1. Introduction

With the rapid growth in the number of smart devices and social multimedia applications, the requests of users for content files (e.g., videos, photos, and music) have increased significantly. It causes an unprecedented explosion of data [1, 2] and will pose a considerable challenge to the architecture of next-generation mobile communication systems. According to a report by Cisco, in 2023, the total number of Internet users will reach 5.3 billion (66% of the global population), with 3.6 network devices per capita and 29.3 billion network devices that will be connected to the network [3]. However, in traditional cellular networks, when users initiate content requests, the content needs to be distributed from the servers of the remote content provider (RCP) to users via mobile communication networks. When the number of users increases steeply, the demands of users for remote content will also rise dramatically, leading to a rapid increase in the number of requests processed by the remote servers, which results in a severe processing load. Therefore, effective caching architecture and methods need to be designed to handle the vast traffic requests and relieve the pressure in the cellular network.

It has been found that most mobile data traffic is generated by a large number of repeated downloads of a limited amount of popular contents by users [4]. Users with similar geographic locations, identity backgrounds, and interests have a higher probability of asking for the same data. However, the same content requested by different users must be repeatedly transmitted through the entire mobile cellular networks, causing network congestion and resource waste. Therefore, the concept of edge caching was proposed to meet the massive demands for content distribution in mobile cellular networks [5, 6]. With edge caching technology, the frequently requested content can be cached closer to the user side, such as macro base station (MBS), small base stations (SBS), edge server, and mobile devices [7]. As a result, edge caching can effectively enhance network performance by offloading network traffic, reducing total system delivery costs, and lowering content delivery latency for terminal users. In addition, device-to-device (D2D) communication technology allows direct communication between devices within a certain distance, supporting data to be shared directly among users without passing through the BSs (MBS and SBS). Thus, D2D communications in cellular networks can increase network throughput and improve spectrum utilization, reducing backhaul costs and transmission delays [1, 8]. Moreover, with the developments of hard disk expansion technologies and personal devices, such as smartphones, larger cache space can be obtained at lower costs [9]. Consequently, it is essential to research how to combine the above technologies to further improve the caching efficiency of the cellular networks.
Nowadays, researchers have achieved some research results for content caching in edge networks. Content caching and bandwidth allocation in an MBS-associated heterogeneous cell network were studied in [10]. In the above literature, users can associate with the MBS with excellent channel conditions and cache the requested contents in advance in the related MBS. Chen et al. [11] formulated the collaborative inter-SBS caching as a multi-intelligence decision problem based on a partially observable Markov decision process, jointly optimizing the delivery costs required to fetch contents from local SBS, nearby SBS, and remote servers. Asheralieva et al. [12] modeled MBS and contents access in each cluster as a queuing system, with cluster heads coordinating D2D connections between users for contents sharing. Then the author developed a distributed contents access and distribution algorithm under this framework. Li et al. [13] considered multiple relay stations randomly and evenly distributed within the coverage area of the central MBS for caching contents files. However, these studies only focus on single-level caches in MBS, SBS, mobile devices, or relay nodes, ignoring the collaboration between layers and lacking the full utilization of the edge network resources.

In practical business scenarios, the probability of requesting the same contents among different users varies due to diverse characteristic attributes (e.g., gender, age, and occupation). Research shows that users initiate content requests based on their individual preferences. Therefore, the user preferences for content have a significant impact on the performance of the caching system [14]. In [15], a personal preference probability model was applied to design a caching strategy, which proved that the use of personal preference character could significantly improve the performance of the D2D network with cache assistance. Liu et al. [16] combined personal preferences with recommender systems. The predicted ratings of users for all videos were obtained from historical rating information, as the probability of user requested for contents to improve the cache hit ratio. The probabilistic latent semantic analysis (PLSA) model was used in [17] to model user request behavior, analyze user preferences, and design caching policies to improve cache offloading performance. It can be seen that taking into account the individual preference information can effectively improve caching performance.

Selecting popular contents for caching is a widely adopted content caching rule [18]. Caching popular contents with equal probability for different users was adopted in [19] to improve the cache hit ratio. A better compromise between energy efficiency and delivery latency was achieved by caching uncompressed or compressed popular video blocks to the edge network in [20]. Mou et al. [21] proposed a content popularity prediction model based on long short-term memory (LSTM) network, considering the characteristics of content popularity over time and caching popular contents based on the prediction results to improve the cache hit ratio. Garg et al. [22] built two online predictive content popularity models. After optimizing the content popularity distribution, content caching was performed to obtain the maximum probability of successful content delivery. In [23], a clustering-based popularity prediction framework was constructed to improve the accuracy of prediction. The content was then cached in the edge server based on the predicted popularity. However, both [19, 20] calculated the popularity of content based on Zipf distribution without considering that it evolved dynamically over time. In [21–23], popularity evolving characteristics were considered, and a prediction model was designed for analysis. Nevertheless, they ignore that the size of the areas covered by MBS, SBS, and content servers varies, as does the number of users and subscribers. This can make the regional popularity of the content vary from the global popularity. Moreover, while existing research has studied several caching strategies based on user preferences or content popularity, it lacks the comprehensive consideration of both in the edge networks, resulting in limited caching performance.

Targeting the aforementioned issues, in this paper, a diversified demands integration based proactive caching strategy towards edge networks is proposed for integrating users’ individual and group demands and considering the global and regional popularity of the contents, so as to enhance the caching efficiency. The contributions of this paper are summarized as follows:

1. An across-layers collaborative caching architecture towards edge networks is designed to make full use of the limited cache space and reduce cache redundancy. Compared with the traditional single-layer caching models, the proposed architecture considers individual requirements at the user layer and addresses the group demands and the redundancy of cached contents at the SBS and MBS layers, respectively. After determining the contents to be cached, the cache placement method is designed to cache the contents to suitable positions.

2. To improve the performance of proactive caching, the individual preference and regional popularity estimation method are proposed to determine which contents should be cached. Specifically, different users may have diverse preferences for the same content due to various characteristic attributes and the variation of the users in the region is smaller than the contents. Therefore, we combined deep learning networks with content-based collaborative filtering algorithms to design a user preference model to accurately perceive users’ personalized demands. Then, we analyzed the differences between individual and group demands and formulate regional popularity as local preference popularity and local content popularity to balance these two different demands.

3. To place the content in the appropriate location, the differentiated preference-awareness caching approach is devised in conjunction with the presented architecture. The user preference content caching method (UP-CCM) is proposed for the user layer to
meet the personalized demands of users. The regional popularity content caching method (RP-CCM) in the SBS layer is presented to meet the overall demands of all users in the local region. The MBS layer caches popular contents that have not been cached by the SBS layer to reduce cache redundancy.

The remainder of this paper is organized as follows. Section 2 introduces a system model, a brief process for the work we propose, and the problem formulation. Section 3 discusses the individual preferences and regional popularity estimation method. Section 4 presents the edge caching approach with differentiated preference awareness. Simulation results and analyses are provided in Section 5, and a conclusion is given in Section 6.

2. System Model

2.1. Across-Layer Collaborative Caching Architecture. An illustration of the hierarchical edge caching architecture towards the edge network is shown in Figure 1, consisting of user, SBS, and MBS layers. All users, SBSs, and MBS are equipped with a local cache. We consider $N$ users in the user layer, denoted as a set $\mathbf{N} = \{u_1, u_2, \ldots, u_n, \ldots u_N\}$, and $u_i$ represent the $n$-th user. The SBS layer has $M$ SBSs, defined as the set $\mathbf{M} = \{b_1, b_2, \ldots, b_m, \ldots b_M\}$, where $b_m$ means the $m$-th SBS. $I_0$ is used to indicate the MBS. Mobile users carrying mobile devices are uniformly distributed in the service area. All the MBS, SBS, and the cache storage capacity of mobile devices in the network are limited, which are denoted by $S_0$, $(S_0^n)_{M \times 1}$, and $(S_0^n)_{N \times 1}$, respectively, $S_0 \gg S_0^n \gg S_0^M$. Suppose the users in the area are relatively fixed (such as the workplace), the areas covered by the SBS do not overlap, and the users can only be served by their belonging SBS. There is a library of $K$ content files, which is denoted as a set $\mathbf{F} = \{f_1, f_2, \ldots, f_k, \ldots f_K\}$, that all users may request in the system, where $f_k$ means the $k$-th content file and $s_{f_k}$ denotes its content size.

The user devices can operate as D2D transmitters or receivers, and the D2D connections can be established between user devices within collaborative communication distance $R_d$ for contents sharing. Assuming that user preferences and content popularity change relatively slowly, popular contents can be precached during off-peak hours (e.g., late at night) to reduce traffic load and avoid possible network traffic congestion [4, 17].

Compared with the traditional single-layer or intra-layer collaboration caching architecture, we design a proactive edge caching architecture with cross-layer collaboration. In the user layer, the contents are cached in users’ local devices based on individual preferences to meet the personalized demands of different users. In the SBS layer, the contents are cached according to regional popularity to meet the demands of the users group. In the MBS layer, to reduce cache redundancy, contents with high popularity and not cached in the SBS layer are cached, thus forming a collaborative edge caching architecture between layers.

2.2. Proactive Caching Strategy Process. The work process of the proposed proactive caching strategy for edge networks is shown in Figure 2. We give an overview here first.

2.2.1. SBS Control System. The SBS control system is primarily responsible for recording data on the request behavior of users in the local region. Then, individual preference perceptions of users and regional popularity estimates are performed. In addition, the SBS has location information for all users in the area and assists in establishing D2D communication between user devices. The following list is available at SBS:

1. User list: a list for recording all users that are connected to and managed by the SBS. Particularly, the list records each user’s ID, gender, age, location information, and so on.
2. D2D neighbors list: this list records the neighbors’ information of all users in the region, including their ID, location, and distance from each other.
3. User request list: this list records the historical request behavior of all users in the area, with the user’s ID, requested content ID, generated timestamp of the request, rating of the content, and so on.
4. User preference list: based on the results of individual preference perception, the preferences of all users in the region for each content are recorded in this list.
5. Content regional popularity list: this list records the popularity of each content based on regional popularity estimates.
2.2. MBS Control System. The MBS control system manages various resources, such as computing and communications resources, to determine how to provide services to mobile users. Specifically, the MBS needs to make caching decisions based on user preference data and regional popularity calculations and record the content cached at each layer. Then, the MBS manages and coordinates the content delivery process based on the generated cache records.

![Diagram of MBS Control System]

2.3. Problem Formulation. The caching policy in this paper is a deterministic policy, which means that the content is precached at the appropriate location as 1 and not cached as 0. The definitions are as follows:

\[ I_{b_m} = \left\{ c_{b_m}^1, \ldots, c_{b_m}^{l_{b_m}}, \ldots, c_{b_m}^{u_{b_m}} \right\}, \quad \forall b_m \in M, \quad (1) \]

\[ I_{u_n} = \left\{ c_{u_n}^1, \ldots, c_{u_n}^{l_{u_n}}, \ldots, c_{u_n}^{u_{u_n}} \right\}, \quad \forall u_n \in N. \quad (2) \]

Next, in order to formulate the problem of content caching, we define the following 0–1 variables for the given caching policy C, by

\[ c_k^\varphi = \begin{cases} 1, & c_k \in C_\varphi, \\ 0, & \text{otherwise} \end{cases}, \quad \varphi \in \{ L_0, b_m, u_n \}, \quad (4) \]

where when \( c_k^\varphi = 1 \), the content \( f_k \) is precached on the MBS, SBS, or the corresponding user device. When \( c_k^\varphi = 0 \), the content is not cached. The \( C_{L_0}, C_{b_m}, \) and \( C_{u_n} \) denote the MBS, SBS, and the users’ local device cache, respectively. In addition, \( C_{L_0} \subseteq I_{L_0}, C_{b_m} \subseteq I_{b_m}, \) and \( C_{u_n} \subseteq I_{u_n}. \) The caching policy denoted by \( C = \{ C_{L_0}, C_{b_m}, C_{u_n} \} \) must satisfy the following storage capacity constraints:

\[ |C_{L_0}| = \sum_{k=1}^{K} c_k^{L_0} s_{f_k} \leq S_0, \quad (5) \]

\[ |C_{b_m}| = \sum_{k=1}^{K} c_k^{b_m} s_{f_k} \leq S_b, \quad \forall b_m \in M, \quad (6) \]

where equation (5) indicates that the sum of the sizes of all contents cached in the MBS is less than or equal to its cache space, and similarly, equations (6) and (7) indicate the storage capacity constraints in the SBS and user devices.

The cache hit ratio and total system delivery costs are important policy evaluation metrics in edge caching [1, 7]. The cache hit ratio is the probability that a user’s request will be hit on the edge network. The total system delivery costs represent the total delivery costs required to satisfy all users’ requests. When user \( u_n \) initiates a content request, the request can be fulfilled at the user’s local device, a neighboring user within D2D communication distance, the SBS, the MBS, or the RCP, as the contents may be cached in a different location. Defining a request matrix \( Z = (f_{k,u_n})_{N \times K} \) and \( f_{k,u_n} \) indicates whether the user \( u_n \) requests a cache behavior for content \( f_k \) expressed by

\[ f_{k,u_n} = \begin{cases} 1, & \text{requested} \\ 0, & \text{otherwise} \end{cases} \quad (8) \]

We do not consider the problem of transmission errors during file delivery. The files requested by the user \( u_n \) are denoted \( f_{k,u_n}. \) The delivery costs of obtaining a unit of content under different conditions can be defined as follows:

\[ W_{u_n} = \begin{cases} w_0, & f_{k,u_n} \in C_{u_n} \\ w_1, & f_{k,u_n} \notin C_{u_n} \cup f_{k,u_n} \in C_{b_m} \cup f_{k,u_n} \in C_{L_0} \\ w_2, & f_{k,u_n} \notin (C_{u_n} \cup C_{b_m} \cup C_{L_0}) \cap f_{k,u_n} \in C_{b_m} \\ w_3, & f_{k,u_n} \notin (C_{u_n} \cup C_{b_m} \cup C_{L_0}) \cap f_{k,u_n} \notin C_{b_m} \\ w_4, & f_{k,u_n} \notin C \end{cases} \quad (9) \]

where \( w_0 < w_1 < w_2 < w_3 < w_4 \) denotes the delivery costs required to transmit each unit of content, which is the combined price needed after considering the logical distance, the wireless access costs, and the transmission energy consumption [1, 24, 25]. The \( C_{b_m} \) denotes the neighboring
user cache within the communication distance $R_d$ of the user $u_n$.

Therefore, the cache hit ratio of each layer and the delivery costs of users to obtain contents from different conditions can be calculated separately in the following cases:

1. When the user content requests are fulfilled at the local device, the local device cache hit ratio, and the required delivery costs are calculated as follows:

$$\text{Hit}_{\text{self}} = \frac{\sum_{n=1}^{N} \sum_{k=1}^{K} f_{k,u_n} \psi_{k}^{u_n}}{\sum_{n=1}^{N} \sum_{k=1}^{K} f_{k,u_n}},$$

$$h_{\text{self}} = \sum_{k=1}^{K} \sum_{n=1}^{N} f_{k,u_n} \psi_{k}^{u_n} s_{f_k},$$

(10) (11)

(2) When the local device does not cache the users requested contents, and it can be satisfied at neighboring users within D2D communication distance $R_d$, given by

$$\text{Hit}_{\text{d2d}} = \frac{\sum_{n=1}^{N} \sum_{k=1}^{K} f_{k,u_n} \psi_{k}^{u_n,R_d}}{\sum_{n=1}^{N} \sum_{k=1}^{K} f_{k,u_n}},$$

$$h_{\text{d2d}} = \sum_{k=1}^{K} \sum_{n=1}^{N} f_{k,u_n} \psi_{k}^{u_n,R_d} w_1 s_{f_k},$$

(12) (13)

where $\psi_{k}^{u_n,R_d}$ denotes whether the $k$-th content is cached at a neighboring user within communication distance $R_d$ of the user $u_n$, denoted as follows:

$$\psi_{k}^{u_n,R_d} = \begin{cases} 1, & \psi_{k}^{u_n,R_d} \in \psi_{R_d}, \\ 0, & \text{otherwise}. \end{cases}$$

(14)

From equations (10) and (12), the cache hit ratio at the user layer can be obtained as follows:

$$\text{Hit}_{\text{user}} = \text{Hit}_{\text{self}} + \text{Hit}_{\text{d2d}} = \frac{\sum_{n=1}^{N} \sum_{k=1}^{K} f_{k,u_n} \psi_{k}^{u_n,R_d}}{\sum_{n=1}^{N} \sum_{k=1}^{K} f_{k,u_n}} + \frac{\sum_{n=1}^{N} \sum_{k=1}^{K} f_{k,u_n} \psi_{k}^{u_n,R_d}}{\sum_{n=1}^{N} \sum_{k=1}^{K} f_{k,u_n}}$$

(15)

3. Individual Preferences and Regional Popularity Estimation Method

Traditional caching strategies often use Zipf distribution to generate the popularity of the contents and the probability of users’ requests, which means that all users are assumed to have the same preference for one content. This ignores the difference in demand among users. In contrast to the traditional model, the caching strategy in this paper analyses users’ request preferences and content popularity from actual business data features, which is more in line with real application scenarios.

3.1. Perception of Individual Preferences. In actual application scenarios, users usually request content according to their individual preferences. The higher the degree of choice for a specific content, the higher the probability of asking it [15]. Based on [17], we define individual preference as the probability distribution of requesting all contents. Then, we design an individual preference perception model based on a deep learning model to analyze users’ demands. Let $q_{u_n} = [q_{f_1,u_n}, \ldots, q_{f_{K},u_n}]$ means the request probability distribution of the $n$-th user, where $q_{f_{i},u_n}$ denotes the probability that the $n$-th user requests the $i$-th file, and $\sum_{i=1}^{K} q_{f_{i},u_n} = 1$. We use a matrix $Q = (q_{f_{i},u_n})_{N \times K}$ to represent the preferences of all users. The design of the model and the analysis of individual preferences are as follows.
3.1.1. Modeling Individual Preference. In this paper, we consider network scenarios where the users in the region are more fixed (e.g., workplace). Therefore, the variation of the users in the region is smaller than the contents. So, based on users’ historical rating information, we combine condition restricted Boltzmann machine (CRBM) and content-based collaborative filtering algorithms to design an individual preference perception model, defined as I-CRBM, to predict unrated content ratings. The users’ rating of the content is used as the users’ level of preference. And the higher the preference, the higher the probability of a request.

The CRBM is based on the restricted Boltzmann machine with the addition of rated and unrated information \( r \in \{0, 1\}^K \) (0 for rated and 1 for unrated). It is commonly used in the field of recommender systems. And it learns users’ implicit information by reconstructing their rating data and then gets users predicted ratings for unrated items. This enables personalized content recommendations. Therefore, CRBM is suitable for user personalized demand analysis.

In our model, each softmax neuron in the visible layer corresponds to a users’ content rating. Suppose there are a total of \( N \) users and \( r \) of them rated the contents, then \( r \) softmax neurons and \( N - r \) missing neurons are used to construct the visible layer of I-CRBM. Each content corresponds to one CRBM, and the connection weights and node biases are shared among the CRBMs. The I-CRBM network structure is shown in Figure 3.

3.1.2. Analysis for the Individual Preference. Define a rating matrix \( G \) to represent the historical rating information of users. Train the I-CRBM network with the historical data, then iteratively update the model parameters, and predict the ratings given by the user \( u_n \) to content \( f_k \). From this, we can obtain the reconstructed users’ rating of all contents. The predicted rating \( \hat{x}_{n,k} \) is defined as the users’ preference for the contents. The higher value of \( \hat{x}_{n,k} \), the higher the probability of request in the next period.

Considering that users usually do not re-request contents that have already been requested [26, 27]. Therefore, based on the historical rating records, reset the \( \hat{x}_{n,k} \) of these contents to 0 and obtain \( x_{n,k} \) as follows:

\[
x_{n,k} = \begin{cases} 
\hat{x}_{n,k}, & \text{no requested}, \\
0, & \text{requested}.
\end{cases}
\]  

(22)

According to \( x_{n,k} \), the request probability of the \( n \)-th user for the \( k \)-th content can be obtained, which is expressed as the ratio of user \( u_n \)’s preference for the content \( f_k \) to the sum of preference for all contents, calculated as follows:

\[
q_{f_k, u_n} = \frac{x_{n,k}}{\sum_{k=1}^{K} x_{n,k}}
\]  

(23)

3.2. Estimates of Content Regional Popularity. In cellular networks, the number of users in a single cell is usually less than that in the area covered by the content server. The content popularity depends on all users who send content requests in the region. Hence, the regional popularity of content in a cell is different from the global popularity observed at the content server. So, content caching based on regional popularity in the SBS of a cell can better meet the demands of groups of users in the local region.

Comprehensive consideration of users’ individual and group demands, we model the regional popularity of content as LPP and LCP. The LPP is analyzed based on users’ preference perceptions. The LCP predictions are made based on LPP data from historical periods. The LPP and LCP of content \( f_k \) are denoted by \( p_{lp,f_k} \) and \( p_{lc,f_k} \), respectively.

3.2.1. Local Preference Popularity. The LPP is analyzed based on user characteristics, taking into account their preferences and activeness characteristics, and reflects the collective request preferences of all users in the local region over time. As a demand statistic for multiple users, it is different from the individual demand. This means that there is a variance between individual and group demand.

Studies have shown that only a few active users are actively creating traffic [1, 26]. Users with high activity levels will request more contents, which will generate a lot of traffic and contribute considerably to preference popularity. Definition \( a_{u_n} \) denotes the users’ activity level, reflecting the frequency of user requests for contents over a while. The LPP of the \( k \)-th content is measured as follows:

\[
p_{lp,f_k} = \sum_{n=1}^{N} a_{u_n} q_{f_k, u_n}
\]  

(24)

And, \( a_{u_n} \) can be derived from the following equation:

\[
a_{u_n} = \frac{\sum_{k=1}^{K} f_{k, u_n}}{K}
\]  

(25)

3.2.2. Local Content Popularity. The LCP is analyzed without considering the difference in preference and activity among users. It focuses on the frequency of contents being requested.
in the local region over a while and can be obtained based on predictions from LCP data over historical periods. Define the LCP of content \( f_k \) to be \( p^t_{lcp,f_k} \) in a time slot \( t \) and \( p^{t+1}_{lcp,f_k} \) in a time slot \( t + 1 \), \( p^t_{lcp,f_k} \approx p^{t+1}_{lcp,f_k} \). The closer the predicted LCP is to the actual value at the time \( t + 1 \), the more accurate it is to cache the contents that the user will request. This improves the performance of the caching policy.

(1) Local Content Popularity Model. According to the existing research, content popularity data has a feature of temporal evolution [21–23]. In addition, recurrent neural networks (RNN) have temporal memory. LSTM, which is a special form of RNN, is even more capable of learning long-distance dependencies, making it very suitable for processing time-series data. In subsequent work, the researcher made improvements based on the LSTM and proposed the gated recurrent units (GRU) [28]. The GRU optimizes the three gate structures in the LSTM network into two gates. This makes it possible to reduce the training parameters and speed up the training while maintaining the LSTM prediction effect. Therefore, we designed the LCP prediction model based on the GRU. The structure of the GRU is schematically shown in Figure 4.

The connection relationship in Figure 4 is given by

\[
\begin{align*}
    r_t &= \sigma(W_{rx}x_t + U_rh_{t-1}) , \\
    z_t &= \sigma(W_{xz}x_t + U_zh_{t-1}) , \\
    h_t &= \tanh(W_{hx}x_t + U_x(r_t \odot h_{t-1})) , \\
    h_t &= (1 - z_t)h_{t-1} + z_tr_t ,
\end{align*}
\]

where \( x \) is input vector, \( h \) is output vector, \( \tilde{h} \) is candidate output, and \( \sigma \) is a sigmoid function.

In our work, we use a single hidden layer GRU network with four neurons to predict the corresponding LCP of content \( f_k \) jointly.

(2) Analysis for the Local Content Popularity. Definition \( p^{t}_{lcp,f_k} \) denotes the LCP of content \( f_k \) for the period \( t \). It is the total number of requests for content \( f_k \) by all users in this period, as a percentage of the sum of requests for all contents by all users, calculated as follows:

\[
p^{t}_{lcp,f_k} = \frac{\sum_{n=1}^{N} f_{k,n}}{\sum_{n=1}^{N} \sum_{k=1}^{K} f_{k,n}}
\]

In our model, the historical LCP data for content \( f_k \) are used as input data, and the output data are predicted \( p^{t+1}_{lcp,f_k} \) for the next period.

Therefore, from the above analysis, the regional popularity of content \( f_k \) can be obtained as follows:

\[
p_{f_k} = p_{lcp,f_k} + p_{lcp,f_k} = \sum_{n=1}^{N} a_{n,q_{f_k,n}} + p_{lcp,f_k}.
\]

4. Edge Caching Approach with Differentiated Preference Awareness

Determining the cache location, selecting the cache contents, and designing the caching method are the three core issues of caching strategy [29]. The first two issues have been analyzed in Sections 2 and 3 of this paper, and the caching method is proposed for the last issue in this section. In the content placement phase, the caching method is designed in the user layer based on preference perception; the caching approach in the SBS layer is designed based on regional popularity, which integrates the differences in requirements between individual users and groups. The MBS layer, on the other hand, focuses on reducing cache redundancy. In the content delivery phase, users initiate requests to obtain the required content based on their preferences.

4.1. Proactive Content Placement. Content placement in the edge cache determines how much total traffic the supply can offload [29]. The proper content placement allows users to efficiently hit their requested content at the edge, eliminating the need to download from the RCP via a backhaul link, thus reducing the total system delivery costs.

4.1.1. Content Placement Method. We designed a content placement method for each layer based on individual preferences and regional popularity estimates in the across-layer collaborative caching architecture.

(1) Content Placement Method in the User Layer. Based on the analysis of user preferences, the probability distribution of each user’s requests for all contents can be obtained, leading to the proposed user preference content caching method (UP-CCM) at the user level. By sensing user preferences, we can determine the content to be cached. Then, the user’s most interesting contents are cached to its local device, and the cache space limits the amount of cached contents. By caching in this way, the individual demands of users can be better satisfied as shown in Algorithm 1.

(2) Content Placement Method in the SBS Layer. Based on the LPP calculation and LCP prediction, the regional popularity of the contents in the SBS coverage area can be obtained, leading to the proposed regional popular content caching method (RP-CCM) for the SBS layer. By estimating the regional popularity of the content, we determine the
contents to be cached in the SBS layer. Then cache the contents with higher regional popularity to SBS under the constraint of cache space. With this caching method, it is possible to consider the individual demands while meeting the user groups’ demands in the local region. It is shown in Algorithm 2.

### (3) Content Placement Method in the MBS Layer

At the MBS layer, popular contents not cached by the SBS layer are cached based on regional popularity. By collaborative caching in this way, the cache redundancy is better reduced, and the limited cache space is fully utilized.

#### 4.1.2. Content Placement Process

Our proactive content placement process is shown in Figure 1 and expressed as follows:

1. **Step 1:** the SBS learns user preferences and content region popularity from historical request records, determines the contents to be cached in each layer in conjunction with content placement methods, and reports to MBS.
2. **Step 2:** the MBS first obtains all the required contents for each layer from the RCP via a backhaul link and then caches the contents for this layer according to its content placement method.
3. **Step 3:** the MBS sends the required contents to the SBS by request record, and then the SBS caches the related contents by its content placement method.
4. **Step 4:** the SBS sends the contents to be cached by each user, and the user caches the contents according to the user-layer content placement method.

### 4.2. Edge Content Delivery

Effective content delivery method allows users to access the requested content quickly and efficiently, which reduces the total system delivery costs [30].

#### 4.2.1. D2D Sharing Model

When a user requests content from a neighbor within communication distance $R_d$, he may find that his requested content is stored at multiple neighbors. In this case, the neighboring user with the highest sharing probability needs to be selected to establish a D2D communication connection for content sharing. We model D2D content sharing among users in terms of physical relations (geographic distance) and preference aspects (preference similarity) [1, 31], respectively.

Definition $r_{n,v}^D$ denotes the geographical distance between user $u_n$ and user $u_v$. The $r_{n,v}^D$ is obtained by the following Euclidean distance formula:

$$r_{n,v}^D = \sqrt{(X_n - X_v)^2 + (Y_n - Y_v)^2}. \quad (29)$$

Define the preference similarity between user $u_n$ and user $u_v$ as $\zeta_{n,v}$. When $\zeta_{n,v}$ is larger, it means that the preferences of user $u_n$ and user $u_v$ are more similar, as shown in the following equation:

$$\zeta_{n,v} = \frac{\sum_{f_k \in \mathcal{F}} q_{f_k,u_n} q_{f_k,u_v}}{\sqrt{\sum_{f_k \in \mathcal{F}} q_{f_k,u_n}^2 \sum_{f_k \in \mathcal{F}} q_{f_k,u_v}^2}}, \quad \forall n \in [1, N], \forall v \in [1, N]. \quad (30)$$

We combine geographic distance and preference similarity to obtain the sharing probability $C_{n,v}^D$ between user $u_n$ and user $u_v$ as follows:

$$C_{n,v}^D = \frac{1}{r_{n,v}^D} \zeta_{n,v}, \quad \forall n \in [1, N], \forall v \in [1, N]. \quad (31)$$

It is known that the probability of content sharing is higher when the geographical distance $r_{n,v}^D$ between user $u_n$ and user $u_v$ is smaller, or the preference similarity $\zeta_{n,v}$ is larger.

#### 4.2.2. Content Delivery Process

Our edge content delivery process is shown in Figure 1 and expressed as follows:

1. **Step 1:** the user initiates a content request based on preferences, and if the request is available in the local device cache, the content is retrieved directly with zero latency. If not, the content request is sent to the SBS in the region to which it belongs.
2. **Step 2:** the SBS first checks whether a user that is adjacent to the requesting user has cached the requested content. If so, the request is notified to the

---

![Algorithm 1](image)

**Algorithm 1:** User Preference Content Caching Method (UP-CCM).

**Input:** $G$; Initialize: Caching matrix $C_u = (0)^{N \times S_u}$

**Output:** Caching matrix $C_u$

1. use I-CRBM to predict $\tilde{x}_{u,i}$
2. reset $\tilde{x}_{u,i}$ by (22) to get $x_{u,i}$
3. compute $Q = (q_{f_k,u_n})^{N \times K}$ by (23)
4. for $n = 1, 2, \ldots, N$ do
5. sort $q_{u,i}$ from largest to smallest to get $q_{u,i} = [q_{f_1,u_i}, \ldots, q_{f_k,u_i}, \ldots, q_{f_L,u_i}]$
6. for $i = 0$ to $S_u^n$ do
7. $C_u|_{u,i} = q_{u,t,i}$
8. end for
9. end for
neighboring user with the highest probability of sharing. The SBS then helps it establish a D2D connection to share the content. Otherwise, SBS will check whether the content has been cached in the SBS. If the content can be found at the SBS, it is sent to the requesting user over the cellular link. If the request is still not fulfilled, the MBS will be requested. In addition, the SBS records the request behavior of each user to analyze individual preferences and region popularity for the next period.

Step 3: the MBS first checks if the content is cached in its space and, if it is, sends it to the user over the cellular link.

Step 4: the content is not cached at the MBS. It is first downloaded from the RCP via the backhaul link and then sent to the requesting user via the cellular link.

5. Simulation Results and Analysis

We use python to simulate a network scenario to evaluate the performance of the proposed caching policy. The Movielens 100K data set is used [16, 17], which contains 100,000 rating records for 1,682 movies by 943 users, where the rating data contains user IDs, movie IDs, ratings, and timestamps. Considering that user requests for content have a temporal order, the rating records of each user are sorted by timestamp. The top 80% of rating records are divided into a training set for the user preferences and content region popularity analysis. The bottom 20% are divided into a test set as actual content requests from users for caching policy verification.

5.1. Parameter Setting. We use the percentage reduced costs (PRC) [1] and edge cache hit ratio as the policy performance metrics. We not only compare the proposed caching policy under different scenarios but also compare it with two baseline schemes, which are the equal probability random caching (EPRC) policy [19] and most popular caching (MPC) policy [13], to reflect the performance of the proposed caching strategies.

The PRC is given by

$$PRC = \frac{|h - \hat{h}|}{h} \times 100\%,$$

(32)

where $\hat{h}$ denotes the total system delivery costs needed when all requests are satisfied directly through RCP.

This paper defines the cache hit ratio as the edge cache hit ratio, as shown in the following equation:

$$Hit = Hit_{user} + Hit_{d2d} + Hit_{mbs}.$$  

(33)

Assuming that the users are uniformly and randomly distributed in the cell, and each user has an equal size of cache space, $S_u = S_m$. Considering each content of the same size and all equal to 1, that is, $s_i = s_{f_i} = 1$. According to the delivery costs parameters set in [24, 25] for obtaining unit content under different conditions, we similarly penalize the content download from the RCP with \([w_0, w_1, w_2, w_3, w_4]) = [0, 1, 10, 100, 1000]$. In particular, the overheads we set here can be adjusted to suit the actual application. In addition, we have D2D communication distance $R_d = 50$, MBS coverage radius $R_{MBS} = 1000$, and SBS coverage radius $R_{SBS} = 150$. We analyze the relationship between cache space size, number of users, and PRC and Hit for each tier to verify the performance of the caching strategy proposed in this paper.

5.2. Analysis of Results

5.2.1. Training of the Model. The root-mean-square error (RMSE) was chosen as an evaluation indicator for the predictive accuracy of the I-CRBM in the individual preference model, and the formula is defined as follows:
\[ \text{RMSE} = \sqrt{\frac{\sum_{u,f} (\hat{r}_{uf} - r_{uf})^2}{|Z|}} \]  

(34)

where the denominator \(|Z|\) represents the size of the data set, the numerator \(\hat{r}_{uf}\) represents the ratings predicted by the model, and \(r_{uf}\) represents the actual ratings of the users in the data set.

In Figure 5, we show the training error of the model gradually converges with the increase in the number of iterations. The model’s prediction accuracy gradually improves with the rise of the number of neurons \(n_h\) in the hidden layer within a specific range, indicating that the model learns more features hidden in the data set, and the prediction effect is relatively better. However, after the number of neurons exceeds a certain number, the model extracts too many features, and the generalization ability decreases. And too many neurons cause the activated units to become more sparse, making the model prediction accuracy slip. Therefore, we choose to use \(n_h = 200\) for rating prediction for user preference analysis.

As shown in Figure 6, the LCP prediction model shows a fluctuating decreasing trend of training loss and faster convergence as the number of training times increases. GRU has one less gate function and a relatively more minor number of parameters compared to LSTM. Thus, the overall training speed is faster.

### 5.2.2. Impact of Individual Preference.

The impact of individual preferences on the caching performance of the proposed policy is analyzed. The UP-CCM at the user layer is used as a variable, and the caching scheme is analyzed for both cases with and without UP-CCM, as shown in Figure 7.

Figure 7(a) illustrates the relationship between PRC and \(S_b^m\) for the proposed cache policy with and without considering UP-CCM at the user layer. Setting the number of users (i.e., the number of devices) \(N = 50\) and having \(S_0 = 600\). It can be found that as the user device cache space (denoted by SD in the figure) increases, the PRC grows as well. Due to the increase in cache space, it allows more contents to be cached in the edge networks. When SD = 20, the scheme with UP-CCM upgrading the PRC by 7.1% on average over the plan considering only the caching method in the BSs layer, and the PRC reaches 61.1% at \(S_0^m = 300\). The reason is that UP-CCM-based can cache the content that users are most interested in locally on the device and share the content through D2D communication, minimizing the content delivery costs. The scheme without UP-CCM does not change due to SD, as only the content cache of the BS is considered at this point and not affected by changes at the user layer. Figure 7(b) shows the relationship between PRC and \(N\). Setting \(S_b^m = 200, S_0 = 600\). It can be found that when there is no user device for content caching (i.e., \(N = 0\), the PRC for each scenario is 50.9%, as it is not affected by UP-CCM at this point. The scheme with UP-CCM has 6.9% higher PRC on average than without UP-CCM at SD = 20, and the PRC reaches 68.1% when \(N = 100\) because there are more devices and space for content caching. Also, the PRC increases more rapidly as \(N\) rises. This is because the higher

### 5.2.3. Impact of Regional Popularity.

The regional popularity is modeled as LPP and LCP as described in Section 3. Since the effect of individual preferences has been verified in the previous subsection, this section focuses on the impact of LCP on the performance of the proposed caching policy, as shown in Figure 8. The performance of the scheme is also simulated for the two cases where the users’ requests are obtained through local region BSs and adjacent region BSs,
to reflect the impact of caching policy based on region popularity, as shown in Figure 9.

Figure 8(a) represents the relationship between PRC and $S^m_b$ for the proposed caching strategy in both cases of considering LCP and not considering LCP when modeling the regional popularity. Setting $N = 50$, $S_0 = 600$. It can be found that the PRC of each scheme increases with the enlarges of $N$ versus SD. And when SD = 20, the PRC of the scheme with LCP increases by 18.1% on average compared to that of the plan without LCP assumed. Figure 8(b) shows
the relationship between PRC and \( N \). Set \( S_b^m = 200, S_0 = 600 \). When \( N = 0 \) and \( S \leq [5, 10, 20] \), the PRC of the scheme with LCP increases by 19.5% compared to the plan without LCP. When \( N = 100 \), the improvement is 18.1%, 16.1%, and 13.9%, respectively. It can be seen that as \( N \) and SD increase, the scheme without LCP adds faster, which is because \( N \) and SD increase; more user-requested contents can be delivered through the D2D sharing. At this point, the LCP considered in the BSs layer has a gradually decreasing impact on the whole system’s performance. When \( SD = 20 \), the PRC of the scheme with LCP improves by 17.5% on average compared to without consideration. It can be seen that considering

![Figure 9: Relationship between PRC and \( S_b^m \) when obtained content from the local area and adjacent area.](image)

![Figure 10: Performance comparison of different caching strategies: (a) relationship between Hit and \( S_b^m \) (\( N = 50 \)) and (b) relationship between PRC and \( N \) (\( S_b^m = 200 \)).](image)
LCP to model regional popularity can considerably improve caching policy performance, and combining LPP and LCP can further satisfy user personalization and group demand. The relationship between PRC and $S^m_k$ where the users' request is obtained through local region BSs and adjacent area BSs is shown in Figure 9. Setting $N = 50$, $S_b = 600$. When $S^m_k = 0$, the PRC of locally obtained content is higher 28.8% than that of the adjacent area get and over 40.5% when $S^m_k = 300$, with an average improvement of 35.0%. In addition, as $S^m_k$ and SD increase, the PRC rise rate is greater when the content is acquired in the local region than in the neighboring areas. This is because the contents cached at the local user layer meet part of the users’ demands. However, the contents cached at the local SBS and MBS are selected based on regional popularity. At the same time, neighboring regions have various users, making the contents have different regional popularity, and the contents cached are more variable. Therefore, when contents are available through neighboring regions, more content must be re-downloaded from the RCP. It is thus clear that content caching based on regional popularity can remarkably improve the performance of the proposed caching policy.

5.2.4. Comparison of Different Caching Strategies. This section compares the performance of the proposed caching strategy with two baseline caching policies in Figure 10.

As illustrated in Figure 10(a), the cache hit ratio $Hit$ of the proposed caching strategy and the two baselines both increase as $S^m_k$ enlarge, and the proposed approach increases faster. This is because the proposed policy provides interlayer collaborative caching. The SBS and MBS caches are not duplicated, reducing cache redundancy. As a result, the limited cache space can be used more efficiently. In addition, caching content based on regional popularity can better meet the demands of groups of users in the local region. For EPRC, the effect of SD gradually decreases as $S^m_k$ increases because the user layer randomly caches restricted content. At this time, more content is obtained through the upper layer; the effect of the device layer gradually decreases; and $Hit$ gradually approaches. For MPC, at $SD = 5$ and $SD = 20$, the $Hit$ overlaps entirely, and as $S^m_k$ increases, the $Hit$ remains unchanged. This is because each layer caches the most popular contents and does not collaborate, leading to the contents cached by the user, and SBS layers can be found in the MBS layer. This makes SD and $S^m_k$ not affect on $Hit$. When $SD = 5$, the $Hit$ of the proposed policy increases over EPRC and MPC by 9.2% and 10.4% on average, respectively, and when $SD = 20$, by 10.3% and 14.5% on average, respectively. From Figure 10(b), we can see that the PRC of both the proposed and baseline policies increases as $N$ and SD rises. The proposed policy grows faster because the approach can cache more targeted contents of interest to users based on UP-CCM in the user layer. When $SD = 5$, the proposed policy improves the PRC over EPRC and MPC by 9.3% and 10.5% on average, respectively, and when $SD = 20$, by 11.2% and 15.6%, respectively. It can be seen that our proposed caching policy performs noticeably better than the two baseline caching policies.

6. Conclusion

In this paper, we propose a diversified demands integration based proactive caching strategy towards edge networks to increase the cache hit ratio and reduce the total system delivery costs. Based on the historical request information of users, we design the users’ preference perception model to analyze individual demands and decide which contents to be cached in the user layer to meet the individual demands of different users. In addition, a popularity prediction model based on deep learning networks is designed for local content popularity prediction. Furthermore, a content region popularity estimation method that combines user preferences and popularity analysis is proposed. We also design the SBS and MBS layers content caching methods to meet the group requirements of all users, utilize the limited cache space effectively, and reduce cache redundancy. The experiment results demonstrate that the proposed caching strategy can significantly improve the edge caching performance. In further work, we will analyze the impact of user mobility on cache performance.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 61901071, 61871062, 61771082, and U20A20157), Science and Natural Science Foundation of Chongqing, China (Grant no. cstc2020jcyj-zdxmX0024), Science and Technology Research Program of the Chongqing Municipal Education Commission (Grant no. KJQN202101324), University Innovation Research Group of Chongqing (Grant no. CXQT20017), and Program for Innovation Team Building at Institutions of Higher Education in Chongqing (Grant no. CXTDX201601020).

References


