework can achieve around 9% and 8% performance improvement compared with the JCC-UA and MAB strategies. Fig. 6(b) proves that the proposed framework can reduce the handover latency significantly. It can be obviously observed that the handover latency gap is larger than the gap of total latency compared with the other baseline schemes. This is because the proposed framework can optimize the content caching policy with historical knowledge, and the matching theory-based lazy re-association strategy can avoid a certain amount of re-association, thereby reducing the handover latency. To a certain degree, we conclude that the proposed framework can associate the MDs with more suitable BSs.

E. Effects of Different Cache Sizes of BSs

In this subsection, we investigate the effects of the proposed framework with these baseline schemes in terms of the total latency, cache hit ratio, handover latency and percentage of traffic offload versus different cache sizes (percentage of the total content size) of each BS (as shown in Fig. 7).

Fig. 7(a) shows that the proposed framework has superior performance in reducing the total latency, especially when the cache ratio is 25%, it can achieve up to 8.8%, 9.5%, 10.2%, and 11.4% improvements compared with MAB, MARL, JCC-UA, CCCP schemes, respectively. From Fig. 7(b), the proposed framework can support around 11.4%, 21.2%, 30.5% and 41.4% of the total content requests at the cache ratio of 10.0%, 20.0%, 30.0% and 40.0%, respectively, which means that the proposed framework has more advantages on optimizing content caching in dynamic multi-access scenarios. Fig. 7(c) evaluates the performance in terms of the handover latency. We can easily observe that the proposed framework has an excellent performance in reducing handover delay, especially when the cache ratio is 30%. This is because the proposed matching theory-based lazy re-association strategy can reduce the number of re-association through the lazy re-association mechanism and the swap and update matching operation to a certain extent. Fig. 7(d) shows the proposed framework can offload the percentage of traffic of the considered MEC network by 6.03% to 50.1% at different cache ratio levels from 5% to 50%, which outperforms the other baseline schemes.

F. Effects of Different Numbers of MDs

In this subsection, we investigate the impacts of different numbers of the MDs on the above four performance metrics in Fig. 8. In particular, the percentage for cache size of each BS is fixed as 25% of the total content size, and we conduct the experiments in the range of [40, 400] MDs, respectively.

Fig. 8(a) shows that the proposed framework can reduce more total latency compared with the other schemes. Specifically, as the number of MDs increases, the proposed scheme is more effective in reducing the total latency and can achieve up to 8.0%, 6.6%, 7.2%, 5.3% improvements compared with the JCC-UA strategy, 8.9%, 8.9%, 10.3% and 10.5% improvements

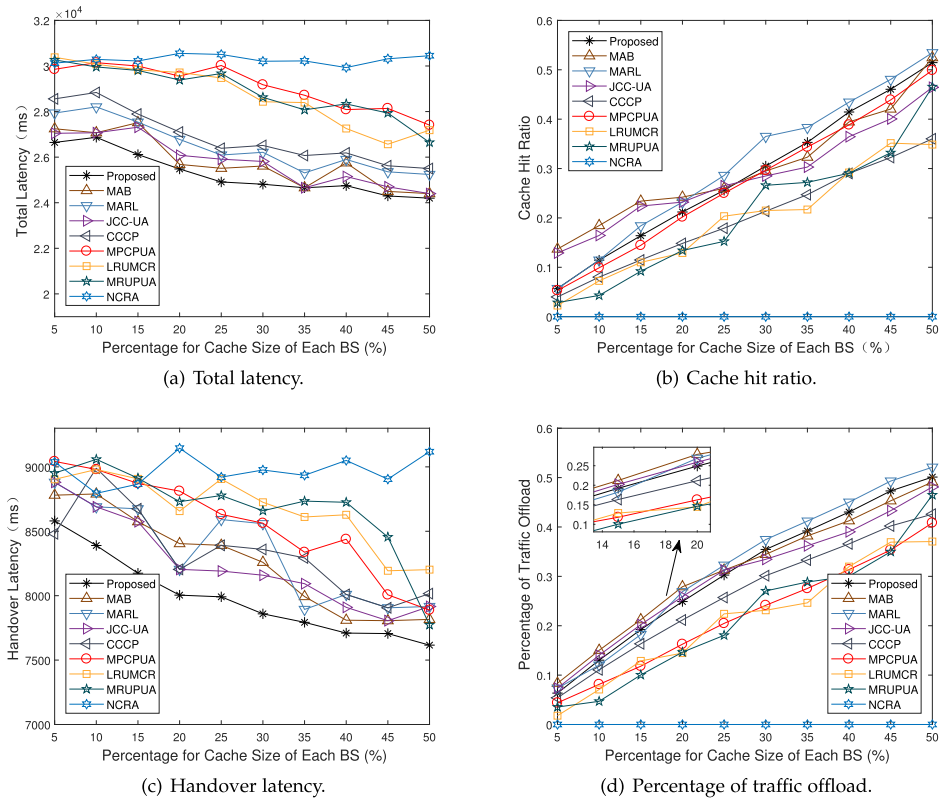


Fig. 7. Total latency, cache hit ratio, handover latency and percentage of traffic offload versus different cache sizes.

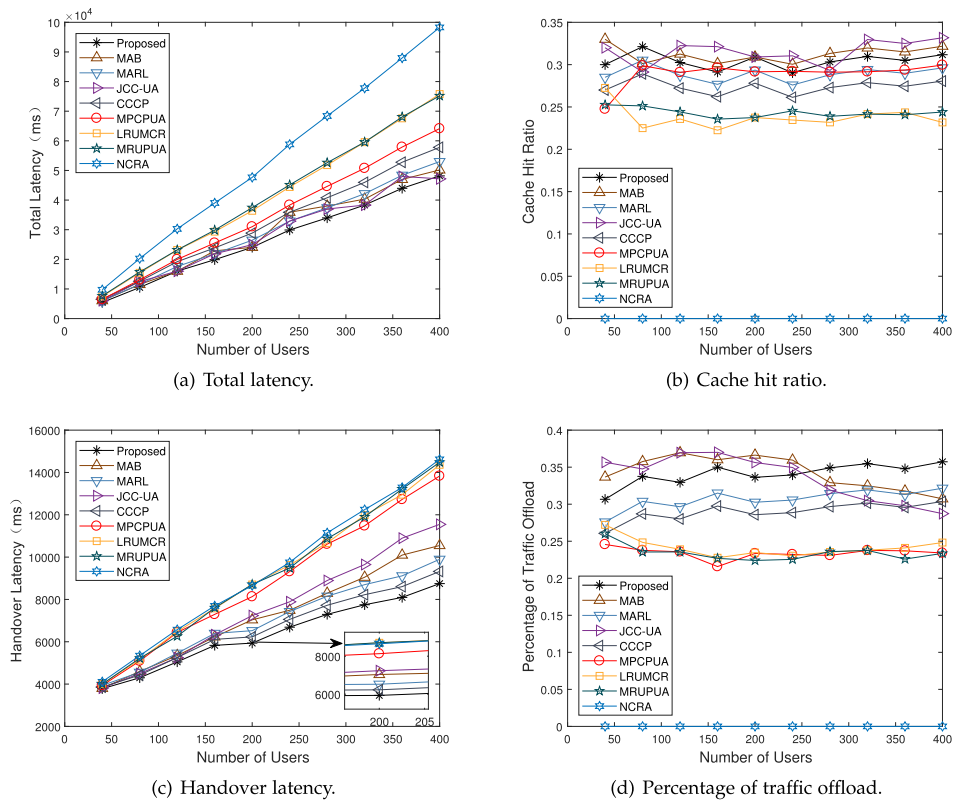


Fig. 8. Total latency, cache hit ratio, handover latency and percentage of traffic offload versus different number of MDs.

compared with the MARL strategy at the MDs of 240, 280, 360 and 400, respectively. Fig. 8(b) proves that the proposed framework can better satisfy content requests. In particular, the cache hit ratio fluctuates to remain within a small range and it can support around 30.0%, 30.2%, 30.9%, 30.3% and 30.5% of the total content requests at the MDs of 40, 120, 200, 280 and 360. From Fig. 8(c), we can find that its trend is similar to Fig. 8(a), and it can achieve up to 37.4%, 48.5%, 44.3% and 44.0% improvements compared with LRUMCR strategy, 38.7%, 48.4%, 45.0% and 46.3% improvements compared with NCPA strategy at the MDs of 160, 200, 240 and 280, respectively. Fig. 8(d) shows that the proposed framework has obvious advantages in terms of offloading the traffic. It can be observed that the overall offloading percentage of the proposed framework, JCC-UA and MAB remains around 0.34, while the other baseline schemes remain at the level of 0.24. To explain it, as the number of MDs increases, the lazy re-association strategy can swap the MDs and satisfy the content requests by the associated BSs rather than the CS. It proves that the proposed framework can perform better when facing a high density of content requests.

VI. CONCLUSION

In this article, we have investigated the issue of joint content caching and user association with high user mobility to minimize the weighted sum of content acquisition latency and handover latency in smart cities. Specifically, we have presented a three-layer MEC network architecture for practically offloading the traffic of duplicated contents downloading and associated the MDs with appropriate BSs. To address the formulated mixed time-scale non-convex BILP optimization problem, we have proposed a weighted LFM based intelligent content caching framework by introducing a weight matrix to associate different latent factors for learning intelligent long-time scale content caching policy at each time frame. Then, we design a short-time scale matching theory-based lazy re-association strategy for determining user association policy at each time slot. Simulation results on the real-world MEC environment have demonstrated the effectiveness of the proposed framework on reducing content acquisition latency and handover latency.

REFERENCES

- [1] H. Li et al., "Mobility-aware content caching and user association for ultra-dense mobile edge computing networks," in *Proc. IEEE Glob. Commun. Conf.*, 2020, pp. 1–6.
- [2] F. Wang, F. Wang, J. Liu, R. Shea, and L. Sun, "Intelligent video caching at network edge: A multi-agent deep reinforcement learning approach," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 2499–2508.
- [3] Y. Zhang, Y. Li, R. Wang, J. Lu, X. Ma, and M. Qiu, "PSAC: Proactive sequence-aware content caching via deep learning at the network edge," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2145–2154, Oct.–Dec. 2020.
- [4] Z. Hu, F. Zeng, Z. Xiao, B. Fu, H. Jiang, and H. Chen, "Computation efficiency maximization and QoE-provisioning in UAV-enabled MEC communication systems," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1630–1645, Apr.–Jun. 2021.
- [5] Z. Ming, X. Li, C. Sun, Q. Fan, X. Wang, and V. C. Leung, "Dependency-aware hybrid task offloading in mobile edge computing networks," in *Proc. IEEE 27th Int. Conf. Parallel Distrib. Syst.*, 2021, pp. 225–232.
- [6] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Fourthquarter 2017.
- [7] Z. Chang, L. Lei, Z. Zhou, S. Mao, and T. Ristaniemi, "Learn to cache: Machine learning for network edge caching in the Big Data era," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 28–35, Jun. 2018.
- [8] S. Chen, Z. Yao, X. Jiang, J. Yang, and L. Hanzo, "Multi-agent deep reinforcement learning-based cooperative edge caching for ultra-dense next-generation networks," *IEEE Trans. Commun.*, vol. 69, no. 4, pp. 2441–2456, Apr. 2021.
- [9] X. Ge, S. Tu, G. Mao, C.-X. Wang, and T. Han, "5G ultra-dense cellular networks," *IEEE Wireless Commun.*, vol. 23, no. 1, pp. 72–79, Feb. 2016.
- [10] X. Li, P. Wu, X. Wang, K. Li, Z. Han, and V. C. M. Leung, "Collaborative hierarchical caching in cloud radio access networks," in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2017, pp. 462–467.
- [11] D. Liu, B. Chen, C. Yang, and A. F. Molisch, "Caching at the wireless edge: Design aspects, challenges, and future directions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 22–28, Sep. 2016.
- [12] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Cooperative content caching in 5G networks with mobile edge computing," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 80–87, Jun. 2018.
- [13] J. Kwak, Y. Kim, L. B. Le, and S. Chong, "Hybrid content caching in 5G wireless networks: Cloud versus edge caching," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3030–3045, May 2018.
- [14] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. G. Andrews, "User association for load balancing in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 2706–2716, Jun. 2013.
- [15] D. Liu et al., "User association in 5G networks: A survey and an outlook," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1018–1044, Secondquarter 2016.
- [16] K. Hou, Q. Xu, X. Zhang, Y. Huang, and L. Yang, "User association and power allocation based on unsupervised graph model in ultra-dense network," in *Proc. IEEE Wireless. Commun. Netw. Conf.*, 2021, pp. 1–6.
- [17] M. Amine, A. Walid, A. Kobane, and J. Ben-Othman, "New user association scheme based on multi-objective optimization for 5G ultra-dense multi-RAT HetNet," in *Proc. IEEE Int. Conf. Commun.*, 2018, pp. 1–6.
- [18] W. Teng, M. Sheng, K. Guo, and Z. Qiu, "Content placement and user association for delay minimization in small cell networks," *IEEE Trans. Veh. Tech.*, vol. 68, no. 10, pp. 10201–10215, Oct. 2019.
- [19] Y. Cui, F. Lai, S. Hanly, and P. Whiting, "Optimal caching and user association in cache-enabled heterogeneous wireless networks," in *Proc. IEEE Glob. Commun. Conf.*, 2016, pp. 1–6.
- [20] T. D. Tran, T. D. Hoang, and L. B. Le, "Caching for heterogeneous small-cell networks with bandwidth allocation and caching-aware BS association," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 49–52, Feb. 2018.
- [21] L. Li et al., "Deep reinforcement learning approaches for content caching in cache-enabled D2D networks," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 544–557, Jan. 2020.
- [22] Z. Zhang, H. Chen, M. Hua, C. Li, Y. Huang, and L. Yang, "Double coded caching in ultra dense networks: Caching and multicast scheduling via deep reinforcement learning," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 1071–1086, Feb. 2020.
- [23] Z. Zhang and M. Tao, "Deep learning for wireless coded caching with unknown and time-variant content popularity," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1152–1163, Feb. 2021.
- [24] W. Jiang, G. Feng, S. Qin, and Y.-C. Liang, "Learning-based cooperative content caching policy for mobile edge computing," in *Proc. IEEE Int. Conf. Commun.*, 2019, pp. 1–6.
- [25] X. Wu, Q. Li, X. Li, V. C. M. Leung, and P. C. Ching, "Joint long-term cache allocation and short-term content delivery in green cloud small cell networks," in *Proc. IEEE Int. Conf. Commun.*, 2019, pp. 1–6.
- [26] W. Teng, M. Sheng, X. Chu, K. Guo, J. Wen, and Z. Qiu, "Joint optimization of base station activation and user association in ultra dense networks under traffic uncertainty," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 6079–6092, Sep. 2021.
- [27] Z. Cheng, M. LiWang, N. Chen, H. Lin, Z. Gao, and L. Huang, "Learning-based joint user-ap association and resource allocation in ultra dense network," in *Proc. IEEE 91st Veh. Technol. Conf.*, 2020, pp. 1–5.
- [28] D. Li, H. Zhang, K. Long, W. Huangfu, J. Dong, and A. Nallanathan, "User association and power allocation based on Q-learning in ultra dense heterogeneous networks," in *Proc. IEEE Glob. Commun. Conf.*, 2019, pp. 1–5.
- [29] H. Zhang, S. Huang, C. Jiang, K. Long, V. C. M. Leung, and H. V. Poor, "Energy efficient user association and power allocation in millimeter-wave-based ultra dense networks with energy harvesting base stations," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 1936–1947, Sep. 2017.

- [30] C. Dai, K. Zhu, R. Wang, and B. Chen, "Contextual multi-armed bandit for cache-aware decoupled multiple association in UDNs: A deep learning approach," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 4, pp. 1046–1059, Dec. 2019.
- [31] Y. Li, H. Ma, L. Wang, S. Mao, and G. Wang, "Optimized content caching and user association for edge computing in densely deployed heterogeneous networks," *IEEE Trans. Mobile Comput.*, vol. 21, no. 6, pp. 2130–2142, Jun. 2022.
- [32] X. Yang, Z. Fei, B. Li, J. Zheng, and J. Guo, "Joint user association and edge caching in multi-antenna small-cell networks," *IEEE Trans. Commun.*, vol. 70, no. 6, pp. 3774–3787, Jun. 2022.
- [33] M. F. Pervej, L. T. Tan, and R. Q. Hu, "User preference learning-aided collaborative edge caching for small cell networks," in *Proc. IEEE Glob. Commun. Conf.*, 2020, pp. 1–6.
- [34] Y. Jiang, M. Ma, M. Bennis, F.-C. Zheng, and X. You, "User preference learning-based edge caching for fog radio access network," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1268–1283, Feb. 2019.
- [35] D. Wu, M. Shang, X. Luo, and Z. Wang, "An L_1 -and- L_2 -norm-oriented latent factor model for recommender systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 10, pp. 5775–5788, Oct. 2022.
- [36] D. Wu, X. Luo, M. Shang, Y. He, G. Wang, and M. Zhou, "A deep latent factor model for high-dimensional and sparse matrices in recommender systems," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 51, no. 7, pp. 4285–4296, Jul. 2021.
- [37] Q. He et al., "A game-theoretical approach for user allocation in edge computing environment," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 3, pp. 515–529, Mar. 2020.
- [38] P. Lai et al., "Optimal edge user allocation in edge computing with variable sized vector bin packing," in *Proc. Int. Conf. Service-Oriented Comput.*, 2018, pp. 230–245.
- [39] B. Serbetci and J. Goseling, "On optimal geographical caching in heterogeneous cellular networks," in *Proc. IEEE Wireless. Commun. Netw. Conf.*, 2017, pp. 1–6.
- [40] X. Li, X. Wang, P.-J. Wan, Z. Han, and V. C. M. Leung, "Hierarchical edge caching in device-to-device aided mobile networks: Modeling, optimization, and design," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 8, pp. 1768–1785, Aug. 2018.



Chuan Sun (Student Member, IEEE) received the B.S. degree from the Wuhan University of Science and Technology, Wuhan, China, in 2017. He is currently working toward the Ph.D. degree with the School of Big Data & Software Engineering, Chongqing University, Chongqing, China. His current research interests include mobile edge computing and caching, reinforcement learning, and federated learning.



Fang Fang (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of British Columbia, Vancouver, BC, Canada, in 2017. She is currently an Assistant Professor with the Department of Electrical and Computer Engineering and the Department of Computer Science, Western University, London, ON, Canada. Prior to joining Western, she was an Assistant Professor with the Department of Engineering at Durham University, Durham, U.K., from 2020 to 2022. From 2018 to 2020, she was a Research Associate with the

Department of Electrical and Electronic Engineering, The University of Manchester, Manchester, U.K. Her research interests include machine learning for intelligent wireless communications, non-orthogonal multiple access (NOMA), reconfigurable intelligent surface (RIS), multi-access edge computing (MEC), and edge AI. Dr. Fang has been the General Chair for EAI GameNets 2022 and the Symposium Chair for IEEE Globecom, 2023. She received the Exemplary Reviewer Certificates of the IEEE Transactions on Communications in 2017 and 2021. She is currently an Associate Editor for IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY.



Hui Li (Student Member, IEEE) received the B.S. degree from Northeast Agricultural University, Harbin, China, in 2019. He is currently working toward the Ph.D. degree with the School of Big Data & Software Engineering, Chongqing University, Chongqing, China. His current research interests include edge computing and caching, and edge intelligence.



Xiuhua Li (Member, IEEE) received the B.S. degree and M.S. degree from the Harbin Institute of Technology, Harbin, China, in 2011 and 2013, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada, in 2018. He is currently a tenure-track Assistant Professor with the School of Big Data & Software Engineering, Chongqing University, Chongqing, China, and also a member of the Haihe Laboratory of ITAI. He is the Head of the Institute of Intelligent Software and

Services Computing associated with the Key Laboratory of Dependable Service Computing in Cyber Physical Society, Chongqing University, Education Ministry. He has authored or coauthored more than 90 technical papers in IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, ICC, and GLOBECOM. His research interests include edge computing and caching, and edge intelligence.



Qilin Fan (Member, IEEE) received the B.E. degree from the College of Software Engineering, Sichuan University, Chengdu, China, in 2011, and the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2017. She is currently an Associate Professor with the School of Big Data and Software Engineering, Chongqing University, Chongqing, China. Her research interests include network optimization, mobile edge computing and caching, network virtualization and machine learning.



Xiaofei Wang (Senior Member, IEEE) received the B.S. degree from the Huazhong University of Science and Technology, Wuhan, China, and the M.S. and Ph.D. degrees from Seoul National University, Seoul, South Korea. He was a Postdoctoral Fellow with The University of British Columbia, Vancouver, BC, Canada, from 2014 to 2016. He is currently a Professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. He has authored or coauthored more than 150 technical papers in IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE INTERNET OF THINGS JOURNAL, IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, IEEE TRANSACTIONS ON MULTIMEDIA, INFOCOM, and ICDCS. His research interests include edge computing, edge intelligence, and edge systems. In 2017, he was the recipient of the IEEE ComSoc Fred W. Ellersick Prize, and in 2022, IEEE ComSoc Asia-Pacific Outstanding Paper Award.



Victor C. M. Leung (Life Fellow, IEEE) is currently a Distinguished Professor of computer science and software engineering with Shenzhen University, Shenzhen, China. He is also an Emeritus Professor of electrical and computer engineering and the Director of the Laboratory for Wireless Networks and Mobile Systems, University of British Columbia (UBC), Vancouver, BC, Canada. He has authored or coauthored widely in his research field, which include wireless networks and mobile systems. Dr. Leung is serving on the editorial boards of IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, IEEE ACCESS, and several other journals. He was the recipient of the 1977 APEBC Gold Medal, during 1977–1981 NSERC Postgraduate Scholarships, IEEE Vancouver Section Centennial Award, 2011 UBC Killam Research Prize, 2017 Canadian Award for Telecommunications Research, 2018 IEEE TCGCC Distinguished Technical Achievement Recognition Award, and 2018 ACM MSWiM Reginald Fessenden Award. He co-authored papers that was the recipient of the 2017 IEEE ComSoc Fred W. Ellersick Prize, 2017 IEEE Systems Journal Best Paper Award, 2018 IEEE CSIM Best Journal Paper Award, and 2019 IEEE TCGCC Best Journal Paper Award. He is a Fellow of the Royal Society of Canada (Academy of Science), Canadian Academy of Engineering, and Engineering Institute of Canada. He is named in the current Clarivate Analytics list of Highly Cited Researchers.