

## A DQN-Based Cache Strategy for Mobile Edge Networks

Siyuan Sun<sup>1,\*</sup>, Junhua Zhou<sup>2</sup>, Jiuxing Wen<sup>3</sup>, Yifei Wei<sup>1</sup> and Xiaojun Wang<sup>4</sup>

<sup>1</sup>School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876, China

<sup>2</sup>State Key Laboratory of Intelligent Manufacturing System Technology, Beijing Institute of Electronic System Engineering, Beijing, 100854, China

<sup>3</sup>Ningbo Sunny Intelligent Technology Co., LTD, Yuyao, 315400, China

<sup>4</sup>Dublin City University, Dublin 9, Ireland

\*Corresponding Author: Siyuan Sun. Email: sunsy@bupt.edu.cn

Received: 25 May 2021; Accepted: 11 July 2021

**Abstract:** The emerging mobile edge networks with content caching capability allows end users to receive information from adjacent edge servers directly instead of a centralized data warehouse, thus the network transmission delay and system throughput can be improved significantly. Since the duplicate content transmissions between edge network and remote cloud can be reduced, the appropriate caching strategy can also improve the system energy efficiency of mobile edge networks to a great extent. This paper focuses on how to improve the network energy efficiency and proposes an intelligent caching strategy according to the cached content distribution model for mobile edge networks based on promising deep reinforcement learning algorithm. The deep neural network (DNN) and Q-learning algorithm are combined to design a deep reinforcement learning framework named as the deep-Q neural network (DQN), in which the DNN is adopted to represent the approximation of action-state value function in the Q-learning solution. The parameters iteration strategies in the proposed DQN algorithm were improved through stochastic gradient descent method, so the DQN algorithm could converge to the optimal solution quickly, and the network performance of the content caching policy can be optimized. The simulation results show that the proposed intelligent DQN-based content cache strategy with enough training steps could improve the energy efficiency of the mobile edge networks significantly.

**Keywords:** Mobile edge network; edge caching; energy efficiency

### 1 Introduction

With the rapid development of wireless communication networks and the fast growing of smart devices, various mobile Internet applications has been spawned, such as voice recognition, autonomous driving, virtual reality and augmented reality. These emerging services and applications put forward higher requirements for high capacity, low latency and low energy consumption. By



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

deploying edge server at the wireless access network, the end user's computing tasks can be executed near the edge of the network, which could effectively reduce the congestion of the backhaul network, greatly shortens the service delay, and meets the needs of delay-sensitive applications. The mobile edge networks are essentially the subsidence of cloud computing capability, and could provide third-party application at the edge nodes, making it possible for service innovation of mobile edge entry.

With the popularity of video services, the traffic of video content grows explosively. The huge data traffic is mainly caused by the redundant transmission of popular content. Edge nodes also have certain storage capacity for caching, and edge caching is becoming more and more important. Deploying a cache in an edge network can avoid data redundancy caused by many repeated content deliveries. By analyzing the content popularity and proactively caching the popular content from the core network to the mobile edge server, then the request for the repeated content could be transmitted directly from the nearby edge nodes without going back to the remote core network, which greatly reduces the transmission delay and effectively alleviates the pressure on the backhaul link and the core network. Edge cache has been widely studied because it can effectively improve user experience and reduce energy consumption [1]. Enabling caching ability in a mobile edge system is a promising approach to reduce the use of centralized databases [2]. However, due to the edge equipment size, the communicating, computing and caching resources in a mobile edge network are limited. Besides, end users' mobility makes mobile edge networks, becoming a dynamic system, where the energy consumption may increase because of improper caching strategies. In order to address these problems, we focus on how to improve energy efficiency in a cache-abled mobile edge network through smart caching strategies. In this article, we study a mobile edge network with unknown content popularity, and design a dynamic and smart content cache policy based on online learning algorithm which utilizes deep reinforcement learning (DRL).

The rest of this article is arranged as follows. Firstly, Section 2 elaborated the existing related work and some solutions on the content cache strategy in mobile edge networks. Section 3 introduce a cache-enabled mobile edge network scenario and builds related system models and energy models. Section 4 analyzes the energy efficiency of the system, and formulates the optimization problem in the system according to the deep reinforcement learning models, then the caching content distribution strategy is proposed to solve the energy efficiency optimization problem of mobile edge networks. Finally, we design some numerical simulations and analyze the results, which show that the proposed strategy could greatly improve mobile edge networks energy efficiency without decreasing the system performance.

## **2 Related Work**

In this section, we will investigate the existing related work and some solutions on mobile edge networks with caching capability and artificial intelligence (AI) in network resource management.

### ***2.1 Energy-Aware Mobile Edge Networks with Cache Ability***

By simultaneously developing computing offloads and smart content caches near the edge of the network, mobile edge network can further improve the efficiency of network content distribution and computing capabilities, and effectively reducing latency and improving service quality and the energy consumption of cellular networks. Thus, adding cache ability at the network edge has become one of the most important approaches [3,4]. Authors in [5] separated the controlling and communicating functions of heterogeneous wireless networks with software defined network (SDN)-based techniques. Under the proposed network architecture, the cache-enabled of macro base stations and relay nodes are

overlaid and cooperated in a limited backhaul scenario to meet service quality requirement. Authors in [6] investigated cache policies which aim to improve energy efficiency in information centric networks. Numerous papers [7,8] focus on caching policies based on user demands, e.g., authors in [9] proposed a light-weight cooperating edge storage management strategy thus the utilization of edge caching can be maximized and the bandwidth cost can be decreased. Jiang et al. [10] proposed a caching and delivering policy for femto cellulars where access nodes and user devices are all able to cache local contents, the proposed policy aims at realizing a cooperative allocation strategy for communicating and caching resources in a mobile edge network. A deep-Learning-based content popularity prediction algorithm is developed for software defined networks in [11]. Besides, due to the users' mobility, how to cache contents properly is challenging in mobile edge network, thus many works [12–15] aimed at addressing mobility-aware caching problems. Sun et al. [16] proposed a mobile edge cloud framework which uses big medical sensor data and aims at predicting diseases.

Those works inspire us to research on how to improve the energy consumption of the mobile edge networks by taking advantage of cache ability. However, cache strategies from the work above are all based on user demands and behavior, which are features that are difficult to extract or forecast. To solve this problem, many researchers work on using the architectures of the network function virtualization or software defined networks. By separating control and communicate planes and virtualizing network device functions, those techniques realized a flexible and intelligent way to manage edge resources in mobile edge networks. Thus, many smart energy efficiency-oriented caching policies are able to be integrated as network applications, which are operated by network managers and run on top of a centralized network controller [17]. Li et al. [18] surveyed Software-Defined Network Function Virtualization. Mobility Prediction as a Service which was proposed in [19] offers an on-demand long-term management by predicting user activities, moreover, the function is virtualized as a network service, which is fully placed on top of the cloud and works as a cloudified service. Authors in [20] designed a network prototype which takes advantage of not only content centric networks but also Mobile Follow-Me Cloud, therefore the performance of cache-aided edge networks are improved. Those works suggested feasible paradigms of mobile edge network virtualization with cache ability. Furthermore, authors in [21] designed a content-centric heterogeneous networks architecture which is able to cache contents and compute local data, in the analyzed network scenario, users associated to various network services but were all allowed to share the communication, computation and storage resources in one cellular. Tan et al. [22] proposed a full duplex-enabled software defined networks framework for mobile edge computing and caching scenario, and the first framework suits network services which are sensitive to data rate, while the second one suits network services which are sensitive to data computing speed.

## ***2.2 Artificial Intelligence in Network Management***

In recent years, deploying more intelligence in networks is a promising approach to realize effectively organizing [23,24], managing and optimizing network resources. Xie et al. [25] investigated how to use machine learning (ML) algorithms to add more intelligence to software defined networks. Reinforcement learning (RL) and related algorithms has been adapted for automatic goal-oriented decision-making for ages [26]. Wei et al. [27,28] suggested a joint optimization for edge resource allocating problem, the proposed strategy uses the model-free actor-critic reinforcement learning to solve the joint optimization problems of content caching, wireless resource allocation and computation offloading, thus the overall network delay performance is improved. Qu et al. [29,30] proposed a novel controlled flexible representation and a novel secure and controllable quantum image steganography algorithm for quantum image, this algorithm allows the sender to control all the process stages during a

content transmitting phrase, thus a better information-oriented security is obtained. Deploying cache in the edge network can enhance content delivery networks and reducing the data traffic caused by a large number of repeated content requests [31]. There are many related works, mainly focusing on computing offloading and cache decisions and resources allocation [32].

However most of RL algorithms needs lots of computation power. Although the SDN architecture offers an efficient flow-based programmable management approach, the local controllers mostly have limited resources (e.g., storage, CPU.) for data processing. Thus an optimization algorithm with controllable computing time is required for mobile edge networks. Amokrane et al. [33] proposed a computation efficient approach which is based on Ant-Colony optimization to solve the formulated flow-based routing problem, the simulation results showed that the Ant Colony-based approach can substantially decrease computation time comparing with other optimal algorithms.

Various deep reinforcement learning main architectures and algorithms are presented in [34], in which reinforcement learning algorithms are reproduced with deep learning methods. The authors highlighted the impressive performance of deep neural networks (DNN) when solving various management problems (e.g., video gaming). With development of deep learning, reinforcement learning has been successfully combined with DNN [35]. In wireless sensor networks, Toyoshima et al. [36] described a design of a simulation system using DQN.

This paper takes advantages of DQN, which can efficiently obtain a dynamic optimized solution without having a priori knowledge of the dynamic statistics.

### 3 System Model

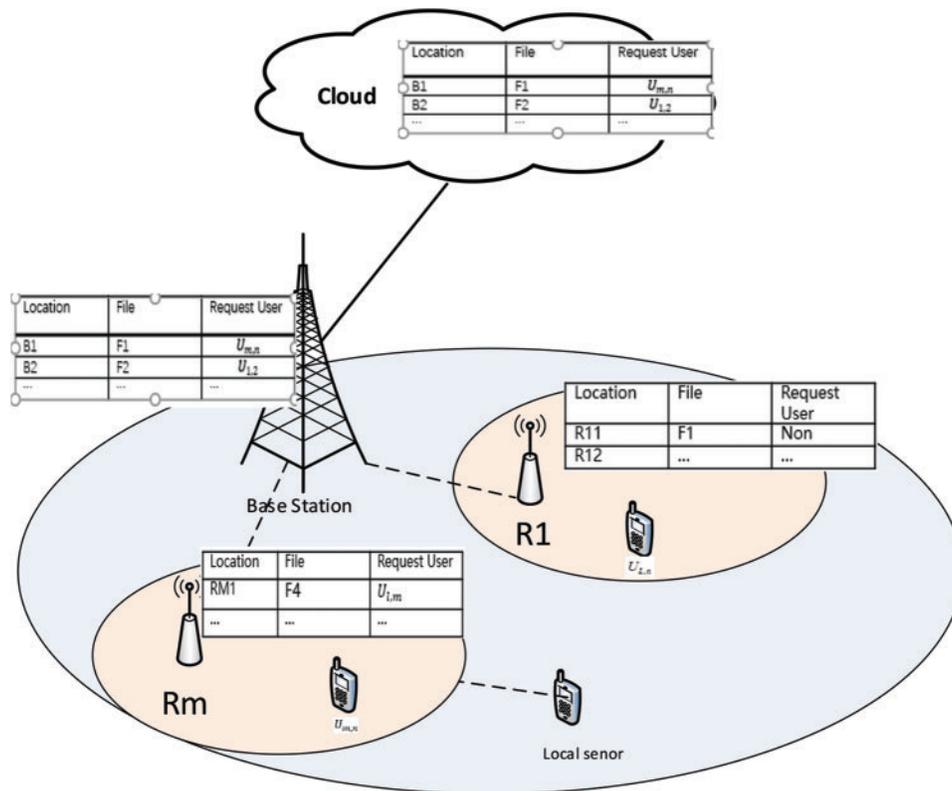
This section analyzes a caching edge network scenario with unknown content popularities, then we build the energy model for the system. The main notations used in the rest of this article are listed in Tab 1.

**Table 1:** Table of main notations

Symbols	Definitions
$B$	Macro base station
$R$	Set of relay nodes
$U_m$	Set of users who are served by the $m^{\text{th}}$ RN
$F^c$	Set of local cached files
$F^r$	Set of requested files
$M$	Number of RNs
$N$	Number of storages on each edge node
$K$	Number of requested files
$X_n$	The $n^{\text{th}}$ storage block on the edge node $X$
$BF$	Cloud Storage
$C_m(t)$	Set of files required by user set $U_m$ at time $t$
$P(t)$	Energy consumption at time $t$
$\eta(t)$	Energy efficiency at time $t$

### 3.1 Network Model

A typical mobile edge network scenario is shown in Fig. 1, which contains not only communicating but also caching resources. The edge accessing devices include macro base stations (MBSs) and relay nodes (RNs), they all have different communicate and cache abilities, but any of them can connect to end users and cache data from end user or far cloud. We consider a set of relay nodes  $R$  with limited storage space (in this paper we assume each RN can store  $N$  files) while the macro base station  $B$  has unlimited cache space since it connects to far cloud. This article considers a wireless network where orthogonal frequency division multiplexing-based multiple access technology is adopted, thus the loss of spectral efficiency can be ignored.



**Figure 1:** Example of an edge cellular network with caching ability

We consider a cellular wireless access network that consists of an MBS and a set of RNs, the a large number of end users who are served by relay  $R_m$  are denoted as  $U_m$ . Each RN can connect to the MBS and its neighboring RNs. Due to the caching ability,  $N$  is the maximum servicing user number of an RN. At time  $t$ , the files cached in  $n^{th}$  cache block on relay node  $R_m$  is denoted as  $F_{m,n}^c(t)$ , the files requested by user  $U_{m,n}$  is denoted as  $F_{m,n}^r(t)$ . We use variables  $I_{m,n}^{ct}$  and  $I_{m,n}^{rt}$  to express how the requested file  $F_{m,n}^r(t)$  are delivered to user  $U_{m,n}$

$$I_{m,n}^{ct}(t) = \begin{cases} 1, & \text{when } F_{m,n}^r(t) \in F^c(t) \\ 0, & \text{when } F_{m,n}^r(t) \notin F^c(t) \end{cases} \quad (1)$$

and

$$I_{m,n}^l(t) = \begin{cases} 1, & \text{when } F_{m,n}^r(t) = F_{m,n}^c(t) \\ 0, & \text{when } F_{m,n}^r(t) \neq F_{m,n}^c(t) \end{cases} \quad (2)$$

### 3.2 Energy Model

The optimization problem here is how to maximize the system energy efficiency. According to the network architecture which has been analyzed in Section 3.1, the system energy consumption in a mobile edge network can be divided into three components:

1. Basic power  $P_{basic}$ : the mobile edge network operational power, it takes into account all the power consumed by the macro base station and RNs other than the transmitting power.
2. Transmitting power in the mobile edge network  $P_{edge}$ : power consumption of data transferring among edge nodes, it happens when a requested file is cached in the mobile edge network.
3. Transmitting power between the mobile edge network and far cloud  $P_{cloud}$ : power consumption of data transferring between local nodes and far cloud, it happens when a requested file is not stored at the mobile edge network.

Thus the system total energy consumption can be written as:

$$P_{sys} = P_{basic} + P_{edge} + P_{cloud} \quad (3)$$

The  $P_{edge}$  and  $P_{cloud}$  are varying in practical system, especially with different placement of edge nodes and users' equipment. To simplify the analysis, here consider a dynamic network, where the location palaces of the communication equipment are unpredictable but follow a specific distribution, thus the power consumption of transmitting a file between two edge nodes can be a constant, we express the constant as  $\alpha_1$ . Similarity, the power consumption of transmitting a file between the edge network and cloud is denoted a constant  $\alpha_2$

According to the formula (1) and (2), the system energy consumption can be written as:

$$P_{sys}(t) = P_{basic} + \alpha_1 \sum_{m=1}^M \sum_{n=1}^N I_{m,n}^l(t) + \alpha_2 \sum_{m=1}^M \sum_{n=1}^N I_{m,n}^l(t), \quad (4)$$

where  $P_{sys}(t)$  denotes the system power consumption at time t.

In a mobile edge network, the propagation model can be represented by a function of radiated power:

$$p^r_x = \theta d^{-\lambda} p^t_x, \quad (5)$$

where  $P^r_x$  and  $P^t_x$  respectively denote transmitted and received power,  $\theta$  denotes correction parameter, d denotes the distance between the transmitting device and receiving device,  $\lambda$  is used to denote the path loss exponent in the formulation.

In order to meet the user QoS, a fixed transmission rate  $R_0$  should be a maintained. Therefore, the received power can be calculated as follow according to Shannon formulation:

$$p^r_x = N_0 W (2^{R_0/W} - 1), \quad (6)$$

where W and  $N_0$  respectively denote transmission bandwidth Gaussian white noise power density.

Comparing to a traditional mobile edge network, a cache-enabled mobile edge network focuses on delivering contents to end users rather than transmitting data. Thus the EE in the analyzed system

is defined as the size of delivered contents during a time slot, instead of transmitted data. The EE can be written as follow:

$$\eta(t) = \frac{N \times M \times L}{P_{system}} \tag{7}$$

where L denote content size of each file.

Therefore, the system EE maximization problem can be written as an optimization formulation as follow:

$$\arg \max_{I^l_{m,n}(t), I^r_{m,n}(t)} \eta(t)$$

$$s.t. \quad SINR(t^u_i) \geq \gamma$$

$$F^c_{m,n}(t-1) = F^r_{m,n}(t), \forall m \in \{1, \dots, M\}, \forall n \in \{1 \dots N\}$$

$$I^l_{m,n}(t), I^r_{m,n}(t) \in \{0, 1\} \tag{8}$$

where the first constraint means that the users' QoS should be satisfied by defining the minimal SINR threshold, the second constraint means that each requested file should be transferred to the edge node which serves the user. We use the optimization variables  $I^l_{m,n}(t)$ ,  $I^r_{m,n}(t)$  to represent the cache distribution strategy.

#### 4 Solution with DQN Algorithm

In this section we first formulate the energy efficiency-oriented cached content distribution optimization problem, which has been discussed in Section 3, according to reinforcement learning model, then we give the solution of a caching strategy based on DQN algorithm.

##### 4.1 Reinforcement Learning Formulation

There is an agent, an action space and a state space in a general reinforcement learning model, the agent learns which action in the action space should be taken at which state in the state space. For example, we use  $s_\tau$  to denote the cached content distribution state in a mobile cache-aided edge network, and  $s_\tau$  updates across adjustment steps  $\tau = \{1, 2 \dots\}$ . Thus, the caching distribution state at step  $\tau$  can be written as

$$s_\tau = F^c_{1,1}(\tau), \dots, F^c_{1,n}(\tau), F^c_{m,1}(\tau), \dots, F^c_{m,n}(\tau) \tag{9}$$

The state space is represented as S, where  $\forall s_\tau \in S$ .

During a time slot, the caching distribution state changes from an initial state  $s_1$  to a terminal state  $s_{term}$ . In order to deliver the requested content to corresponding user, the first caching distribution state  $s_1$  of time slot  $t$ , is the terminal state of last time slot,

$$s_1(t) = F^c_{1,1}(t-1), \dots, F^c_{1,n}(t-1), F^c_{m,1}(t-1), \dots, F^c_{m,n}(t-1). \tag{10}$$

And the terminal state of time slot t  $s_{term}(t)$  is the users' requests at t time slot, it can be represented as formulation (11):

$$s_{term}(t) = F^c_{1,1}(t), \dots, F^c_{1,n}(t), F^c_{m,1}(t), \dots, F^c_{m,n}(t). \tag{11}$$

In our problem, we assume that there is a blank storage block  $v$  in a mobile edge network, thus the cache distribution in the mobile edge network can be adjusted through swapping the empty cache space. The vacant storage block can cache the contents from a valid storage block which is located in a neighboring accessing node, thus the vacant storage block is movable in the edge network. The swapping action which is executed in iteration step  $\tau$  is represented as  $a_\tau$ . Thus the action space  $\mathcal{A}_\tau$  at iteration step  $\tau$  of the designed model can be written as

$$a_\tau = \begin{cases} \{a_{\text{neighbour RNs}}, a_{\text{BS}}\}, & \text{when } v \text{ is on an RN} \\ \{a_{\text{all RNs}}\}, & \text{when } v \text{ is on the BS} \end{cases} \quad (12)$$

The executed action serial, which starts from the initial state  $s_1$  to the terminal state  $s_{\text{term}}$ , during the time slot  $t$  is represented by  $\mathcal{A}_t$ ,

$$\mathcal{A}_t = \{a_\tau | \tau = 1, 2, \dots\} \quad (13)$$

The reinforcement learning system obtains an immediate reward after an action is performed. Because reinforcement learning algorithm is using for a model-free optimization problem. The value of the immediate reward is zero until the agent reaches the terminal state. We denote the immediate reward as  $r_\tau$ , according to the energy efficiency optimization problem which has been analyzed in Section 3,  $r_\tau$  can be calculated as following formulation:

$$r_\tau = \begin{cases} R(s, a), & \text{if } s_{\tau+1} = s_{\text{term}}, \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

where  $R(s, a)$  is an energy consumption function related to state and action.

#### 4.2 DQN Algorithm

RL problems can be described as the optimal control decision making problem in MDP. Q-learning is one of effective model-free RL algorithms, which has been analyzed in Section 4.2. The long-term reward, which is learned by Q-learning algorithm, of performing action  $a_\tau$  at environment state  $s_\tau$  is represented as a numerical value  $Q(s_\tau, a_\tau)$ . Thus, the agent in a dynamic environment performs the action which can obtain a maximal Q value. In this paper, the EE optimized caching strategy adjusts cached content distribution state in a mobile cache-aided edge network by using learned information of Q-learning algorithm. In each iteration, the Q-value is updated according to the formulation as follows:

$$Q(s_\tau, a_\tau) \leftarrow (1 - \alpha)Q(s_\tau, a_\tau) + \alpha(r_\tau + \gamma \max_{a_{\tau+1}} Q(s_{\tau+1}, a_{\tau+1})), \quad (15)$$

where  $\alpha$  is the learning rate ( $0 < \alpha < 1$ ),  $\gamma$  is the discount factor ( $0 < \gamma < 1$ ),  $r_\tau$  is the reward received when moving from the current state  $s_\tau$  to the next state  $s_{\tau+1}$ ,  $r_\tau$  can be calculated according to formulation (14).

Based on (15), Q-learning algorithm is effective when solving the problem with small state and action spaces, where a Q-table is able to be stored to represent all the Q values of each state-action pair. However, the process becomes extremely slow in a complex environment where the state and action space dimensions are very high. Deep Q-learning is a method that combines neural network and Q-learning algorithm. Neural network can solve the problem that needs to store and retrieve a large number of Q values.

The agent in our RL model is trained by DQN algorithm in this paper, the DQN algorithm uses two neural networks to realize the convergence of value function. As we have analyzed, due to the complex and high-dimensional network cached content distribution state space, calculating all the optimal Q-values requires significant computations. Thus, we approximate Q-values with a Q-function, which can be trained to the optimal Q value by updating the parameter  $\theta$ . The calculation formula for Q-function is as follows:

$$Q(s_\tau, a_\tau, \theta) \leftarrow (1 - \alpha)Q(s_\tau, a_\tau, \theta) + \alpha(r_\tau + \gamma \max_{a_{\tau+1}} Q(s_{\tau+1}, a_{\tau+1}, \theta^-) - Q(s_\tau, a_\tau, \theta)). \quad (16)$$

where  $\theta$  is the weight vector of the neural network, it is updated according to network training iterations.

DQN uses a memory bank to learn the previous experience. At each update, some previous experience is randomly selected for learning, while the target network is updated with two network structures with the same parameters and different network structures, making the neural network update more efficient. When the neural network can't maintain convergence, there will be some problems such as unstable training or difficult training. DQN uses experience playback and target networks to improve on these issues. Experience replay is usually used to store the collected sequence of observations  $(s, a, r, s')$ . This transfer information is stored in the cache, and the information in the buffer becomes the experience of the agent. The core idea of the experience replay mechanism is to train the DQN using the transfer information in the cache, rather than the information at the end of the loop. The experiences of each cycle are correlated with each other, so randomly selecting a batch of training samples from the cache will reduce the correlation between experiences and help to enhance the generalization ability of agents. At each epoch, the neural network is trained by minimizing the following loss function:

$$L(\theta) = \mathbb{E}[(y_i - Q(s_\tau, a_\tau, \theta))^2], \quad (17)$$

where  $y_i$  is the target value for iteration  $\tau$ .

The process of the proposed DQN-based cache strategy based can be illustrated with pseudocode as shown in Algorithm 1.

---

**Algorithm 1:** The pseudocode of the Cache Strategy process based on DQN Algorithm

---

Reset neural network net\_performance with random weights  $\theta$ .

Reset target neural network net\_target with weights  $\theta^- = \theta$ .

Reset the experience replay memory  $D_{memory}$  for the DQN with capacity C.

**For**  $epi = 1, \dots, K$  **do**

Reset the initial state vector sequence  $x(\mathbf{0})$  with the random caching distribution.

**For**  $t = 1 \dots T$  **do**

Choose an action  $a(t)$  from the action space with  $\epsilon$ -greedy policy:

Execute the selected action  $a(t)$ , calculate the immediate reward.

Store the tuple  $(x(t), a(t), R(t), x(t+1))$  in Dmemory.

Randomly sample a tuple  $(x(j), a(j), R(j), x(j+1))$

Set  $y_j = \begin{cases} R(j), & \text{if episode terminates at step } j+1 \\ R(j) + \epsilon \max Q_{net\_target}(x, a; \theta^-), & \text{otherwise} \end{cases}$

Iteratively optimize the DNN parameters by means of minimizing the loss function.

Update the neural network parameters  $Q_{net\_target} \leftarrow Q_{net\_performance}$  every S steps.

**End For**

**End For**

---

## 5 Simulation

This section conducts a number of simulations to evaluate and test the performance of the proposed DQN-based caching strategy in mobile edge networks. The numerical simulations are conducted in the simulation software MatLab 2020a. We firstly test and verify the energy efficiency improvement with the DQN training-based caching policy in a simple cellular, then we evaluate the energy efficiency of the proposed caching policy in a more general complex network scenario.

Firstly, we set a cellular network scenario that consists of an MBS, an RN and 3 end users. The cellular covers an area with radius  $R_a = 0.5km$ . We set a polar coordinate, and place the location of MBS in the center. We assume that there are 10 files cached in the network. The end users are distributed in the overall area according to the Poisson point process. The popularity distribution of all the service contents follow the zipf distribution. The basic energy consumption of an MBS is set as  $P^m = 66mW$ , while an RN assumes 10% energy of an MBS. The calculated path loss from MBSs to RNs is formulated as  $PL_{B,R} = 11.7 + 37.6lg(d)$ , the path loss from RNs to MBSs is formulated as  $PL_{R,B} = 42.1 + 27lg(d)$ , the path loss among RNs is formulated as  $PL_{R,R} = 38.5 + 27lg(d)$  and the path loss from RNs to end users is formulated as  $PL_{R,U} = 30.6 + 36.7lg(d)$ . The Q-learning algorithm parameters are set as  $\alpha = 0.2$ ,  $\gamma = 0.8$ ,  $R = 100$ . The parameters of the adapted DNN can be seen in Tab. 2.

**Table 2:** Parameters of adopted DNN

Layer	DNN1	DNN2	DNN3	DNN4
Input size	22	37	35	39
Activation	ReLU	ReLU	ReLU	ReLU
Output size	37	35	39	\

As shown in Fig. 2, we evaluate the system energy efficiency with a random caching policy, where the service contents are cached in either the MBS or RN randomly, and compare with the energy efficiency performance of the caching policy with DQN training. The red line with triangles in the figure denotes the system energy efficiency with DQN training, while the blue line with squares denotes the system energy efficiency with random caching policy. We generate user distribution ten times and each of them is used for a certain training step number. As can be seen in the figure, with enough training steps, the energy efficiency of DQN training caching policy can improve the system energy efficiency.

For a better analysis on the performance of proposed policy and its convergency, we simulate the relationship between training steps and system energy cost. It can be seen in Fig. 3 that system energy cost declines with the increasing of training steps, which means that the algorithm trends to converge with 1500 steps.

In order to test the proposed caching policy in complex network scenes, we simulate a mobile edge network with different numbers of RNs. In this part, we set the end user number is 20, the numbers of RNs are  $M = 2, 3, 4, 5, 6$ . Consider a polar coordinate situation, the MBS is located at the place  $[0,0]$ , and the  $m$ -th RN is located at the place  $[500, 2\pi m/M]$ . The Q-learning algorithm parameters in the formulation 15 are fixed.

We evaluate and compare the energy efficiency performance of the networks under three types of mobile edge networks architecture, includes a mobile edge network without caching ability, a

mobile network with random caching strategy and a mobile edge network with the proposed DQN-based caching strategy. A mobile edge network without caching ability (as shown in Fig. 4 “non cache”) firstly send the requested contents to the MBS from far cloud, then deliver the contents to corresponding end users through edge nodes. A mobile edge network with random caching strategy caches contents in edge nodes by random, under this caching strategy, a requested content has to be acquired from cloud when it is not stored in an edge node who cannot connect to corresponding end user directly.

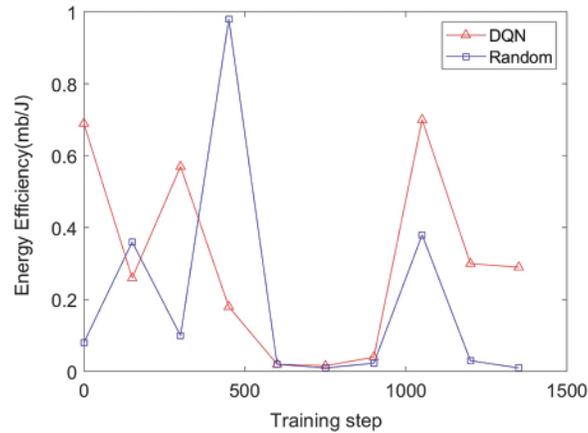


Figure 2: Comparison of the random and DQN policies

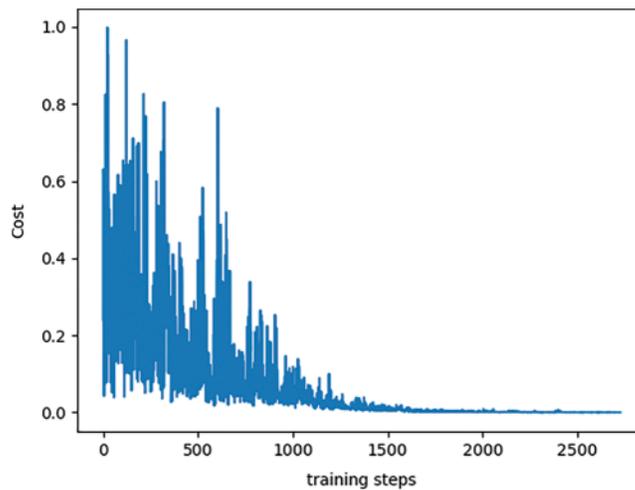


Figure 3: Cost in each training as function training numbers

The energy efficiency comparison with multi-RNs is showed in Fig. 4. By comparing the mobile edge networks energy efficiency with and without storage function, we can see that adding caching ability to mobile edge networks can significantly increase system energy efficiency. By comparing the energy efficiency of mobile edge networks with DQN-based caching strategy and random caching strategy, we can see that the proposed DQN-based caching strategy shows better energy efficiency improvement than random caching strategy when  $M > 2$ . Besides, we can also see the network energy

efficiency as a function of RN numbers, the more RNs in a mobile edge network, the better network performance the proposed caching strategy can achieve.

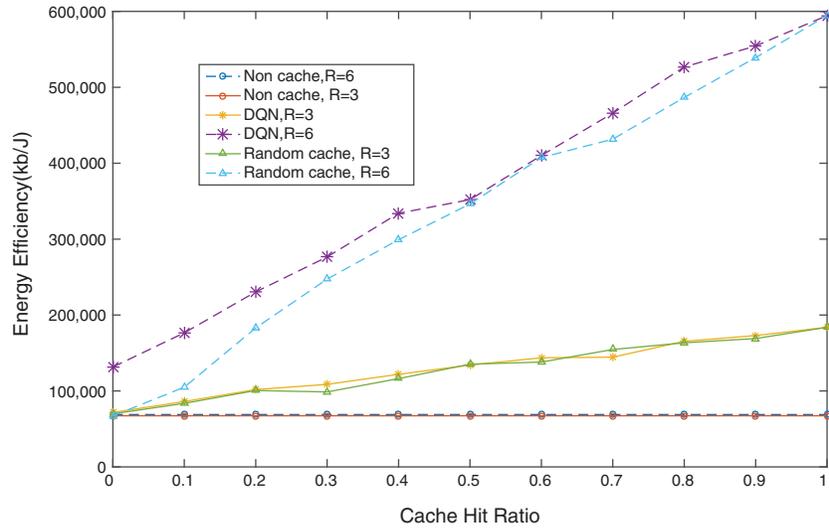


Figure 4: Energy efficiency comparison with multi-RNs

Fig. 5 shows the computation time when algorithm converges for Q-learning and DQN. Due to the enormous computation burden, general Q-learning algorithm usually costs lots of computation time, which is unacceptable in real scenes. The experimental results show that the algorithm convergence speed of the proposed DQN policy is significant faster than general Q-learning method.

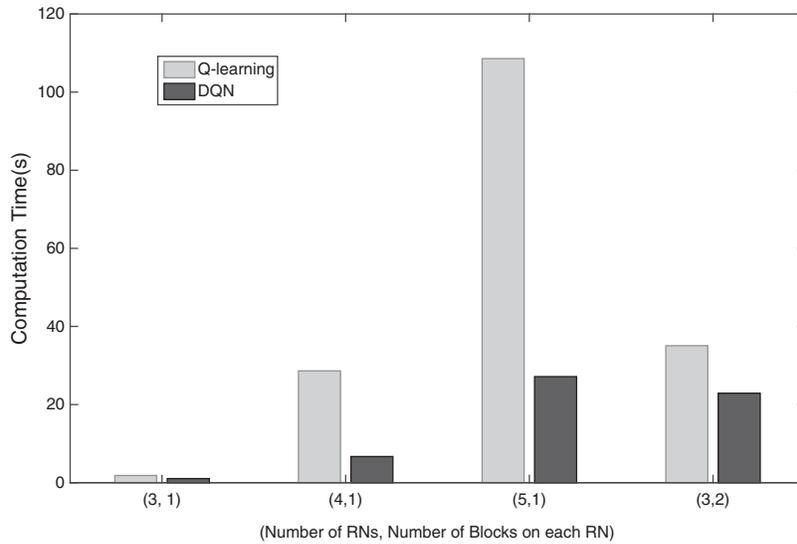


Figure 5: Computation time comparison

## 6 Conclusion

In this paper, we investigate and focus on the energy efficiency problem in mobile edge networks with caching capability. A cache-enabled mobile edge network architecture is analyzed in a network scenario with unknown content popularity. Then we formulate the energy efficiency optimization problem according to the main purpose of a content-based edge network. To address the problem, we put forward a dynamic online caching strategy using the deep reinforcement learning framework named deep-Q learning algorithm. The numerical simulation results indicate the proposed caching policy can be found quickly and it can improve the system energy efficiency performance significantly in both networks with single RN and multi-RNs. Besides, the convergence speed of the proposed DQN algorithms is significant faster than general Q-learning.

**Funding Statement:** This work was supported by the National Natural Science Foundation of China (61871058, WYF, <http://www.nsf.gov.cn/>).

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani and R. Buyya, "Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges," *IEEE Commun. Surveys Tuts*, vol. 16, no. 1, pp. 337–368, 2014.
- [2] C. Wang, Y. He, F. Yu, Q. Chen and L. Tang, "Integration of networking, caching, and computing in wireless systems: A survey, some research issues, and challenges," *IEEE Commun. Surveys Tuts*, vol. 20, no. 1, pp. 7–38, 2017.
- [3] X. Wang, M. Chen, T. Taleb, A. Ksentini and V. C. M. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, 2014.
- [4] P. Liu, S. R. Chaudhry, T. Huang, X. Wang and M. Collier, "Multi-factorial energy aware resource management in edge networks," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 1, pp. 45–56, 2019.
- [5] J. Zhang, X. Zhang and W. Wang, "Cache-enabled software defined heterogeneous networks for green and flexible 5G networks," *IEEE Access*, vol. 4, pp. 3591–3604, 2016.
- [6] C. Fang, F. R. Yu, T. Huang, J. Liu and Y. Liu, "A survey of energy-efficient caching in information-centric networking," *IEEE Communications Magazine*, vol. 52, no. 11, pp. 122–129, 2014.
- [7] A. Ioannou and S. Weber, "A survey of caching policies and forwarding mechanisms in information-centric networking," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2847–2886, 2016.
- [8] A. Sengupta, S. Amuru, R. Tandon, R. M. Buehrer and T. C. Clancy, "Learning distributed caching strategies in small cell networks," in *Proc. 11th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Barcelona, Spain, pp. 917–921, 2014.
- [9] S. Borst, V. Gupta and A. Walid, "Distributed caching algorithms for content distribution networks," in *Proc. IEEE INFOCOM*, San Diego, CA, USA, pp. 1–9, 2010.
- [10] W. Jiang, G. Feng and S. Qin, "Optimal cooperative content caching and delivery policy for heterogeneous cellular networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 5, pp. 1382–1393, 2017.
- [11] W. Liu, J. Zhang, Z. Liang, L. Peng and J. Cai, "Content popularity prediction and caching for ICN: A deep learning approach with SDN," *IEEE Access*, vol. 6, pp. 5075–5089, 2018.
- [12] R. Wang, X. Peng, J. Zhang and K. B. Letaief, "Mobility-aware caching for content-centric wireless networks: Modeling and methodology," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 77–83, 2016.
- [13] Y. Guan, Y. Xiao, H. Feng, C. -C. Shen and L. J. Cimini, "Mobicacher: mobility-aware content caching in small-cell networks," in *Proc. IEEE Global Commun. Conf.*, Austin, TX, USA, pp. 4537–4542, 2014.

- [14] C. Jarray and A. Giovanidis, "The effects of mobility on the hit performance of cached D2D networks," in *Proc. 14th Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw.*, (WiOpt), Tempe, AZ, USA, pp. 1–8, 2016.
- [15] H. Li and D. Hu, "Mobility prediction based seamless RAN-cache handover in hetNet," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Doha, Qatar, pp. 1–7, 2016.
- [16] L. Sun, Q. Yu, D. Peng, S. Subramani and X. Wang, "Fogmed: A fog-based framework for disease prognosis based medical sensor data streams," *Computers, Materials & Continua*, vol. 66, no. 1, pp. 603–619, 2021.
- [17] C. Liang, F. R. Yu and X. Zhang, "Information-centric network function virtualization over 5 g mobile wireless networks," *IEEE Network*, vol. 29, no. 3, pp. 68–74, 2015.
- [18] Y. Li and M. Chen, "Software-defined network function virtualization: A survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.
- [19] M. Karimzadeh, Z. Zhao, L. Hendriks, R. Schmidt, S. Fleur *et al.*, "Mobility and bandwidth prediction as a service in virtualized LTE systems," in *Proc. IEEE 4th Int. Conf. on Cloud Networking, CloudNet*, Niagara Falls, ON, Canada, pp. 132–138, 2015.
- [20] A. S. Gomes, B. Sousa, D. Palma, V. Fonseca, Z. Zhao *et al.*, "Edge caching with mobility prediction in virtualized LTE mobile networks," *Future Generat. Comput. Syst.*, vol. 70, pp. 148–162, 2017.
- [21] Y. Zhou, F. R. Yu, J. Chen and Y. Kuo, "Resource allocation for information-centric virtualized heterogeneous networks with in-network caching and mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 1339–1351, 2017.
- [22] Z. Tan, F. R. Yu, X. Li, H. Ji and V. C. M. Leung, "Virtual resource allocation for heterogeneous services in full duplex-enabled SCNs with mobile edge computing and caching," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1794–1808, 2018.
- [23] P. Liu, S. R. Chaudhry, X. Wang and M. Collier, "PROGRES: A programmable green router with controlled service rate," *IEEE Access*, vol. 7, pp. 3792–3804, 2019.
- [24] A. Ghiasian, P. Liu, X. Wang and M. Collier, "Energy-efficient operation of a network of openFlow switches featuring hardware acceleration and frequency scaling," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 6, pp. 613–619, 2019.
- [25] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu *et al.*, "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 393–430, 2019.
- [26] R. S. Sutton and A. G. Barto, *In Reinforcement Learning: An Introduction*, Cambridge, UK: MIT press, 1998.
- [27] Y. Wei, F. R. Yu, M. Song and Z. Han, "Joint optimization of caching, computing, and radio, resources for fog-enabled IoT using natural actor-critic deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2061–2073, 2019.
- [28] Y. Wei, F. R. Yu, M. Song and Z. Han, "User scheduling and resource allocation in hetNets with hybrid energy supply: An actor-critic reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 17, no. 1, pp. 680–692, 2018.
- [29] Z. G. Qu, S. Y. Chen and X. J. Wang, "A secure controlled quantum image steganography algorithm," *Quantum Information Processing*, vol. 19, no. 380, pp. 1–25, 2020.
- [30] Z. G. Qu, S. Y. Wu, W. J. Liu and X. J. Wang, "Analysis and improvement of steganography protocol based on bell states in noise environment," *Computers, Materials & Continua*, vol. 59, no. 2, pp. 607–624, 2019.
- [31] L. Li, Y. Wei, L. Zhang and X. Wang, "Efficient virtual resource allocation in mobile edge networks based on machine learning," *Journal of Cyber Security*, vol. 2, no. 3, pp. 141–150, 2020.
- [32] Y. Wei, Z. Wang, D. Guo and F. R. Yu, "Deep q-learning based computation offloading strategy for mobile edge computing," *Computers, Materials & Continua*, vol. 59, no. 1, pp. 89–104, 2019.
- [33] A. Amokrane, R. Langar, R. Boutaba and G. Pujolle, "Flow-based management for energy efficient campus networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 4, pp. 565–579, Dec. 2015.

- [34] K. Arulkumaran, M. P. Deisenroth, M. Brundage and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag*, vol. 34, no. 6, pp. 26–38, 2017.
- [35] W. G. Hatcher and W. Yu, "A survey of deep learning: Platforms, applications and emerging research trends," *IEEE Access*, vol. 6, pp. 411–432, 2018.
- [36] K. Toyoshima, T. Oda, M. Hirota, K. Katayama and L. Barolli L, "A DQN based mobile actor node control in WSN: Simulation results of different distributions of events considering three-dimensional environment," *Advance in Internet, Data and Web Technologies*, vol. 47, pp. 197–209, 2020.