Mobility-Aware and Preference-Driven Collaborative Edge Caching in SDN-enabled VLC Networks

Vuai Ali Kombo [*], **Adnan Ali** [**] , **Omar Kombo** [***], **Natalia Kryvinska** [****], **Mohammed Abdalsalam** [†],

[*] *School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan 430063, Hubei, PR China, (e-mail: vuaiali15@gmail.com)*
[**] *School of Computing, Communication and Media Studies, The State University of Zanzibar, 192 Tunguu Road, 72214 Central, Zanzibar, Tanzania, (e-mail: ali.adnan@suza.ac.tz)*
[***] *School of Computing, Communication and Media Studies, The State University of Zanzibar, 192 Tunguu Road, 72214 Central, Zanzibar, Tanzania, (e-mail: omar.kombo@suza.ac.tz)*
[****] *Department of Information Management and Business Systems, Head Faculty of Management, Comenius University Bratislava, Odbojárov 10, 82005 Bratislava 25, Slovakia, (e-mail: natalia.kryvinska@uniba.sk)*
[†] *Faculty of Electronics,Telecommunications and Informatics, Gdańsk University of Technology, Gudanski, Poland, (e-mail: mohammed.abdalsalam@pg.edu.pl)*

Abstract: The growing demand for low-latency content delivery in indoor environments has driven the need for efficient caching strategies in Mobile Edge Computing (MEC) systems. Visible Light Communication (VLC), combined with MEC, offers a promising solution for enhancing content delivery performance by utilizing high-speed, light-based communication. However, traditional caching strategies that rely solely on content popularity often fail to account for user mobility and preferences, which are critical in dynamic indoor settings. In this paper, a hybrid location-aware caching strategy is proposed, integrating user location prediction with content preference modeling to enhance cache efficiency in MEC-enabled VLC networks. Received Signal Strength (RSS)-based multilateration and Bidirectional Gated Recurrent Unit (BiGRU) models are implied to predict user movement patterns and Enhanced Collaborative Filtering (ECF) to anticipate content requests based on historical interactions. By combining these techniques, the system dynamically caches popular content at VLC-based Next Generation Node B small cells (VLC-gNB) near predicted user locations, improving cache hit ratios and reducing content retrieval latency. Additionally, a comprehensive system model is presented incorporating VLC and mmWave technologies, coordinated by Software-Defined Networking (SDN) technology, to manage content delivery and cache placement. Extensive simulation results demonstrate that the proposed caching strategy significantly improves cache hit ratios and reduces latency compared to traditional methods. This study highlights the importance of incorporating both mobility and user preferences in designing efficient caching mechanisms for next-generation MEC-enabled networks.

Keywords: Mobile Edge Computing, SDN, Edge Caching, Visible Light Communication, BiGRU, Collaborative Filtering.

1. INTRODUCTION

Due to the rapidly increasing use of mobile data, there is a growing need for efficient low latency and content delivery, especially inside buildings such as shopping centers, hospitals and office buildings. Mobile Edge Computing (MEC) is developed as an effective method to overcome these issues by moving computational resources and data closer to end users and reducing transport delay while improving the overall Quality of Experience (Hu et al. (2015)). Content caching is among the main enabling technologies in MEC, providing storage for storing frequently accessed content in close proximity of users and reducing the time needed to fetch data from the core network. On the other hand, user mobility and preferences are highly dynamic indoors, making the content caching process more complicated (Yao et al. (2019)).

In recent years, Visible Light Communication (VLC) networks have gained attention as a complementary technology for wireless communication (Chi et al. (2020)). The VLC utilizes low-cost light-emitting diodes (LEDs), providing high-speed, secure communication channels in indoor environments. The integration of VLC with MEC offers a unique opportunity for edge caching, where popular content can be cached at edge nodes near VLC Access Points (APs) to further reduce content delivery delays (Ali et al. (2021)). However, efficient content caching strategies remain challenging, particularly in dynamic environments where user preferences and mobility patterns change frequently.

One of the unique challenges in indoor environments is the difficulty in accurately predicting content popularity. Unlike edge caching in open areas, content popularity in indoor environments tends to be location-dependent, with mobile users in different public buildings exhibiting highly diverse preferences for downloadable content. For example, visitors in a museum likely require content related to exhibitions, while patients in a hospital are more interested in healthcare information. Furthermore, user preferences can vary significantly even within the same building. For example, patients in different departments may have distinct content needs in a hospital, just as customers in different mall sections may prefer different product-related content. This location-dependent nature of content demand is complex to capture using traditional edge caching schemes that rely on universal models across all locations.

Traditional caching strategies rely on content popularity predictions based on historical access patterns, often neglecting critical factors such as user mobility and preferences, particularly in indoor environments. To address these limitations, this paper introduces a hybrid caching strategy that integrates user location prediction and preference modelling within a software-defined VLC network environment. The system enhances location prediction accuracy by employing VLC Received Signal Strength (RSS)-based multilateration for accurate user positioning and Bidirectional Gated Recurrent Unit (BiGRU) models to capture temporal movement patterns. Additionally, Enhanced Collaborative Filtering (ECF) is utilized to forecast content requests, allowing for dynamic cache placement at the most appropriate Next Generation Node B small cells (VLC-gNB). This hybrid approach optimizes cache hit ratios, reduces latency, and adapts to changing signal conditions and user mobility, offering a more efficient solution for content delivery in edge computing environments.

The primary contributions of this paper are summarized as follows:

(1) A hybrid approach combining VLC RSS-based multilateration with BiGRU techniques to improve the user location prediction in the VLC network is introduced.
(2) A joint content caching strategy is proposed using ECF, integrating user location prediction with user preference modelling to optimize caching decisions in MEC-enabled VLC networks.
(3) A caching technique with location awareness is proposed, which is dynamically adapted to the mobility of users in the MEC-enabled VLC scenarios and utilizes predicted user positions in order to cache the content in advance.
(4) Simulations have been performed on a wide scale, and performance of the proposed caching strategy has been compared to classic caching approach, showing better results.

The rest of the paper is organized as follows: Section 2 reviews related work. Section 3 presents the system model, followed by Section 4, which details the proposed caching strategy based on user location and preference prediction. Section 5 provides the experimental setup and results, and Section 6 concludes the paper with directions for future research.

2. RELATED WORKS

Mobile edge caching has emerged as a vital solution to address the growing demand for high-quality, low-latency content delivery, particularly with the surge in data consumption from mobile applications and services. As mobile devices become more data-intensive, traditional network architectures struggle to deliver content efficiently. This limitation has led to the development of Mobile Edge Computing to enhance mobile networks, enabling local caching of content at edge nodes to increase network capacity, reduce latency, and enhance the Quality of Experience for users. MEC offers advantages over conventional techniques, including energy savings, improved throughput, enhanced privacy, and security (Zaman et al. (2021)).

The literature on content caching strategies has been extensively explored. For instance, (Sang et al. (2021)) proposed a cooperative caching mechanism for heterogeneous networks that focuses on minimizing delay using greedy cache placement and replacement techniques. (Yuan et al. (2021)) presented a cooperative service placement and request routing scheme in MEC, using a greedy strategy and heuristic routing to optimize service placement. Additionally, fuzzy caching mechanisms based on content distribution prediction were proposed by (Somesula et al. (2021)), incorporating echo state networks to manage deadlines and improve cache decisions. Furthermore, a Q-learning-based cooperative caching mechanism developed by (Chien et al. (2020)) aimed to minimize latency and reduce backhaul stress. However, these strategies often neglect user mobility and focus on static popularity models, limiting their effectiveness in dynamic environments.

Mobility-aware caching schemes have also been proposed to address these limitations. In (Li et al. (2020)), a mobility-aware online caching algorithm was introduced, combined with a lazy re-association algorithm to track user locations and update cached content dynamically. Similarly, (Yu et al. (2020)) proposed a proactive edge caching scheme using federated learning (MPCF), which collaboratively predicts content popularity and caches the most relevant content using a context-aware adversarial autoencoder. Another approach presented in (Wei et al. (2020)) addressed service caching by predicting user destinations using an idealized geometric model, facilitating content delivery to the base station to which the user is most likely to connect upon completing their movement. While these mobility-aware approaches improve caching

performance, they often neglect the personalized preferences of users, which play a significant role in determining content demand.

Many prior works have focused on caching strategies that account for user mobility, with some also considering user preferences. Some studies have explored the connection between user mobility and association with specific base stations. For instance, (Zaman et al. (2023)) addressed mobility issues through caching strategies designed to handle users on the move, particularly in video streaming scenarios. However, most existing methods store large amounts of content on edge nodes without considering the impact of user preferences and mobility jointly. As a result, these approaches can be less effective in real-world environments where user behaviour is more dynamic.

Moreover, current research often overlooks the complex relationship between user mobility, content popularity, and personal preferences, especially in indoor environments. While content popularity in outdoor networks tends to be universal across various regions, indoor environments such as hospitals, shopping malls, and other public buildings present distinct challenges. User preferences in these spaces are highly location-dependent and can vary significantly. For instance, patients in different hospital departments may have varying content needs, just as customers in different mall sections might be interested in different types of products. These variations are further complicated by the fact that users spend more than 90% of their time in such indoor environments (Cetin and Abo Aisha (2023)), with their entrance and departure influencing the local content demand.

Despite these challenges, research on indoor edge caching has been limited. While some works, such as (Li et al. (2016)) and (Ma et al. (2020)), have introduced novel indoor-outdoor caching relay systems to improve network throughput and enhance wireless resource utilization, they lack a comprehensive integration of user preferences and location-based caching decisions. Solutions such as the collaborative indoor edge caching framework proposed in (Shen et al. (2022), Kombo et al. (2025)), which enable personalized content popularity modelling, also need to adequately address the joint effect of user mobility and preferences on caching performance. This creates a research gap in existing content caching schemes that reduces their ability to meet user demands and optimize cache performance.

To address these challenges, this paper proposes a hybrid content caching strategy that integrates user location prediction with user preference modelling to optimize caching decisions in MEC-enabled VLC networks. This strategy improves the cache hit ratio and reduces response time, thereby enhancing the Quality of Experience for users in indoor environments.

3. SYSTEM MODEL

In this study, an indoor wireless network architecture designed to support proactive edge caching via hybrid VLC and mmWave communication is considered, as illustrated in Figure 1. The network comprises a Macro Base Station (MBS), a set of Remote Radio Light Heads

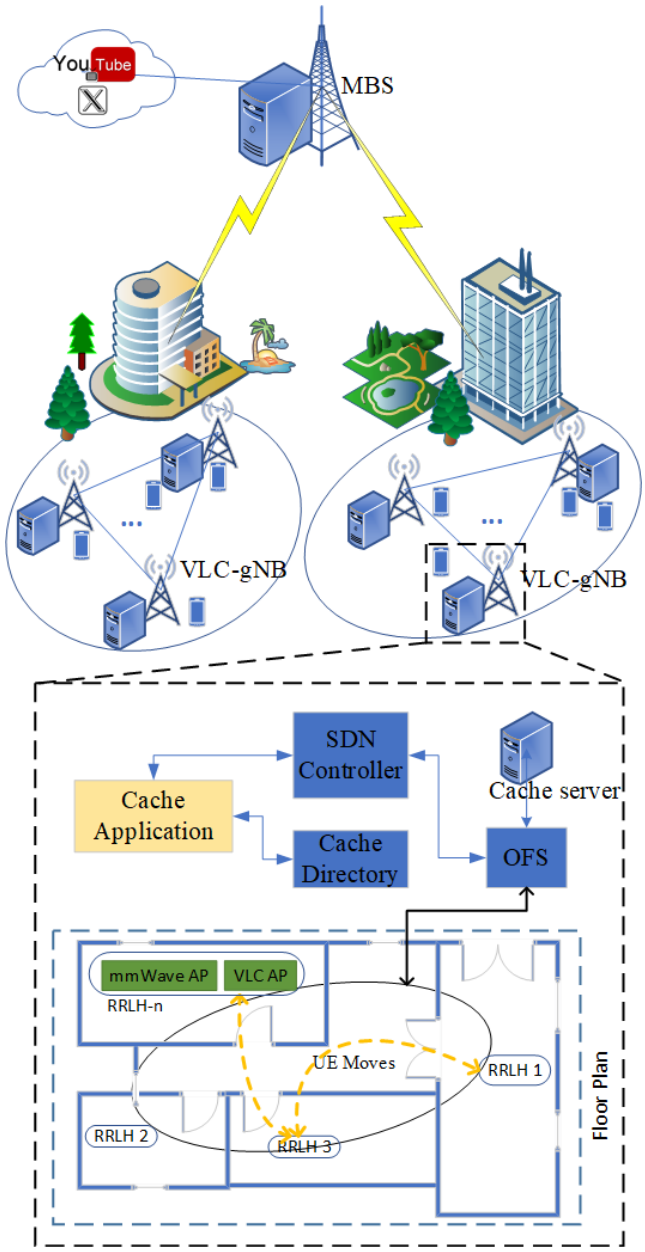

Fig. 1. System model

(RRLHs) mounted on the ceiling, and a set of cache-enabled VLC Next Generation Node Bs (VLC-gNBs) that serve as edge nodes. Each RRLH is equipped with both VLC and mmWave beam modules and functions as a high-capacity access point (AP) for confined areas. These APs are interconnected via an OpenFlow Switch (OFS), and managed by a centralized Software-Defined Networking (SDN) controller that coordinates data flow, cache content, and link selection policies. Users (UEs) communicate with the VLC-gNB using either VLC or mmWave links, depending on their location and the network load.

The environment is divided into L discrete indoor locations $\mathcal{L} = \{l_1, l_2, \ldots, l_L\}$, with a set of mobile users $\mathcal{U} = \{u_1, u_2, \ldots, u_U\}$ moving across them. Each user device communicates with a nearby VLC-gNB $b \in \mathcal{B} = \{b_1, b_2, \ldots, b_B\}$, depending on location and network conditions. The SDN controller assigns either VLC or mmWave

access to the user based on real-time link state and spatial proximity. The controller also oversees cache decisions and predicts user mobility and content demands to support edge-level prefetching.

The MBS maintains the global content repository $\mathcal{J} = \{v_1, v_2, \ldots, v_N\}$, where each item v_j has a size ϕ_j. The edge caches at VLC-gNBs have finite capacity C_b, and thus only a subset of content can be stored locally. When a user $u_i \in \mathcal{U}$ requests item v_j, the request is first forwarded to the associated gNB's cache. If the content is found locally, it is delivered immediately; otherwise, the request is forwarded to neighboring caches or routes the request to the MBS. Upon retrieval, the content is cached temporarily at the edge to reduce future delays.

To improve caching efficiency, the SDN controller continuously collects users' historical location traces and content request patterns. These data are used to predict each user's future location $\hat{l}_{u,t+1} \in \mathcal{L}$ using a BiGRU-based trajectory model, and their content interest $\hat{r}_{ij}$ using a hybrid collaborative filtering mechanism that accounts for both rating history and spatial proximity. The objective is to cache, at each VLC-gNB, the subset of content most likely to be requested by users predicted to enter its coverage zone. Table 1 explains the notations used in the equations and formulas.

Table 1. Notations and definitions

Symbol	Meaning
$\mathcal{L}$	Set of discrete locations
$\mathcal{U}$	Set of mobile users u_i
$\mathcal{B}$	Set of VLC-enabled gNBsb
$\mathcal{J}$	Set of video items v_j
$l_{u_i,t}$	Historical 2D location of user u_i at time t
$\hat{l}_{u_i,t+1}$	Predicted location of user u_i at time $t+1$
m_i	Order of Lambertian emission of an AP
$\theta_{1/2}$	Half-power angle of the lamp
ϕ_j	Video content size
I_i	Transmitted light intensity from the AP
A_r	Effective area of the receiver
z_i	Vertical distance between AP and UE
d_i	Distance from AP to the user
$r_{ij}, \hat{r}_{ij}$	Actual Predicted content score for user u_i on item v_j
$D_b(j)$	Predicted demand of content v_j at node b

4. CACHING STRATEGY BASED ON USER LOCATION AND PREFERENCE PREDICTIONS

Efficient content caching in MEC networks relies on accurately predicting user location and content preferences. In indoor VLC networks, where user mobility and content requests are highly dynamic, predicting user location can improve the accuracy of content popularity forecasts, thereby enhancing cache efficiency. By predicting a user's future location and combining this with user preference models, content popularity predictions become more targeted and caching strategies more effective.

The proposed caching strategy involves two key steps: predicting the user's location and forecasting content requests. BiGRU is used for user location prediction, and Enhanced Collaborative Filtering (ECF) predicts content requests based on past behavior, location, and preferences. These predictions guide the caching process, where high-demand content is stored at VLC-gNBs near predicted user locations to maximize cache hit ratios and reduce latency.

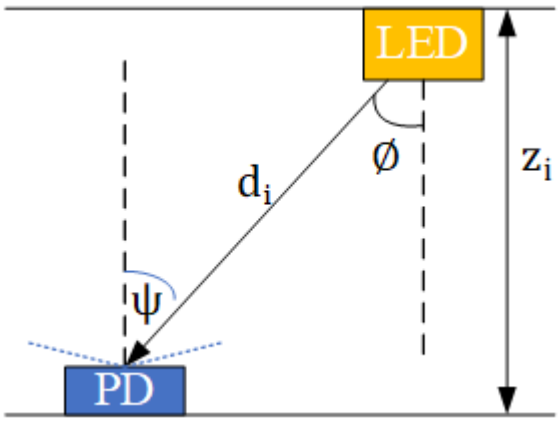

Fig. 2. VLC link configuration

The caching process involves:

- User Location Prediction: BiGRU forecasts the user's future location based on their movement history, helping decide which AP (VLC or mmWave) serves the user and which VLC-gNB should cache the relevant content.
- Content Request Prediction: ECF predicts the content a user is likely to request, guiding cache placement at VLC-gNBs to meet user demands.

4.1 User location prediction using BiGRU

In indoor VLC-enabled networks, user mobility significantly influences the caching performance, as users frequently change their point of attachment to access points. Proactive caching in such settings requires not only accurate knowledge of current user locations but also robust prediction of future positions to ensure content availability ahead of demand. This section presents a mathematical framework that combines real-time location estimation using VLC RSS signals with future location forecasting via a Bidirectional Gated Recurrent Unit (BiGRU) network.

The system first estimates the user's current location based on RSS from multiple VLC Access Points using multilateration technique. Each AP $i \in \mathcal{B}$ emits light with radiant intensity I_i and Lambertian order m_i, and the RSS received by user u from AP i at time t is modeled as:

$$\mathrm{RSS}_i = \frac{(m_i+1)A_i I_i \cos^{m_i}(\phi)\cos(\psi)}{2\pi d_i^2} \tag{1}$$

where A_i is the detector area, ϕ and ψ are the irradiance and incidence angles (equal in a perpendicular layout), and d_i is the distance between the user and AP i as shown in Figure 2. The Lambertian order m_i is computed from the half-power angle $\phi_{1/2}$ as:

$$m_i = \frac{-\log(2)}{\log(\cos(\phi_{1/2}))} \tag{2}$$

where $\phi_{1/2}$ is the semi-angle at half power of the light. Given a perpendicular layout, and vertical height z_i, this work assumes $\phi = \psi = \theta$ and $\cos(\theta) = \frac{z_i}{d_i}$. Substituting $\cos(\theta)$ into (1):

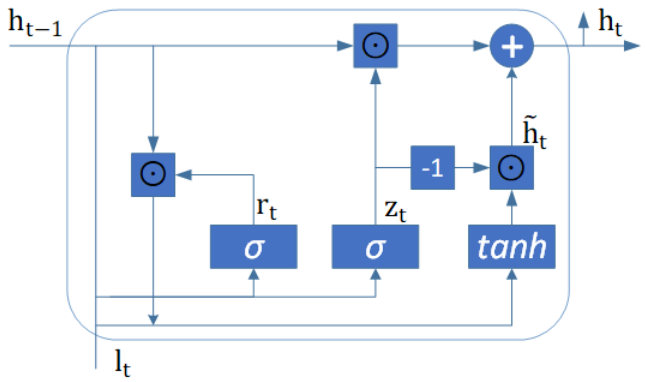

Fig. 3. GRU network structure diagram

$$\mathrm{RSS}_i = \frac{(m_i+1)A_i I_i \left(\frac{z_i}{d_i}\right)^{m_i}\left(\frac{z_i}{d_i}\right)}{2\pi d_i^2} = \frac{(m_i+1)A_i I_i z_i^{m_i+1}}{2\pi d_i^{m_i+3}} \tag{3}$$

Given the received RSS values, the horizontal distance is estimated as:

$$d_i = \left(\frac{(m_i+1)A_i I_i z_i^{m_i+1}}{2\pi \cdot \mathrm{RSS}_i}\right)^{\frac{1}{m_i+3}} \tag{4}$$

With known AP coordinates (x_i, y_i) and computed distances d_i, the user's current 2D location (x_t, y_t) is obtained by solving the nonlinear system:

$$\begin{cases} (x_t - x_1)^2 + (y_t - y_1)^2 &= d_1^2, \\ (x_t - x_2)^2 + (y_t - y_2)^2 &= d_2^2, \\ & \vdots \\ (x_t - x_n)^2 + (y_t - y_n)^2 &= d_n^2, \end{cases} \tag{5}$$

This system is solved using least-squares estimation, providing an accurate baseline location $l_{u,t} = (x_t, y_t)$. To forecast the user's future position, a temporal sequence of previous locations $X_{u,t} = [l_{u,t-N}, \ldots, l_{u,t}]$ is utilized, and model the next-step location $\hat{l}_{u,t+1}$ using a BiGRU architecture (Chung et al. (2014)). The GRU unit updates its hidden state at each time step using:

$$\begin{cases} r_t &= \sigma(W_r l_t + U_r h_{t-1} + b_r), \\ z_t &= \sigma(W_z l_t + U_z h_{t-1} + b_z), \\ \tilde{h}_t &= \tanh(W_h l_t + U_h(r_t \odot h_{t-1}) + b_h), \\ h_t &= z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t, \end{cases} \tag{6}$$

where l_t is the input at time t, h_{t-1} represents the previous hidden state, r_t and z_t are reset and update gates, $W_r, W_z, W_h, U_r, U_z, U_h$ are weight matrices. b_r, b_z, b_h represent biases. σ is the sigmoid activation function, and $\odot$ denotes element-wise multiplication. $\tilde{h}_t$ is the candidate state, and h_t represents the final hidden state as depicted in Figure 3. In the BiGRU, both forward and backward GRU states are computed over the sequence and concatenated as:

$$h_t = [\overrightarrow{h}_t; \overleftarrow{h}_t] \tag{7}$$

The concatenated hidden state is then projected to the predicted coordinates using a fully connected layer:

$$\hat{l}_{u,t+1} = W_o h_t + b_o. \tag{8}$$

with trainable parameters W_o and b_o. The model is trained to minimize the mean squared error between predicted and true locations:

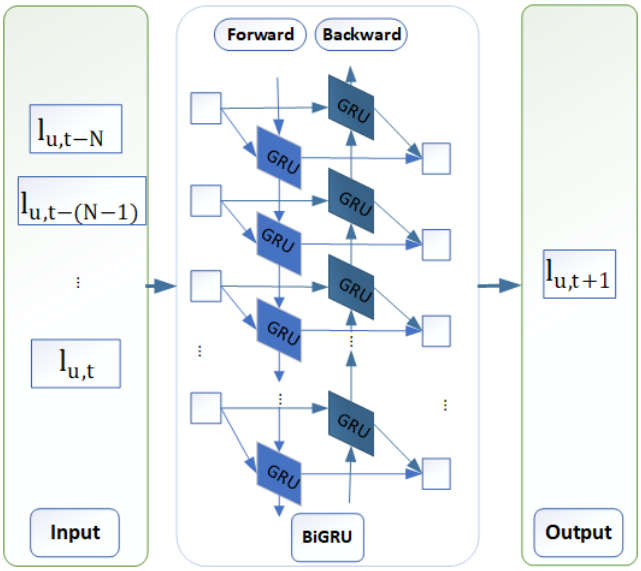

Fig. 4. Structure diagram of the BiGRU network

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{M}\sum_{i=1}^{M}\left\|\hat{l}_{u_i,t+1} - l_{u_i,t+1}\right\|^2 \tag{9}$$

This predictive module enables the system to anticipate future user positions and pre-position content accordingly at the nearest VLC-gNBs. The predicted location $\hat{l}_{u,t+1}$ serves as a key input to the content demand prediction and caching optimization processes described in the subsequent sections. Figure 4 shows a brief outline of the proposed BiGRU model architecture the pseudocode for the model is shown in Algorithm 1.

Algorithm 1 User location prediction algorithm using BiGRU

Require: Historical user locations $X_{u,t}$
Ensure: Predicted next user location (x_{t+1}, y_{t+1})
1: **for** each user u_j **do**
2: load and process the input data as input elements for the BiGRU model
3: split the dataset into training, validation and test sets
4: feed the data to the BiGRU model
5: using the model to forecast the next user location
6: **end for**
7: **Return:** Predicted next location (x_{t+1}, y_{t+1}).

4.2 ECF for Content Request Prediction

In addition to mobility prediction, accurate content request prediction is essential for optimizing content caching. The proposed Enhanced Collaborative Filtering (ECF) model integrates user location and user preferences to improve video content prediction accuracy in indoor VLC networks. By combining predicted user locations with user preferences, the model generates more relevant content recommendations, considering both past user behavior and spatial proximity. To enhance prediction accuracy, the model uses a hybrid similarity measure that combines two key factors, user behavior based on historical ratings and spatial proximity based on user location.

The ECF model predicts the likelihood of a user requesting specific content by incorporating location data into the user-item matrix, resulting in a User-Item-Location (A) matrix, as shown below:

$$\mathrm{A} = \begin{array}{c|cccc|c} & v_1 & v_2 & \dots & v_j & \text{Location} \\ \hline u_1 & r_{11} & r_{12} & \dots & r_{1j} & (x_1, y_1) \\ u_2 & r_{21} & r_{22} & \dots & r_{2j} & (x_2, y_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ u_m & r_{m1} & r_{m2} & \dots & r_{mj} & (x_m, y_m) \end{array} \tag{10}$$

where r_{ij} represents the rating given by user u_i to video v_j, the last column contains the user location (x_i, y_i), representing the user's geographical coordinates.

The global average rating μ is first computed to establish a baseline for content predictions. This is essential in estimating overall user satisfaction with videos:

$$\mu = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} r_{ij}}{|\{r_{ij} : r_{ij} \neq \text{null}\}|} \tag{11}$$

where r_{ij} is the rating from user u_i for video v_j, m is the number of users, and n is the number of videos. This baseline can then be refined using individual user biases and location-based factors. The model employs cosine similarity to compute user similarity based on ratings. The cosine similarity between users u_i and u_k is calculated as:

$$\text{sim}(u_i, u_k) = \frac{\sum_{j \in I_{ik}} r_{ij} \cdot r_{kj}}{\sqrt{\sum_{j \in I_{ik}} r_{ij}^2} \cdot \sqrt{\sum_{j \in I_{ik}} r_{kj}^2}} \tag{12}$$

where I_{ik} is the set of common videos rated by both users u_i and u_k, and r_{ij}, r_{kj} are the ratings given by users u_i and u_k, respectively.

In addition to rating similarity, the location-based similarity is computed using Euclidean distance between the locations of two users:

$$\text{loc\_sim}(l_i, l_k) = \frac{1}{1 + \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2}} \tag{13}$$

where (x_i, y_i) and (x_k, y_k) represent the locations of users u_i and u_k, respectively. This similarity measure ensures that users closer in proximity have a higher similarity score. The overall similarity between users u_i and u_k is a weighted combination of rating and location similarity:

$$\text{sim}(u_i, u_k, l_i) = \alpha \cdot \text{sim}(u_i, u_k) + (1-\alpha) \cdot \text{loc\_sim}(l_i, l_k) \tag{14}$$

where $\alpha \in [0, 1]$ is a weighting factor that balances the importance of rating similarity and location similarity.

Once the combined similarity is computed, the model employs two classical techniques to estimate the predicted score $\hat{r}_{ij}$. The first is the matrix factorization-based Singular Value Decomposition (SVD) (Koren et al. (2021)), which decomposes the rating matrix into latent user and item embeddings. The prediction is expressed as:

$$\hat{r}_{ij}^{\text{SVD}} = \mu + b_i + b_j + q_i^{\top} p_j \tag{15}$$

where μ is the global average rating (see (11)), b_i and b_j are user and item biases, and $q_i, p_j \in \mathbb{R}^k$ are k-dimensional latent feature vectors for user u_i and item v_j, respectively.

The second method uses a user-based k-nearest-neighbor (KNN) approach (Nguyen et al. (2023)). The rating prediction is computed as a weighted average over ratings from users similar to u_i:

$$\hat{r}_{ij}^{\text{KNN}} = \frac{\sum_{u_k \in \mathcal{U}_j} \text{sim}(u_i, u_k, l_i) \cdot r_{kj}}{\sum_{u_k \in \mathcal{U}_j} |\text{sim}(u_i, u_k, l_i)|} \tag{16}$$

where $\mathcal{U}_j \subseteq \mathcal{U}$ is the set of users who have rated item v_j and r_{kj} is the rating given by user u_k to video v_j.

The hybrid prediction combines both techniques, improving prediction accuracy:

$$\hat{r}_{ij} = \beta \cdot \hat{r}_{ij}^{\text{SVD}} + (1 - \beta) \cdot \hat{r}_{ij}^{\text{KNN}} \tag{17}$$

where $\hat{r}_{ij}$ is the final hybrid prediction, and β is a weighting factor. The hybrid approach takes advantage of both latent factor-based and user similarity-based predictions to provide more accurate and context-aware recommendations. To adapt dynamically to varying prediction reliability, β is adjusted based on the variance between the SVD and KNN predictions:

$$\beta = \min\left(1, \text{Var}(\hat{r}_{ij}^{\text{SVD}}, \hat{r}_{ij}^{\text{KNN}})\right) \tag{18}$$

This adaptive weighting ensures the model provides more reliable predictions when one model's output is more consistent with the data. The final prediction $\hat{r}_{ij}$ guides the caching decision, prioritizing content that is more likely to be requested by the user.

4.3 Content Caching Strategy

Given the user location forecasts and personalized content preference scores, the objective of the caching module is to determine which content items should be stored at each VLC-enabled gNB to maximize the cache hit rate under constrained storage capacity.

Let $q_{b,j} \in \{0, 1\}$ be a binary decision variable indicating whether content v_j is cached at node b. The predicted demand for item v_j at node b, denoted $D_b(j)$, is computed by aggregating the predicted ratings $\hat{r}_{ij}$ over users likely to be associated with b based on mobility prediction:

$$D_b(j) = \sum_{u_i \in \mathcal{U}_b} \hat{r}_{ij} \cdot p_{ij} \tag{19}$$

where $\mathcal{U}_b$ is the set of users within the coverage zone of node b, and p_{ij} is the likelihood of requesting item v_j which accounts for the user similarity based on ratings and location proximity, computed as:

$$p_{i,j} = \frac{\sum_{k \in U_j} \text{sim}(u_i, u_k, l_i)}{|U_j|} \tag{20}$$

where U_j is the set of users who have previously interacted with content j.

To complement the proactive caching strategy, a mathematical definition of cache hit rate that reflects predicted user behavior is introduced. Unlike conventional definitions, the proposed model incorporates both predicted user location (via BiGRU) and predicted preference score (via hybrid ECF), enabling more accurate evaluation of proactive caching decisions. The enhanced cache hit rate Hit_t^b is computed using weighted contributions from forecasted user-content interactions and location predictions. The cache hit rate at each VLC-gNB is defined as:

$$\text{Hit}_t^b = \frac{1}{|\mathcal{N}_t^b|} \sum_{n \in \mathcal{N}_t^b} D_b(j) . q_{b,j_n}, \tag{21}$$

where $\mathcal{N}_t^b$ is the set of content requests routed to node b during window t, and j_n is the index of the requested

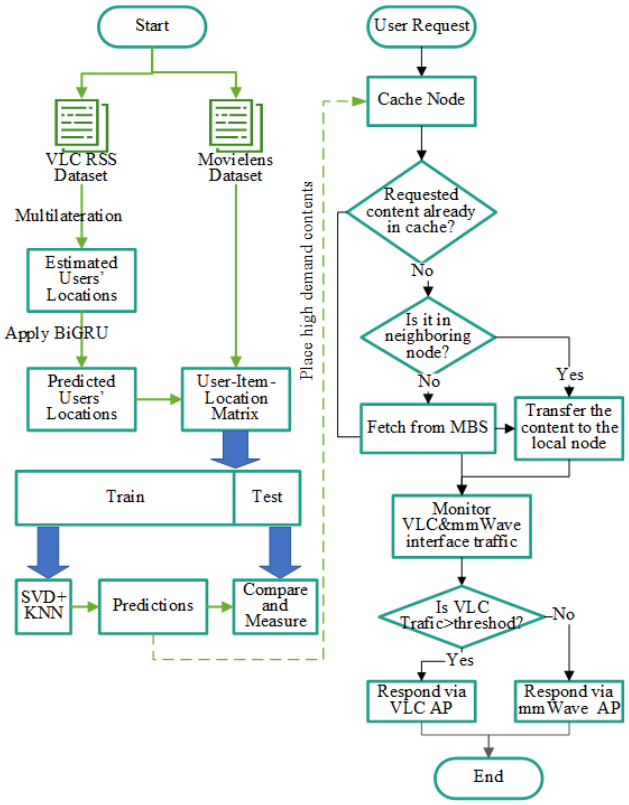

Fig. 5. Content Caching Process

content in the n-th request. The average cache hit rate is then given by:

$$H = \frac{1}{B}\sum_{b=1}^{B} \mathrm{Hit}_t^b \tag{22}$$

The objective function for cache placement is driven by the need to maximize cache hits. This can be mathematically expressed as:

$$\max_{\{q_{b,j}\}} \quad H = \frac{1}{B}\sum_{b=1}^{B} \frac{\sum_{n\in\mathcal{N}_t^b} D_b(j).q_{b,j}}{\mathcal{N}_t^b} \tag{23}$$

$$\text{subject to} \quad C1: \sum_{j=1}^{N} q_{b,j}\cdot\phi_j \le C_b, \quad \forall b\in\mathcal{B}, \tag{23a}$$

$$C2: \sum_{j\in J} q_{b_j} = 1, \quad \forall b\in\mathcal{B}, \tag{23b}$$

$$C3: q_{b,j}\in\{0,1\}, \quad \forall b\in\mathcal{B},\ j\in\mathcal{J}. \tag{23c}$$

where the constraint *C1* ensures that no VLC-gNB can cache more data than its allocated storage. Constraints *C2* and *C2* show a binary variable that indicates whether content j is cached in the VLC-gNB, which ensures the content that has already been cached in a VLC-gNB should not be cached multiple times. This avoids redundant content placement and inefficient use of cache resources.

The SDN controller periodically updates the cache at each VLC-gNB based on the predicted demand and user mobility patterns. This dynamic caching approach ensures that content with higher demand remains available, while less relevant content is replaced. The location-based content caching algorithm is shown in Algorithm 2, and the overall caching process is illustrated in Figure 5.

Algorithm 2 Location-Based Content Caching Algorithm Using ECF

Require: $\mathcal{U}$: Set of users, $\mathcal{J}$: Set of content, $\mathcal{B}$: Set of VLC-gNB, $\hat{l}_{u,t+1}$: Predicted location of user u at time $t+1$, r_{ij}: Rating of content j by user i, C_b: Cache capacity of VLC-gNB b

Ensure: Optimal caching decisions for VLC-gNBs to maximize the overall cache hit rate.

for each user $u_i \in U$ **do**
  Retrieve predicted location $\hat{l}_{u,t+1}$ from algorithm 1
  for each content $j \in J$ **do**
    Calculate the predicted rating $\hat{r}_{ij}$ using Equation (17)
  end for
end for
for each VLC-gNB $b \in B$ **do**
  Identify users located within the coverage area of VLC-gNB b
  for each user u_i in Step 8 **do**
    Select contents with the highest $D_b(j)$ using (19)
  end for
  if the cache is not full **then**
    Place the selected popular contents at the VLC-gNB
  else
    Remove outdated content and replace it with the new high-demand content
  end if
end for
for each user request for content j **do**
  if content j is cached at VLC-gNB b **then**
    Count the request as a cache hit
  else
    Fetch the content from a neighboring VLC-gNB or MBS
  end if
  Calculate the cache hit rate for VLC-gNB b using (21)
  Maximize the total cache hit rate across all VLC-gNBs using (23)
end for
Return: Optimal cache placement decisions for all VLC-gNBs.

5. EXPERIMENT AND ANALYSIS

5.1 Experimental environment

Experimental setup The experimental setup simulates an indoor Mobile Edge Computing environment with VLC and SDN technologies. The simulation is conducted using a hybrid approach that incorporates both MATLAB and Python for different aspects of the experiment. MATLAB is used to simulate the MEC environment, user mobility using RSS-based multilateration, and cache management, while external Python models handle user location prediction and content preference modeling.

To simulate user mobility, MATLAB uses RSS-based multilateration techniques to estimate the current location of users in terms of x and y coordinates. A BiGRU model is implemented in Python using PyTorch to predict future

user locations. The BiGRU model receives historical user movement data and predicts users' future coordinates. This predicted location data is fed into the MATLAB simulation to optimize cache placement decisions based on anticipated user movement. At the same time, an ECF model is also implemented in Python to predict user content preferences.

Experimental dataset This work uses two distinct datasets to simulate the prediction of user locations and content requests in MEC-enabled VLC networks. The first dataset used in this work contains RSS values measured from four different VLC APs, labeled as VLC Light A, B, C, and D (Ali et al. (2021)). The RSS values were recorded at various spatial locations and time intervals. RSS-based multilateration technique was used to estimate the users' locations and further refined using a BiGRU model, which captures temporal movement patterns to forecast the user's future location. The second dataset is the widely-used MovieLens dataset (Harper and Konstan (2015)), which contains user-item interactions and ratings for various movies. This dataset is used to model and predict user content preferences based on the predicted users' location. By applying ECF techniques, the model forecasts which content a user is likely to request based on their locations and previous behaviour and that of similar users. The predicted content requests are then used to make intelligent cache placement decisions at edge nodes, optimizing the cache hit ratio and enhancing content delivery efficiency.

Experimental parameters A simulation experiment platform was established to assess the performance of the edge caching algorithm proposed. The parameters for the simulation are detailed in Table 2.

Table 2. System Simulation Parameters

Parameters	**Value**
Half-power angle of the lamp, $\phi_{1/2}$	30°
Transmitted light intensity, I_i	35dBm
Effective area of the receiver, A_r	1cm^2
Vertical distance between AP and user, z_i	17.50cm
Capacity of edge cache node	[32,256]GB
Number of UEs	[5,25]
Relative cache size	[10%,100%]
Content Size	[50,160]MB

Evaluation metrics The metrics shown below, which have been used in this work, provide a comprehensive evaluation of the system's performance in predicting user mobility and content preferences and optimizing cache placement in the indoor edge computing environment.

- The hit ratio: This metric measures the percentage of user content requests successfully served by the edge cache without fetching data from the core network. A higher cache hit ratio indicates better caching efficiency. It is calculated as:

$$\text{Cache Hit Ratio} = \frac{\sum_{j=1}^{N} \text{Cache Hits}_j}{\sum_{j=1}^{N} \text{Total Requests}_j} \tag{24}$$

where Cache Hits_i represents the number of requests for content v_j that were served directly by the cache, Total Requests_i is the total number of requests for content v_j whether served by the cache or fetched from the core network, N is the total number of distinct content requests made by all users during the considered time period. This formula calculates the Cache Hit Ratio by summing up the successful cache hits for each content item and dividing this by the total number of requests for the content.

- MAE for BiGRU: Mean Absolute Error (MAE) evaluates the accuracy of the predicted user location by calculating the average absolute difference between the predicted and actual coordinates. It is computed as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left(|\hat{x}_i - x_i| + |\hat{y}_i - y_i|\right) \tag{25}$$

where $\hat{x}_i$ and $\hat{y}_i$ are the predicted location, x_i and y_i represent the actual coordinates, and n is the number of user locations.

- RMSE for BiGRU: Root Mean Squared Error (RMSE) measures the prediction error for user locations, considering the square root of the average squared differences between predicted and actual locations. It is calculated as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2\right]} \tag{26}$$

where $\hat{x}_i$ and $\hat{y}_i$ represent the predicted location, x_i and y_i are the actual coordinates, and n is the number of user locations.

- The R^2 for BiGRU: The coefficient of determination measures the proportion of variance in the actual values that is predictable from the predicted values. The R^2 can be calculated as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left((\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2\right)}{\sum_{i=1}^{n}\left((\bar{x} - x_i)^2 + (\bar{y} - y_i)^2\right)} \tag{27}$$

where $\hat{x}_i$ and $\hat{y}_i$ represent the predicted location, x_i and y_i are the actual coordinates, $\bar{x}$ and $\bar{y}$ are the mean of the actual locations, and n is the number of user locations.

- MAE for ECF: The MAE measures the average magnitude of the errors between the predicted and actual ratings without considering their direction, which can be computed as:

$$\text{MAE} = \frac{1}{N}\sum_{(i,j)\in N}|\hat{r}_{ij} - r_{ij}| \tag{28}$$

where r_{ij} is the actual rating, $\hat{r}_{ij}$ is the predicted rating, and N is the total number of predictions in the test set.

- RMSE for ECF: RMSE evaluates the error between the actual content ratings and the predicted ratings in collaborative filtering. Lower RMSE values indicate that the predicted content preferences are closer to the actual preferences.

$$RMSE = \sqrt{\frac{1}{N}\sum_{(i,j)\in N}(r_{ij} - \hat{r}_{ij})^2} \tag{29}$$

where r_{ij} is the actual rating, $\hat{r}_{ij}$ is the predicted rating, and N is the total number of predictions in the test set.

Benchmark algorithms To evaluate the performance of the algorithm proposed, the following existing baseline approaches were selected.

- Gated Recurrent Unit (GRU): A simpler Gated Recurrent Unit that captures temporal dependencies but processes data in only one direction, limiting its ability to predict future events based on past data.
- Bidirectional Long Short-Term Memory (Bi-LSTM): A variant of LSTM designed to capture long-term dependencies using memory cells, and it processes data in both directions, forward and backward.
- Long Short-Term Memory (LSTM): A type of RNN (Recurrent Neural Network) with memory cells that is designed to handle long-term dependencies in sequential data but processes data in only one direction.
- Mobility-aware Proactive Caching with Federated Learning (MPCF) (Yu et al. (2020)): MPCF is a proactive edge caching scheme designed to address the mobility challenges in vehicular networks. It utilizes federated learning to predict content popularity collaboratively across multiple vehicles while preserving user privacy. The algorithm adjusts the cache replacement policy based on vehicle mobility patterns and preferences.
- Regression-based Proactive User Caching (RPUC) (Yang et al. (2018)): RPUC is a caching strategy focusing on location-customized caching by estimating future content hit rates using regression models. It incorporates user mobility and content popularity variations to make caching decisions.
- Least Frequently Used (LFU): LFU ranks content based on its access frequency, with less frequently accessed content being evicted first. This approach can maintain popular content in the cache for a longer time.
- Least Recently Used (LRU): LRU is a widely used caching algorithm that prioritizes content based on its recent access history. The least recently accessed content is replaced when the cache reaches its capacity.

5.2 Results and analysis

Prediction models performance evaluation

(1) Performance evaluation for user's location prediction

Loss is a crucial aspect of neural networks, as it represents the error in the predictions made by the model. Figure 6 shows the validation loss of the proposed location prediction model after 50 iterations, with the cumulative loss approaching 0.015.

The performance of several models for user location prediction, including BiGRU, GRU, Bi-LSTM, and LSTM, was evaluated using RMSE, MAE, and R^2. The BiGRU model, which processes data in both forward and backward directions, outperformed all other models, achieving the best RMSE of 0.8349, MAE of 0.6655, and R^2 of 0.9709. This highlights its superior accuracy in user location prediction, capturing both past and future dependencies.

The GRU, a unidirectional model, showed lower performance with an RMSE of 2.0307, MAE of 1.2517, and R^2 of 0.9484, indicating that the lack of bidirectionality limits its effectiveness. The Bi-LSTM, despite being bidirectional, performed less effectively than BiGRU, with an RMSE of 3.2854, MAE of 2.7457, and R^2 of 0.8950. The Bi-LSTM model struggles with long-term dependencies due to the complexity of its memory structure. Lastly, the LSTM model, also unidirectional, showed the highest RMSE of 3.9946, MAE of 3.1784, and the lowest R^2 of 0.8467, performing the worst among the models. The results confirm that BiGRU is the most effective model for user location prediction, with its bidirectional nature providing better predictions compared to the other models, including Bi-LSTM and LSTM. The simpler architecture of BiGRU contributes to its superior performance, particularly in this specific task.

Fig. 6. Training and Test Loss over Epochs

Table 3. Location prediction results

Model	RMSE	MAE	R^2
BiGRU	0.8349	0.6655	0.9709
GRU	2.0307	1.2517	0.9484
Bi-LSTM	3.2854	2.7457	0.8950
LSTM	3.9946	3.1784	0.8467

(2) Performance evaluation for content request prediction

The content request prediction using the ECF model shows strong performance, with an RMSE of 0.9293 and an MAE of 0.7414. These metrics indicate the model's capability to predict user preferences accurately. The ECF model captures nuanced patterns by integrating user preferences with location-based proximity, ensuring context-aware recommendations. The results validate that incorporating location data into collaborative filtering improves prediction accuracy, making the ECF model suitable for environments where their physical context influences user behavior. This approach enhances the content's relevance and provides a personalized user experience, as predictions are adjusted based on rating patterns and spatial factors.

Caching Performance of the Proposed Strategy

(1) The effect of the relative cache size on cache hit ratio

As the cache capacity increases, the cache hit ratio generally improves for all algorithms, as seen in the Figure 7. With more storage space at the edge nodes,

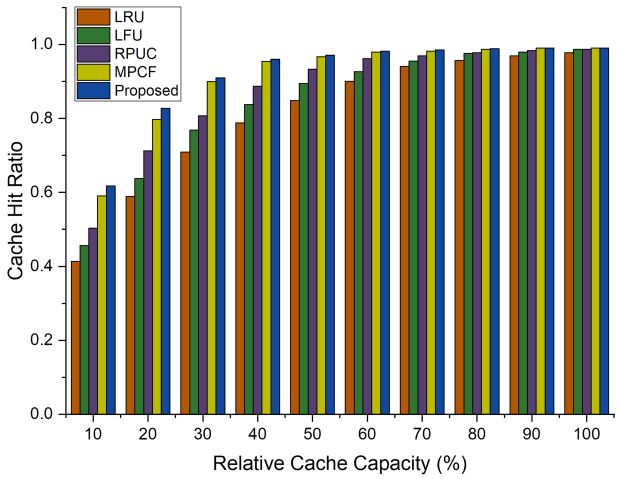

Fig. 7. The Effect of relative cache size on performance

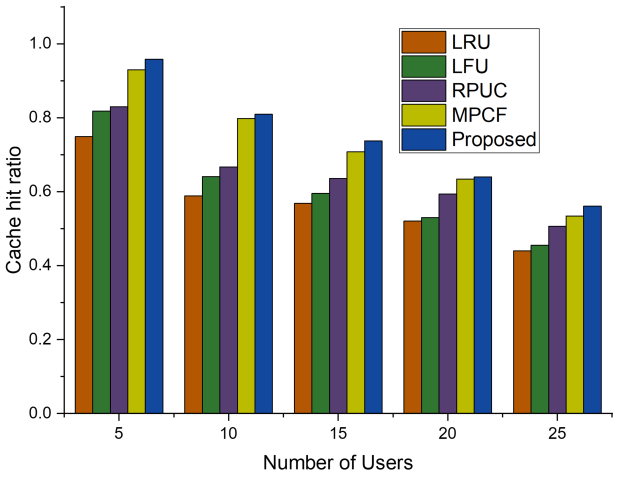

Fig. 8. The effect of the number of users on performance

they can accommodate a larger variety of content, increasing the chances of fulfilling user requests locally without retrieving data from the central servers. Basic algorithms like LRU and LFU experience an initial boost in hit ratio as capacity grows; however, they plateau as their simple caching mechanisms struggle to prioritize content optimally in diverse user environments. More advanced algorithms like RPUC and MPCF show a steadier increase in cache hit ratio as they incorporate predictive models based on user location and mobility patterns. The proposed algorithm outperforms all others across varying cache capacities.

(2) The effect of the number of users on cache hit ratio

The Figure 8 shows that as the number of users grows beyond the initial phase, the cache hit ratio of most algorithms begins to decrease as cache storage capacity becomes insufficient to handle the increasing variety of content requests. Traditional algorithms like LRU and LFU show a sharper decline in cache hit ratio because they rely on basic recency and frequency heuristics, which are less effective at handling diverse content requests. On the other hand, advanced algorithms such as RPUC and MPCF maintain a relatively higher cache hit ratio for a more extended period by incorporating sophisticated prediction mechanisms. RPUC leverages location-customized caching and ridge regression to estimate content hit rates, adapting more effectively to changing user demands at different locations. MPCF, with its federated learning approach, predicts content popularity based on collective user preferences across different vehicles and edge nodes, making it more resilient to increasing user diversity. However, the proposed algorithm consistently outperforms the other methods across all user densities.

(3) The effect of the number of edge cache nodes on cache hit ratio

Figure 9 illustrates that as the number of edge cache nodes increases, all caching algorithms display improvements in cache hit ratio, reflecting their ability to take advantage of additional storage and more localized content placement. Traditional algorithms such as LRU and LFU, prioritizing recently or frequently requested content, show initial gains but

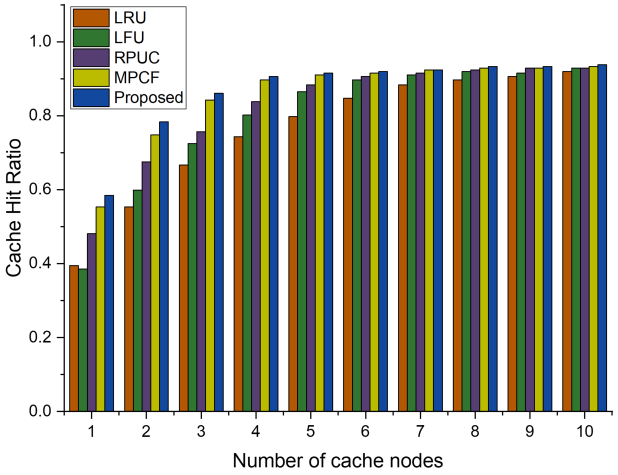

Fig. 9. The effect of the number of edge cache nodes on performance

plateau more quickly as additional nodes are added. This suggests that these methods fail to fully utilize the additional resources provided by more edge cache nodes. More advanced techniques, such as RPUC and MPCF, leverage user mobility and location-awareness to distribute cached content across the network more effectively. As a result, these models see more consistent improvements with increased cache nodes. RPUC's location-based caching strategy and MPCF's federated learning approach make them better suited to dynamically changing content demands and mobile users, enabling them to sustain higher cache hit ratios as more nodes are introduced.

6. CONCLUSION AND FUTURE WORKS

This study proposes an efficient edge caching strategy for mobile edge computing, combining BiGRU-based user location prediction with location-aware collaborative filtering to accurately anticipate content demands. By leveraging user mobility and preference patterns, popular content is proactively cached at VLC-gNBs, reducing retrieval latency.

The hybrid framework demonstrates strong applicability and adaptability to real-time, high-mobility indoor scenarios, especially within hybrid VLC-mmWave networks managed via SDN orchestration, enhancing caching efficiency by aligning placement with user mobility and preferences.

However, for large-scale deployments involving hundreds of users and extensive content libraries, real-time responsiveness poses challenges. Frequent cache updates introduce management overhead that can affect system performance. To address these concerns, future work will explore the integration of lightweight and adaptive cache replacement policies such as LRU and LFU algorithms hybridized with predicted demand metrics. These heuristics aim to balance caching effectiveness with operational cost, minimizing overhead while sustaining cache hit rates.

Moreover, large-scale environments are expected to benefit from parallel processing and distributed edge computing capabilities to maintain low latency and scalability. Future investigation will focus on optimizing these aspects alongside considerations of network load balancing, energy efficiency, and fairness in content delivery. These advancements will enhance the practicality, scalability, and real-time adaptability of the proposed caching framework.

REFERENCES

Ali, K., Cosmas, J., Zhang, Y., Zhang, H., Meunier, B., Jawad, N., Zhang, X., Shi, L., Gbadamosi, J., and Savov, A. (2021). Measurement campaign on 5g indoor millimeter wave and visible light communications multi component carrier system. *IEEE Transactions on Broadcasting*, 68(1), 156–170.

Cetin, M. and Abo Aisha, A.E.S. (2023). Variation of al concentrations depending on the growing environment in some indoor plants that used in architectural designs. *Environmental science and pollution research*, 30(7), 18748–18754.

Chi, N., Zhou, Y., Wei, Y., and Hu, F. (2020). Visible light communication in 6g: Advances, challenges, and prospects. *IEEE Vehicular Technology Magazine*, 15(4), 93–102.

Chien, W.C., Weng, H.Y., and Lai, C.F. (2020). Q-learning based collaborative cache allocation in mobile edge computing. *Future generation computer systems*, 102, 603–610.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Harper, F.M. and Konstan, J.A. (2015). The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4), 1–19.

Hu, Y.C., Patel, M., Sabella, D., Sprecher, N., and Young, V. (2015). Mobile edge computing—a key technology towards 5g. *ETSI white paper*, 11(11), 1–16.

Kombo, V.A., Kryvinska, N., Abdalsalam, M., and Adnan, A. (2025). Collaborative video caching strategy based on delay and energy for software-defined hybrid vlc and mmwave networks. *Journal of Control Engineering and Applied Informatics*, 27(1), 51–62.

Koren, Y., Rendle, S., and Bell, R. (2021). Advances in collaborative filtering. *Recommender systems handbook*, 91–142.

Li, H., Sun, C., Li, X., Xiong, Q., Wen, J., Wang, X., and Leung, V.C. (2020). Mobility-aware content caching and user association for ultra-dense mobile edge computing networks. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*, 1–6. IEEE.

Li, Y., Feng, L., Yu, P., Yang, Y., and Li, W. (2016). Performance analysis of indoor-outdoor wireless caching relay system. In *2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 1–4. IEEE.

Ma, J., Li, D., and Wang, J. (2020). Indoor-outdoor wireless caching relay system for power transformation station in smart grid. In *The 8th International Conference on Computer Engineering and Networks (CENet2018)*, 794–801. Springer.

Nguyen, L.V., Vo, Q.T., and Nguyen, T.H. (2023). Adaptive knn-based extended collaborative filtering recommendation services. *Big Data and Cognitive Computing*, 7(2), 106.

Sang, Z., Guo, S., Wang, Q., and Wang, Y. (2021). Gcs: Collaborative video cache management strategy in multi-access edge computing. *Ad Hoc Networks*, 117, 102516.

Shen, S., Yu, C., Zhang, K., and Ci, S. (2022). Collaborative edge caching with personalized modeling of content popularity over indoor mobile social networks. In *ICC 2022-IEEE International Conference on Communications*, 4114–4119. IEEE.

Somesula, M.K., Rout, R.R., and Somayajulu, D.V. (2021). Deadline-aware caching using echo state network integrated fuzzy logic for mobile edge networks. *Wireless Networks*, 27(4), 2409–2429.

Wei, H., Luo, H., and Sun, Y. (2020). Mobility-aware service caching in mobile edge computing for internet of things. *Sensors*, 20(3), 610.

Yang, P., Zhang, N., Zhang, S., Yu, L., Zhang, J., and Shen, X. (2018). Content popularity prediction towards location-aware mobile edge caching. *IEEE Transactions on Multimedia*, 21(4), 915–929.

Yao, J., Han, T., and Ansari, N. (2019). On mobile edge caching. *IEEE Communications Surveys & Tutorials*, 21(3), 2525–2553.

Yu, Z., Hu, J., Min, G., Zhao, Z., Miao, W., and Hossain, M.S. (2020). Mobility-aware proactive edge caching for connected vehicles using federated learning. *IEEE Transactions on Intelligent Transportation Systems*, 22(8), 5341–5351.

Yuan, B., Guo, S., and Wang, Q. (2021). Joint service placement and request routing in mobile edge computing. *Ad Hoc Networks*, 120, 102543.

Zaman, S.K.u., Jehangiri, A.I., Maqsood, T., Ahmad, Z., Umar, A.I., Shuja, J., Alanazi, E., and Alasmary, W. (2021). Mobility-aware computational offloading in mobile edge networks: a survey. *Cluster Computing*, 1–22.

Zaman, S.K.u., Jehangiri, A.I., Maqsood, T., Haq, N.u., Umar, A.I., Shuja, J., Ahmad, Z., Dhaou, I.B., and Alsharekh, M.F. (2023). Limpo: Lightweight mobility prediction and offloading framework using machine learning for mobile edge computing. *Cluster Computing*, 26(1), 99–117.