

## Research Article

# Blockchain-Enabled Intelligent Video Caching and Transcoding in Clustered MEC Networks

Yan Li <sup>1,2</sup> and Zheng Wan <sup>1</sup>

<sup>1</sup>School of Information Management, Jiangxi University of Finance and Economics, No. 665, West Yuping Road, Nanchang, Jiangxi 330032, China

<sup>2</sup>Nanchang Institute of Technology, No. 289, Tianxiang Road, Nanchang, Jiangxi 330099, China

Correspondence should be addressed to Zheng Wan; [wanzheng97@163.com](mailto:wanzheng97@163.com)

Received 23 May 2021; Revised 15 July 2021; Accepted 16 August 2021; Published 8 September 2021

Academic Editor: Ke Gu

Copyright © 2021 Yan Li and Zheng Wan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, the number of smart devices has exploded, leading to an unprecedented increase in demand for video live and video-on-demand (VoD) services. Also, the privacy of video providers and requesters and the security of requested video data are much more threatened. In order to solve these issues, in this paper, a blockchain-enabled CMEC video transmission model (Bl-CMEC) for intelligent video caching and transcoding will be proposed to ensure the transactions' transparency, system security, user information privacy, and integrity of the video data, enhance the ability of servers in actively caching popular video content in the CMEC system, and realize transcoding function at network edge nodes. Furthermore, we chose a scheme based on deep reinforcement learning (DRL) to intelligently access the intracluster joint caching and transcoding decisions. Then, the joint video caching and transcoding decision smart contract is specially designed to automatically manage the transaction process of the joint caching and transcoding service, which records key information of joint caching and transcoding transactions and payment information on a continuous blockchain. The simulation results demonstrate that the proposed Bl-CMEC framework not only can provide users with better QoE performance for video streaming service but also can ensure the security, integrity, and consistency for the video providers, video requesters, and video data.

## 1. Introduction

People are becoming more and more dependent on network services, especially during the period of COVID-19 pandemic, many activities and works are carried out on the network. Furthermore, the most important network service is the video streaming service. In recent years, because the number of smart devices has exploded, there is an increasing demand for video live and video-on-demand (VoD) services. In video streaming services, higher data rates and larger system capacity are usually required to meet the ever-increasing users' needs, which has become a more challenging task. According to the summary of the Cisco Visual Network Index [1], mobile smart devices' videos compose around more than 50% of the total data traffic. And, it is expected to grow further to about 79% of the total data traffic in 2022.

Because of the huge demand for mobile smart devices' videos, the operators of mobile networks are not able to meet the users' demand for high quality of experience (QoE) in video live and VoD services.

In order to solve this issue, the proposal of mobile edge computing (MEC) has brought new opportunities for the optimization of wireless video transmission [2–7]. Utilizing the communication, caching, computing, and control (4C) capabilities of edge devices, it could provide proactive video caching, transcoding, and distribution services at the network edge in the mobile networks. Also, it could reduce the burden of the backbone network and improve the video quality of experience for requested users.

At the same time, smart device users may have different needs for specific videos because of the heterogeneity of smart device users' caching and computing capabilities and

changes in network conditions. For example, the smart device users with better network always ask for high QoE videos, while the smart device users with poor network generally prefer the videos with appropriate QoE. To reduce the computational load of the cloud center and transmission cost at different formats and versions of videos in the backbone network, the Content Distribution Network (CDN) is proposed which may only push a certain format and version of the video stream to the network edge and requires intelligent transcoding and distribution for tasks, as well as adaptive allocation of network resources. Especially, the cooperative transcoding decision and task assignment are needed to decide which edge nodes carry out cooperative transcoding and what kind of transcoding task is assigned to each node. The influencing factors of video transcoding and caching decisions include video content popularity, user demand and distribution, the capabilities of each edge node, and bandwidth resources between nodes. Because of this issue, adaptive bitrate streaming (ABR) [8] has been widely proposed to improve the QoE of video data which serve for smart device users in the Internet.

In addition, the blockchain has developed rapidly in recent years, which is a new fashion application mode. Its core content includes P2P transmission, encryption algorithm, distributed data storage, and consensus mechanism [9, 10], and the consensus mechanism is the most important content in blockchain. Blockchain has acted as a very effective distributed management framework which has been widely used in many fields. Through the blockchain module integrated under the framework of MEC, the data resource security protection and monitoring can be realized for the processing of video data at network edge.

In this paper, a blockchain-enabled framework for Clustered Mobile Edge Computing (CMEC) system is proposed, which can integrate the MEC networks and the CDN networks by setting the CDN tips. The experiments on comparison of QoE and bandwidth cost between CMEC and other schemes have been performed in our own previous paper [11]. The CMEC method can promote intracenter collaboration among the MEC nodes in one cluster. So, it can reduce the additional processing costs and backhaul consumption. Furthermore, the proposed blockchain-enabled CMEC-based video transmission model in this paper can seamlessly enable the blockchain scheme into our Clustered MEC network, connecting with the popular CDN video transmission system. Then, we deploy the blockchain structure into the CMEC system, which can be set at CDN tips or edge clusters. In the proposed scheme, the CDN tip and some edge nodes make the network edge area in the local network area. Then, the proposed model optimizes the entire network transmission of wireless video data by using the collaborative capabilities of edge cluster in communication, caching, computing, and control (4C). Specifically, the main contributions of this paper can be summarized as follows:

- (i) Blockchain-enabled CMEC-based video transmission model (Bl-CMEC): a video transmission system framework model with incorporating

blockchain technology is designed to actively cache popular video content in the CMEC system and realize transcoding function at network edge nodes. This model is used to improve the allocation of video caching resources and computing resources in edge cluster nodes and also to optimize user QoE from the perspective of mobile users.

- (ii) Blockchain empowerment: the joint caching and transcoding transactions between network edge node and smart device users are implemented in blockchain by a smart contract. The smart contract is specially designed to manage the transaction process of the joint caching and transcoding service, which records the joint caching and transcoding transaction and payment information on a continuous blockchain. Furthermore, the smart contract can check the integrity of results returned from the network edge node to achieve adaptive video transmission optimization and make security protection of video data in a blockchain-enabled CMEC-based environment.
- (iii) Intelligent scheme design: it can intelligently obtain and implement the decision of allocating the caching and computing resources at the network edge node in one cluster. It has two sections, namely, DQN-Based Video Caching and Transcoding Algorithm (DQN-VCT) (section 1) and Implementation of Video Caching and Transcoding Decision Smart Contract (section 2). According to mobile users' demand changes and network time-varying conditions, each network edge node determines the optimal caching and transcoding price. Then, we use the deep reinforcement learning-based algorithm to acquire the smart device users' optimal caching and transcoding decision. The decision is to meet the best QoE needs of smart device users and obtain better video services.
- (iv) Sufficient performance evaluation: because of extensive and sufficient simulations, we analyze the performance of the proposed intelligent video caching and transcoding scheme in this paper based on blockchain-enabled CMEC system environment.

This paper is organized as follows. Section 2 presents related work. The system model design is described in Section 3. Furthermore, Section 4 formulates the blockchain empowerment mode. Problem formation and intelligent video caching and transcoding scheme is introduced in Section 5. The analysis of simulation experiments is given in Section 6. Lastly, the conclusions are given in Section 7.

## 2. Related Work

In recent years, the number of smart devices has exploded, leading to an unprecedented increase in demand for video live and VoD services. Also, the privacy of video providers and requesters and the security of requested video data are much more threatened in the mobile edge computing

system. Thus, the blockchain technology can help to enable the privacy of video providers and the security of requested video data. Through the distributed storage mechanism of the blockchain, the security of video information is improved, the collection of individual video data is realized through the personal ledger of blockchain, and the authorized use of video data is realized through asymmetric encryption and public and private key design in blockchain [12–15]. It is essential to introduce the blockchain into the adaptive video services in the mobile edge computing system.

The integration of blockchain and MEC to solve the corresponding practical problems is currently a hot spot for many scholars, which is a very promising development direction [12, 16–19]. On the one hand, MEC nodes in network edge can provide a low-latency, much more convenient and distributed computing offloading scheme for smart mobile devices with only limited resources. Edge computing devices have powerful computing and storage capabilities compared to general user mobile devices, while blockchain services require powerful computing capabilities. Therefore, MEC provides the possibility for mobile users to enjoy blockchain services. On the other hand, blockchain can be used as an auxiliary framework to manage the provision of mobile edge computing resources and turn the supply of edge computing resources into a blockchain application. This not only enhances the security of MEC resources but also regulates the occupation and purchase of edge computing resources.

Based on the literature on the combination of mobile edge computing and blockchain, the recent research mainly focuses on two aspects. First, such research about security and privacy protection mainly lies in how to introduce the blockchain technology into MEC system to achieve the security and safety of cached content. Furthermore, MEC-enabled blockchain-based distributed video system architecture is used to solve the problem about the allocation of decentralized resource for video caching, transcoding, and delivery at blockchain-based video streaming system.

*2.1. Blockchain-Enabled MEC-Based Caching Strategy.* In the process of task offloading, the transmitted information is vulnerable to attacks, resulting in incomplete data. In view of this challenge, Xu et al. [20] proposed a blockchain-enabled computing offloading method which is called BeCome. In BeCome, to ensure the data integrity, they introduced the blockchain technology into the edge computing scheme. To address the issues of data security and users privacy, Feng et al. [21] adopted blockchain technology to ensure the reliability and irreversibility of cache data in the MEC network system. Also, they developed a framework of cooperative computing sharing and resource allocation for the blockchain-enabled MEC-based network system. Guo et al. [22] proposed a blockchain-enabled MEC-based framework for adaptive computing offloading and resource allocation in the future wireless networks. In this method, blockchain is used to provide management and control function. Then, the problem acted as a joint optimization issue and deep

reinforcement learning based methods are adopted to address this problem. Because of the possession of very sensitive personal information, the vehicle may be unwilling to cache its content to an untrusted cache provider. Dai et al. [23] integrated DRL and permissioned blockchain into the vehicle network, so as to acquire a secure and smart content caching method, by which a distributed content caching framework based on blockchain has been proposed. Content caching in Mobile Cyber-Physical System also faces some security issues. To address these issues, Xu et al. [24] proposed a new blockchain-based trusted network edge caching solution for mobile smart device users in a Mobile Cyber-Physical System.

Applying blockchain to MEC cache system mainly solves the safety problem of cache content, MEC servers, mobile equipment users, etc. In terms of blockchain-enabled MEC-based systems, on the one hand, such research mainly lies in the use of blockchain technology to realize the privacy protection and security of cached content. On the other hand, blockchain technology can be adopted to solve data integrity issues in edge computing and to monitor the resources of edge computing devices.

*2.2. MEC-Enabled Blockchain-Based Distributed Video Delivery Strategy.* In the blockchain-based video delivery network system, in order to meet the different needs of smart device users, a lot of computing resources are needed to transcoding them into different versions and formats for the heterogeneous quality and format of video streams. To solve this problem, Liu et al. [25] and Liu et al. [26] have proposed a MEC-enabled blockchain-based architecture using MEC technology, with a series of smart contracts that can acquire self-organization at video transcoding and delivery services, especially without a centralized controller. Some emerging video streaming platforms want to build cryptocurrency-based payment systems and p2p content distribution architectures by using the blockchain technology. Liu et al. [27] proposed a novel transcoding framework that supports the MEC network for blockchain-based video delivery scheme, designing an adaptive block size mode for the underlying blockchain. Furthermore, Zhang et al. [28, 29] proposed an incentive mechanism for blockchain-based cache and delivery systems. By this incentive mechanism, the willingness of both MEC network cache nodes and D2D can be guaranteed by meeting their expected rewards for cache sharing. The existing offloading methods based on DRL always suffer from a slow convergence which is caused by the high-dimensional action spaces. Qiu et al. [30] presented a new free-model DRL online computing offloading mode. This method is used to solve the computing offloading of data processing tasks and mining tasks in wireless blockchain networks. Effective computing diversion cannot be achieved in MEC with blockchain because mobile devices do not always have enough tokens to bear the cost of diversion services. Zhang et al. [31] analyzed the combined computing offloading and coin loan problems of blockchain-empowered MEC to optimize the total cost of all smart mobile equipment.

Such research mainly focuses on two aspects. The first aspect is to use mobile edge computing to solve the mining tasks and data processing tasks in the wireless blockchain network. In addition, in the blockchain-based video system, it works to implement the distributed resource allocation issues for video transcoding and delivery.

**2.3. Motivation.** Because of the high dynamics of the MEC network system, the data security and privacy protection of network edge service providers are a major challenge. Blockchain technology can construct a decentralized and secure resource sharing scheme, while Artificial Intelligence (AI) can explore and solve issues with time-varying, uncertain and complex characteristics [32–36]. The blockchain technology and MEC system both have the same decentralized characteristics, making their combination natural. Motivated by recent research results, blockchain technology can be introduced into the MEC system to support many management and security services in mobile edge computing. Also, DRL-based video transmission strategy in the MEC environment is extensively studied recently [37–43]. The DRL method can jointly solve the problems of cache content location decision, cache update strategy, and cache content delivery. Deep reinforcement learning is used to analyze and learn network information through deep learning, so as to use reinforcement learning to achieve resource scheduling.

Based on the related work, applying blockchain to MEC cache system mainly addresses the safety threat of the cached content, MEC servers, mobile equipment users, etc. In terms of blockchain-enabled MEC-based systems, such research only lies in the use of blockchain technology at the caching strategy, which realizes the security and privacy protection of the cached content. Furthermore, pursuing for the new distributed data management mode of computer technology, such as p2p transmission, distributed data storage, encryption algorithm, and consensus mechanism, the blockchain technology is integrated into the MEC environment architecture. Also, a blockchain-enabled distributed video content caching and transcoding framework is proposed. In this framework, edge nodes perform video content caching and transcoding and the CDN tips maintain a licensed blockchain to ensure the integrity and accuracy of cached video data, leading to design the best video caching and transcoding scheme.

### 3. System Model Design

For convenience, the major notations used in this article are summarized in Table 1.

**3.1. BI-CMEC System Model Design.** To satisfy the requirements for video distribution across the entire network and maximize the role of edge computing nodes, it is necessary to perfectly integrate edge nodes with the existing wireless network environment and video transmission technology, fully cooperating with cloud center and user terminals

[44–50]. At the same time, the blockchain technology can be integrated into the mobile edge computing environment. The blockchain can be used as an auxiliary framework to manage the provision of mobile edge computing resources and turn the supply of edge computing resources into a blockchain application.

In Figure 1, this paper intends to present a blockchain-enabled CMEC-based video transmission model. The first important aspect is the Clustered Mobile Edge Computing (CMEC) system. In this CMEC system, mobile edge computing is seamlessly connected with CDN tips. Also, the CDN tips and some mobile edge computing nodes make the mobile edge area. Based on the CMEC model, the storage, computing, and communication capabilities in one cluster can be collaboratively used to optimize wireless video streaming transmission quality through the whole network.

In this system, the CDN tip is a central server directly connected to each edge cluster. It can provide the original cached video resources for each edge cluster and the computing power support required for the implementation of deep reinforcement learning algorithms and is also responsible for maintaining the permission blockchain to ensure the transparency, security, privacy, and integrity of cached video data and user information. Also, the proposed blockchain-enabled CMEC-based video transmission model in this paper will seamlessly enable the blockchain scheme into our Clustered MEC network, combining with the popular CDN video transmission system by connecting edge clusters with CDN tips.

Furthermore, the second important aspect is the blockchain-enabled intelligent video caching and transcoding framework for CMEC-based video transmission system. In this framework, mobile edge nodes perform joint video caching and transcoding by using the deep reinforcement learning method in one mode (independent or federated) and the CDN tips will maintain a licensed blockchain to ensure the transparency, security, privacy, and integrity of cached video data and user information. The mobile edge nodes in the cluster, which can participate in the task of intelligent video caching and transcoding, will depend on the security model of the blockchain.

The security model of the blockchain includes four levels of the encryption guarantee layer, the consensus guarantee layer, the economic guarantee layer, and the social security layer. Encryption guarantee is used to ensure that only safe and legal edge nodes can participate in the task of intelligent video caching and transcoding services. The consensus mechanism layer is to complete the verification and confirmation of the transaction in a short time through the voting of trusted edge nodes to ensure the accuracy of the information. Economic security can reward good behavior (with block rewards and fees) and punish bad behavior (by cutting margin or withholding future rewards). Also, social security is the last guarantee. If the consensus attack exceeds the stage of economic security, the society can still reject it by manually lifting the control of the miner.

So, the blockchain designed in this article has four functions as follows:

TABLE 1: Summary of major notations.

Notation	Description
$t$	The time stage
$M(t)$	The serial number of MEC servers in a cluster
$U(t)$	The decision of video cache updating
$z_v$	The probability of the requests of video $v$
$C_b(t)$	The corresponding bandwidth cost
$P(n, t)$	The unit bandwidth price
$W(n, t)$	The amount of bandwidth usage in the MEC server $n$
$B(i, t)$	The bitrate assigned to user $i$ at time slot $t$
$O(t)$	The video transcoding cost
$L(t)$	The buffer occupancy rate
$R(t)$	The video rebuffering time of playback buffer
$q(t)$	The video quality rate
$CRH(t)$	The video cache hit rate
$(pk_{i,j}, sk_{i,j})$	The public/private key pairs of user mobile device $i$ connecting to edge node $j$
$(pk_j, sk_j)$	The public/private key pairs of edge node $j$
$ID(V_{i,j})$	The ID of video content $V_{i,j}$
$H(ID(V_{i,j}))$	The generated hash value of $ID(V_{i,j})$

- (a) *Transaction Transparency.* Any party in this network system (including users, mobile edge nodes, and CDN tips) can choose to download and access this blockchain to obtain information about cached and transcoded video transactions.
- (b) *System Security.* The blockchain uses asymmetric encryption technology, so that the security of the blockchain can be achieved by using private and public keys. Blockchain can provide identity verification function and access control to protect CMEC system.
- (c) *User Information Privacy.* The user privacy in the blockchain is achieved by public key anonymity. Immutability and consensus ensure the privacy of the database stored on the blockchain
- (d) *Video Data Integrity.* As users and edge nodes reach a consensus, edge nodes and users will add blocks of newly completed transactions (smart contracts) broadcasted by honest entities to the blockchain. The consensus mechanism not only ensures the integrity of the video data but also makes the content of the video data transparent in the blockchain network formed by all nodes, that is, unmodifiable means to monitor the integrity and authenticity of data.

**3.2. Video Caching Model.** In the CMEC system, to reduce the delay time of video streaming services, the more popular video content is proactively cached in the network edge node. In our CMEC system, the video content is actively cached in any edge MEC server in one edge cluster. Videos with high video popularity are generally cached at edge MEC servers, while videos with low video popularity are generally cached at CDN tips. In a cluster, the specific location of the video cached with high video popularity needs to be intelligently selected according to users' needs. In CMEC system, the video cache updating, which means the caching action at each time stage  $t$ , is denoted as  $\text{Cache}(M(t), U(t))$ .

According to this caching model, the videos are needed to be proactively cached according to the video popularity, which is the user's preference for the video. Therefore, the popularity of the video content will be requested to be obtained as the basic data information. The important key to solve the video caching problem is videos' popularity distribution. During the caching process, the video content cached by the MEC server also needs to be updated continuously. In our caching model, we update the caching video by using the first-in-first-out (FIFO) method, selecting the more popular videos at the mean time. Then, the probability of video  $v$  is defined as

$$Z_v = \frac{v^{-\alpha}}{\sum_{v=1}^V v^{-\alpha}}, \quad (1)$$

where  $\alpha > 0$  is the parameter in Zipf distribution, indicating the skewness degrees [51].

In addition, the corresponding bandwidth cost will be generated during the caching process, which is represented by the symbol  $C_b(t)$ . Bandwidth cost is one of the main costs that need to be considered. In the CMEC system, we suppose that the price of bandwidth remains constant within a time stage. Then, the bandwidth cost  $C_b(t)$  [52] of MEC servers in one cluster can be defined as follows:

$$C_b(t) = \sum_{n=1}^M P(n, t) \cdot W(n, t), \quad (2)$$

where  $M$  is the number of MEC servers in one cluster at time stage  $t$ . Also, the bandwidth cost in MEC server  $n$  can be computed by the following formula:

$$W(n, t) = \sum_{i \in U^t} B_u(i, t) \cdot I^t(i, n), \quad n \in \{0, \dots, M-1\}, \quad (3)$$

where  $q(t)$  is the user group in one cluster at time stage  $q(t) = \beta \log(B(t))$ . The symbol  $\beta$  is an indicator, representing if user  $CRH(t)$  needs be connected to the MEC server  $T$  at time stage  $t$ .

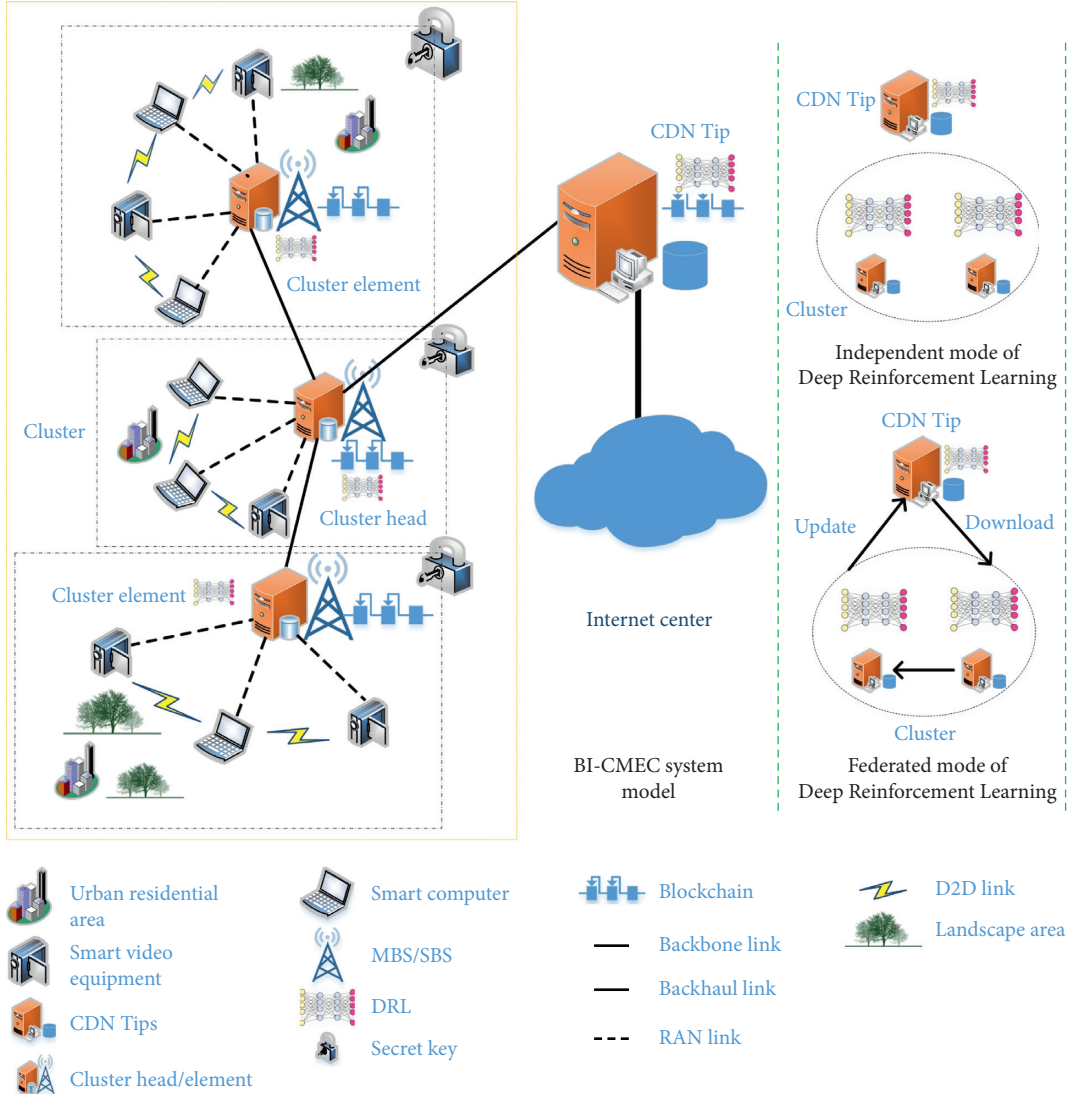


FIGURE 1: Blockchain-enabled intelligent video caching and transcoding framework for Clustered Mobile Edge Computing system.

**3.3. Video Transcoding Model.** To reduce the delay time of video services and improve the QoE of video services, proactive popular video caching has been carried out on the network edge server. However, in real life, different users have different network conditions, which leads to different users' needs for the same video. That is, users with good network conditions need high-definition video sources, while users with poor network conditions only need video sources with general definition. This requires the network edge server to cache different versions of the video source for the same video, but this requires a huge caching space to satisfy.

Therefore, it is considered that the functions of caching and transcoding can be realized at the CMEC network edge server at the same time. Thus, the CMEC network edge server can perform video transcoding in real time according to different user needs and actual conditions to satisfy the needs of different users, improving the efficiency and QoE of video services.

In our CMEC system, let  $B_u(i, t) \in \{B_1, B_2, \dots, B_{\max}\}$  be the set of all video layers in video transcoding service. Also, the symbol  $B_{\max}$  represents the original video level cached at

the MEC server. So, the video transcoding action is video transcoding layer decision and which MEC server in the cluster will carry out this task of video transcoding at each time stage  $t$ .

Because the network edge server will implement video transcoding in real time according to different user needs and actual conditions, the video services also need to consider the transcoding cost. Generally, the target video bit-rate, the input video bit-rate, the number of CPU cores, and the video length will closely affect the transcoding cost. These need to be considered in the video pricing model. Then, the video transcoding cost can be defined as

$$O(t) = \sigma * (L_{\max} - l) * T_v * N_{\text{cpu}}, \quad l \in \{L_1, L_2, \dots, L_{\max}\}, \quad (4)$$

where the symbols  $l$ ,  $N_{\text{cpu}}$ ,  $T_v$ , and  $\sigma$  represent level of input video data, the number of CPU cores required for transcoding task, the length of video, and an adjustable parameter, respectively.

**3.4. Rebuffer Model.** The video playback buffer is usually set on the user's smart device to ensure continuous playback of the video, in which the video block will be downloaded to the buffer [53]. Let  $W(t)$  represent the wireless transmission rate of smart device users. Also, the symbol  $B(t)$  denotes the bitrate of the chunk of the video data. So, the buffer occupancy rate  $L(t)$  will be obtained as

$$L(t) = \frac{\text{buffer occupancy}}{\text{buffer size}}. \quad (5)$$

Furthermore, in the rebuffer model, the buffered video time and rebuffering time could be usually introduced [54]. The video rebuffering time of playback buffer is denoted as  $R(t)$ :

$$R(t) = \max(d(t) - T(t), 0), \quad (6)$$

where  $T(t)$  denotes the buffered video time at playback buffer at the beginning of time stage  $t$ . Also, the total downloading time of one chunk during time stage  $t$  is denoted by  $d(t)$ .

**3.5. Video Quality-Rate and Cache Hit Rate Model.** Under normal circumstances, for video quality evaluation, the video quality of a rate-encoded video can be approximated by the following logarithmic function [55]:

$$q(t) = \beta \log(B(t)), \quad (7)$$

where the value  $\beta$  will be obtained from the video encoder when encoding in the video source. Furthermore, the high-definition video generally has a higher bit rate, while the standard-definition video generally has a lower bit rate.

In addition, the quality of video streaming service can generally be evaluated and analyzed using the cache hit rate. The cache hit rate of  $T$  requests in the time period [51] can be obtained as

$$CRH(t) = \frac{\sum_{i=1}^T 1(H_i)}{T}, \quad (8)$$

where  $1(H_i)$  is an indicator function.

## 4. Blockchain Empowerment Mode

**4.1. Integration of Blockchain and Video Streaming Service Network.** Blockchain is a layered architecture, which includes data layer, network layer, consensus layer, incentive layer, contract layer, and application layer. Also, it is a decentralized system composed of P2P networks. In our CMEC system, the CDN tip and some mobile edge computing nodes make the mobile edge area. Because of the large difference in capabilities among mobile edge nodes, a clustered mobile edge computing model is introduced to cluster edge nodes at the network local area. Then, there are also some mobile edge nodes in one cluster, and one cluster head is selected based on the storage, computing, and communication capabilities of the network edge node. The users can also be divided into some different user groups in one mobile edge area, according to user preferences for

videos. Generally speaking, users in the same group often have the same type of edge nodes to provide video streaming services.

As shown in Figure 2, the detail of blockchain-enabled process for intelligent video streaming service has been given as follows:

- (a) Requesting and making decision: when there is a request, the requested edge node uses the deep reinforcement learning method to make a video caching and transcoding decision based on the user's request content and the actual network environment and also returns the decision result to the user
- (b) Create smart contract: according to the decision, the video caching and transcoding smart contracts are created based on user requests and task requests in the edge cluster
- (c) Execute smart contract: the smart contract will be executed to realize the edge nodes in the edge cluster to cache the video, complete the video transcoding task, and provide the transcoded requested video to the user
- (d) Record and release the smart contract: finally, put the smart contract transaction data on the blockchain and release the smart contract

In the CMEC system, video streaming services can be implemented by P2P connection between edge nodes and user equipment. All the edge nodes and user equipment are made to be consensus nodes. Only the trusted consensus nodes are able to access CMEC video streaming service system, in which the consensus nodes should have passed authorization in blockchain. In the CMEC system, we need to consider where to cache the requested video and choose which version of the requested video should be transcoded for the users. However, in real life, different users have different network conditions, which leads to different users' needs for the same video.

Therefore, selecting the appropriate video resolution according to the user's real-time network conditions is a relatively complex decision-making problem in the high-dimensional state space.

To solve this problem, we use a framework based on deep reinforcement learning to automatically acquire intracluster collaborative caching and transcoding decisions. These decisions are executed on real time based on user demand predictions, video data popularity, and the capabilities of the MEC server. Based on the blockchain technology, we use smart contracts to implement edge collaborative caching and transcoding applications. Then, the packaging node uploads the key information of the edge collaborative cache and transcoding application to the blockchain to save the certificate and ensure efficiency and accuracy. Through the smart contract, the video streaming service transaction between the edge node and the user is realized. The user pays a certain token, and the edge node that provides the service can receive the token reward.

By blockchain-enabled framework for CMEC-based video transmission system, the blockchain technology is introduced into the CMEC system to ensure the transactions'

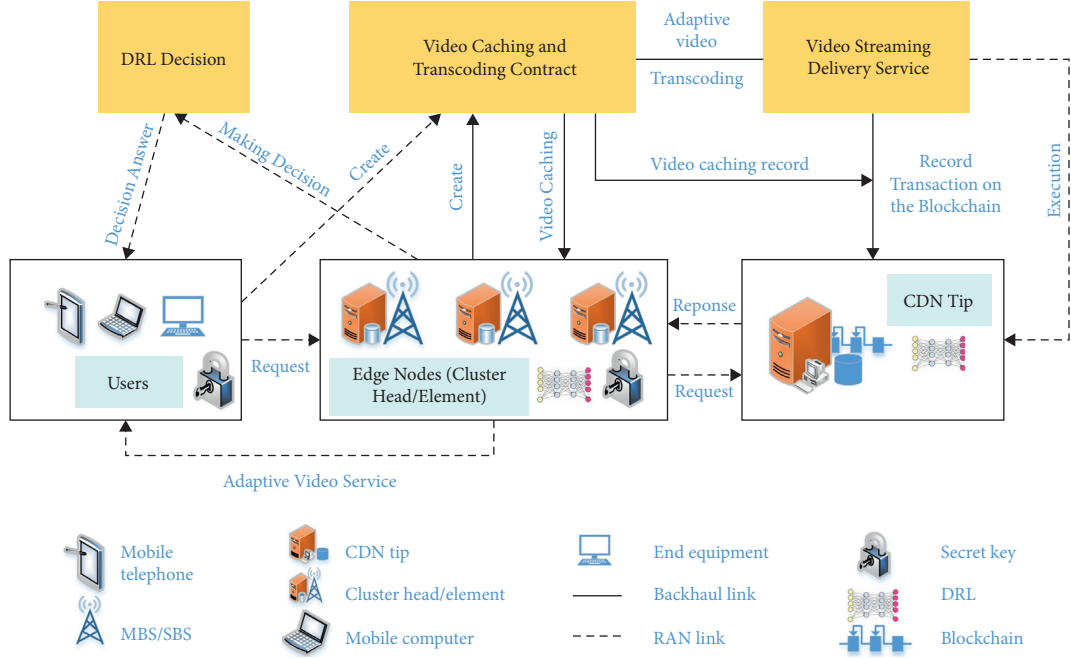


FIGURE 2: Blockchain-enabled process for intelligent video streaming service.

transparency, system security, user information privacy, and integrity of the video data. Also, the intelligent video streaming service has been provided, which has a series of smart contracts to acquire self-organization at video caching and transcoding services, especially without a centralized controller, only based on real-time DRL decision results.

In this system model, deep reinforcement learning is generally performed at the edge nodes in the cluster. Also, in blockchain-enabled CMEC-based video transmission system, the edge nodes are generally macro base station or small base station which is equipped with servers. Their computing power is strong enough to quickly implement learning algorithms and give decisions. The delay time of this process is much shorter, which can be ignored in comparison. Furthermore, the real-time computing power required for video caching and transcoding is not high. Even when the required computing power is higher, the CDN tip directly connected to the edge cluster can help to complete the task and quickly return the calculation result. Therefore, we need not consider the latency of the process of deep reinforcement learning in the experiments.

**4.2. Joint Video Caching and Transcoding Decision Smart Contract Design.** In the blockchain-enabled video streaming service, we use tokens to realize the video streaming service transaction [11]. Since the video streaming service interaction between network edge nodes and smart mobile device users is not supervised by other parties, malicious users will deliberately refuse to pay the caching and transcoding service fees of edge nodes. Based on this, first, we plan to use the blockchain technology to supervise the video caching and transcoding transactions between network edge nodes and smart mobile device users in a distributed mode. In this paper, we use blockchain to realize collaborative caching and

transcoding transactions among network edge nodes and smart mobile device users. The joint caching and transcoding transaction between network edge nodes and smart device users is implemented in blockchain by a smart contract. The smart contract is specially designed to manage the transaction process of the joint caching and transcoding service, which records the joint caching and transcoding transaction and payment information on a continuous blockchain. In particular, the smart contract can check the integrity of results returned from the network edge node to achieve adaptive video transmission optimization and make security protection of video data in the blockchain-enabled CMEC-based environment.

The detail of the joint video caching and transcoding decision smart contract construct is shown in Algorithm 1.

- (a) Initialization: the initialization is the setup function of the joint video caching and transcoding service, where the user's smart mobile device  $i$  interacts with the network edge node  $j$ . The initialization formulates smart contracts for the joint video caching and transcoding service. The user smart mobile device  $i$  and network edge node  $j$  both generate public and private key pairs, which are represented by  $(pk_{i,j}, sk_{i,j})$  and  $(pk_j, sk_j)$ , respectively. Every smart contract contains a set of variables, including cached video location  $M_{i,j}$ , cache update decision  $U_{i,j}$ , cached video content size  $S_{i,j}$ , service user video version  $B_u(i, t)$ , network edge node transcoding price  $p_j$ , deployment time  $dTime$ , timestamp  $tStamp$ , and contract service period time  $cTime$ . Finally, in this function, the user smart mobile device  $i$  and the network edge node  $j$  will both sign the contract by using their own private keys  $sign(ID_{i,j})$  and  $sign(ID_j)$ .



- (b) Create: after the user smart mobile device  $i$  and the network edge node  $j$  reach an agreement, they will deploy a new smart contract by using the creating function on the blockchain. The output of the creating function is the contract address on the blockchain, which is public to all network edge nodes and smart device users. To ensure the execution of the smart contract and prevent malicious behavior, both the user smart mobile device  $i$  and the network edge node  $j$  must submit certain deposits from their own accounts to the smart contract. The paid deposits can be represented by  $\text{deposit}_{i,j}$  and  $\text{deposit}_j$ , respectively.
- (c) Transaction execution: if a specific smart contract has been set on the blockchain, the transaction function is executed. When the user smart mobile device  $i$  requests a video streaming from the network edge node  $j$ , based on the user network environment and the load of the nodes at the cluster in CMEC system, using the deep reinforcement learning algorithm, the user mobile device  $Q(s, a; \theta)$  should generate a cooperative caching and transcoding token with the help of the edge device; its value is  $\delta_{i,j} = (\text{ID}(V_{i,j}), M_{i,j}, U_{i,j}, B_u(i, t), h_{i,j})$ , where  $\text{ID}(V_{i,j})$  is the ID of video content  $V_{i,j}$  and  $h_{i,j}$  is  $H(\text{ID}(V_{i,j}))$ , the generated hash value, where the symbol  $H$  is the public hash function. After that, the user smart mobile device signs  $\delta_{i,j}$  by  $\text{sign}(\delta_{i,j})$ . The user smart mobile device sends the signature  $\text{sign}(\delta_{i,j})$  and  $\delta_{i,j}$  to the network edge node  $j$ . Once the network edge node  $j$  receives  $\delta_{i,j}$ , it will firstly use the public key  $pk_{i,j}$  of the user smart mobile device to verify  $\delta_{i,j}$ . Simultaneously, verify  $h_{i,j}$  and implement collaborative caching and transcoding in  $\delta_{i,j}$ . After completing the above process, network edge node  $j$  generates the following joint video caching and transcoding transaction:

$$\text{Transaction} \longrightarrow (h_{i,j}, \text{sign}(\delta_{i,j}), M_{i,j}, U_{i,j}, B_u(i, t), p_j). \quad (9)$$

- (d) Recording: after the transaction is completed, the network edge node  $j$  in cluster sends the transaction to the selected smart contract. The network edge node of cluster heads and user group leaders in the network will use the DPOs consensus protocol to record transactions on the blockchain.
- (e) Penalty and settlement: at this stage, because the smart contract can monitor the video content transmission service between the user's smart mobile device and the network edge node, if any of them does not abide by the signed agreement, the penalty function will be called to implement the penalty. Finally, when the smart contract is completed and reaches the service period, financial settlement will be performed and all assets owned by the smart contract will be released.

## 5. Problem Formulation and Intelligent Video Caching and Transcoding Scheme

To use deep reinforcement learning algorithm for network resource optimization at the BI-CMEC system, we used the DQN method to address the joint video caching and transcoding optimization problem.

*5.1. Problem Formulation.* In this paper, the objective function of the joint video caching and transcoding issue is to maximize the expected average reward. Therefore, we model the dynamic optimization problem as a Markov Decision Process, which is

$$\begin{aligned} \max_{M(t), U(t), B_u(i, t)} J(t) &= E \left[ \sum_{t=0}^{T-1} \gamma^t r(t) \right], \\ \text{s.t. C1: } M(t) &\in \{0, 1, \dots, M\}, \forall t, \\ \text{C2: } U(t) &\in \{0, 1\}, \forall t, \\ \text{C3: } B_u(i, t) &\in \{B_1, B_2, \dots, B_{\max}\}, \forall t, \end{aligned} \quad (10)$$

where  $\gamma \in [0, 1]$  is a discount factor and  $r(t)$  is the reward at the time stage  $t$  in this optimization problem.

In general, it is difficult to solve optimization problems with a large number of states in the state space. However, for large-scale optimization problems that do not require any prior knowledge of state transition probability, the DRL algorithm has been proven to be a very effective mathematical tool.

*5.2. DQN for Intelligent Video Caching and Transcoding Decisions.* The basic idea of many reinforcement learning algorithms is to gradually estimate the Q value function by using the Bellman equation as an iterative update, so that

this value iterative algorithm converges to the optimal Q value function. Since the advent of the deep Q network [56–58] in 2013, many scholars have chosen to use the DQN algorithm as an algorithm and method to solve practical application problems. The value iteration algorithm to solve the optimal Q value function is as follows:

$$Q_{i+1}(s, a) = \mathbb{E} [r + \gamma \max_{a'} Q_i(s', a') | s, a]. \quad (11)$$

In DQN, Mnih et al. refer to a neural network function approximator with weights  $\theta$  as a Q-network, which can be trained by minimizing a sequence of loss functions  $L_i(\theta_i)$  that changes at each iteration  $i$ .

$$L_i(\theta_i) = \mathbb{E}_{s,a \sim \rho(\cdot)} \left[ (y_i - Q(s, a; \theta_i))^2 \right], \quad (12)$$

$$y_i = r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) | s, a,$$

where  $y_i$  is the target for iteration  $i$  and  $\rho(\cdot)$  is a probability distribution over sequences and actions which we refer to as the behavior distribution.

In deep reinforcement learning of DQN method, there are three basic elements, which are the action, state, and reward of the optimization issue. In our joint video caching and transcoding optimization issue, they can be obtained as follows:

$$\begin{aligned} S(t) &= \{C_b(t), L(t), R(t), q(t)\}, \\ A(t) &= \{M(t), U(t), B_u(i, t)\}, \\ r(t) &= \omega_1 CRH_{sl}(t) + \lambda q(t) - \omega_2 \|q(t) - q(t-1)\| \\ &\quad - \omega_3 R(t) - \omega_4 C_b(t) - \omega_5 O(t), \end{aligned} \quad (13)$$

where in the state  $S(t)$ ,  $C_b(t)$  is the current bandwidth cost,  $L(t)$  is the current buffer occupancy rate,  $R(t)$  is the current playback buffer, and  $q(t)$  is the current video quality downloaded during time stage  $t$ . Also, the action is selected from the action set  $A(t)$ . Lastly, at the reward, the weighted sum of the short- and long-term cache hit rate  $CRH_{sl}(t)$  for each step is obtained as

$$CRH_{sl}(t) = CRH_s(t) + \mu * CRH_l(t), \quad (14)$$

where  $\mu$  is the weight to balance the short- and long-term cache hit rate.

In the reward, it consists of video quality, video quality variation, video playback rebuffering time, and two penalty. Furthermore, the user perceived QoE in video streaming service is directly depended by the total cache hit rate, video quality variation, video quality, and video playback rebuffering time. Symbols  $\omega_1, \lambda_1, \omega_2, \omega_3, \omega_4$ , and  $\omega_5$  are the weighting parameters in the formula.

**5.3. Intelligent Video Caching and Transcoding Scheme.** Our proposed intelligent video caching and transcoding scheme has a series of smart contracts which can acquire self-organization at video caching and transcoding services, especially without a centralized controller. The proposed intelligent video caching and transcoding scheme has two sections, namely, DQN-Based Video Caching and Transcoding Algorithm (DQN-VCT) (section 1) and Implementation of Video Caching and Transcoding Decision Smart Contract (section 2).

In section 1, there are two major factors that support DQN and make it extremely powerful. The two major factors are experience replay and fixed Q-targets. Through these two factors, the correlation between the learning samples is removed, and the learning efficiency of DQN is getting higher and higher.

In DQN-Based Video Caching and Transcoding Algorithm (DQN-VCT), the inputs of the deep neural network are the video service system states  $S(t)$ , which are listed in equation (13), and the outputs of the network are the Q value

function,  $Q(s, a; \theta)$  for each action are listed in equation (11). Based the method in our previous article [59], we illustrate the details of the DQN-based video caching and transcoding algorithm in section 1 in Algorithm 2.

Then, based on the decision of video caching and transcoding in section 1 by the algorithm, the selected edge node in cluster will execute automatically the implementation of video caching and transcoding decision smart contract. The smart contract will be strictly executed according to Algorithm 1.

## 6. Simulation and Analysis

This section contains two parts. First, the experiment settings were illustrated. Then, the experimental simulations were carried out to prove the performance of the proposed scheme.

### 6.1. Experimental Settings

**6.1.1. Data Generation.** In the experiments, the smart device user data of requests will be generated randomly. The video data of smart device users' requests were generated under the Zipf function distribution. Different numbers of requests in one episode have been adopted as the testing data, such as 50, 70, and 100. The video data in smart device users' different numbers of requests were obtained by unchanged popularity distribution, in which the Zipf function parameter is set as 1.3.

**6.1.2. Parameters Setting.** In the experiments, we deploy 7 MEC network nodes in one cluster, which will serve 30 smart device users in this region and also provide about 50 videos for smart device users' requests. There are four video layers of the video in the experiment, with the original layer at the MEC node in the cluster as  $B_{\max}$ . The video transcoding from  $B_{\max}$  to  $B_1, B_2$ , and  $B_3$  will need, respectively, 2, 4, and 6 CPU cycles. Then, we set the parameter in the experiments as given in Table 2.

**6.1.3. Deep Neural Network Setting for DQN.** In the experiments, a fully connected neural network was adopted, which consists of 2 hidden layers, 256 and 512 in size, respectively. The loss function we used was the mean square error function. The naive  $\epsilon$ -greedy strategy was adopted for exploration, in which the probability of randomly choosing an action during the training stage was  $\epsilon$ . The degree of exploration continues to shrink when the learning progresses. The size of experience replay in DQN and the learning rate were adopted as 2000 and 0.01. Also, the number 0.90 was chosen as the attenuation parameter which is used to update the target Q network. Then, the batch size in stochastic batch gradient descent was 32. Finally, the experiment simulations were carried out by using Python.

**6.1.4. Environment Setting for Blockchain.** To assist the simulation experiment, in the simulation implementation of

- (1) **Initialization:**
- (2) Initialize the input data  $ID_{i,j}, ID_j, M_{i,j}, U_{i,j}, S_{i,j}, B_u(i, t), p_j$
- (3) Initialize state  $\{(pk_{i,j}, sk_{i,j}), (pk_j, sk_j), dTime, tStamp, cTime\}$
- (4) sign  $(ID_{i,j})$  and sign  $(ID_j)$  on the selected smart contract
- (5) **Creat:**
- (6) Output the smart contract address
- (7) Input:  $deposit_{i,j}, deposit_j$
- (8) Verify:  $deposit_{i,j} \geq p_j, deposit_j$
- (9) **Transaction Execution:**
- (10) Verify the state:  $t > dTime$
- (11) Edge cluster asks the video for caching
- (12) The selected edge node in edge cluster implements the transcoding task
- (13) Edge cluster sends the appreciate video to users
- (14) Edge cluster broadcasts collaborative caching and transcoding transaction
- (15) Transaction as:
- (16) Transaction  $\longrightarrow (h_{i,j}, \text{sign}(\delta_{i,j}), M_{i,j}, U_{i,j}, B_u(i, t), p_j)$
- (17) **Recording:**
- (18) Edge node in cluster sends the transaction to the selected smart contract
- (19) Edge cluster and users in the network use the DPoS consensus protocol to record transactions on the blockchain
- (20) **Penalty and Settlement:**
- (21) Verify the state:  $t > cTime$
- (22) Penalty execution:  $penalty_{i,j}, penalty_j$
- (23) Settlement:
- (24)  $(ID_{i,j}, deposit_{i,j} - p_j + penalty_{i,j}), (ID_j, deposit_j + p_j + penalty_j)$

ALGORITHM 1: Joint video caching and transcoding decision smart contract algorithm.

the blockchain, we use Ganache to simulate the operating environment of Ethereum and deploy smart contracts through Truffle. When using Truffle to deploy a smart contract, the network address and network number of the current simulation environment will be used so that Truffle can deploy the smart contract to Ganache's Ethereum test environment through this interface.

**6.2. Experimental Simulation Results.** In this section, we perform intelligent video caching and transcoding by deep reinforcement learning method in the independent mode. Also, we compare the proposed BI-CMEC scheme (called BI-CMEC method) with the CMEC scheme without blockchain technology (called CMEC method). Because of the characteristics of DRL, for the proposed algorithm and compared method, all the reported results would be acquired from the average of 20 algorithm executions.

Figure 3 shows the convergence performance of DQN-VCT algorithm, which is with the set of full weight at different learning rates. From Figure 3, we can see that the performance of learning rate 0.01 is the best among the three different learning rates. It is better than the performance with learning rate 0.1 and 0.001. Because a large update step will lead the average reward converging to a local optimal solution, convergence performance in learning rate 0.1 becomes worse. Generally, the appropriate learning rate always depends on the real-time state of the environment at the current optimization step.

Figure 4 shows the comparison of the QoE value on video streaming service performance with and without blockchain empowerment. It can be seen from Figure 4 that

the QoE value of the BI-CMEC method empowered by the blockchain is slightly worse than that of the CMEC method. This is because the blockchain empowerment introduces a consensus mechanism, which causes a certain time delay, reducing the QoE value of video streaming. Comparing to the CMEC method, blockchain empowerment brings security and privacy protection to video streaming service systems based on edge computing in the BI-CMEC method. Based on blockchain-enabled framework for CMEC-based video transmission system, the blockchain technology is introduced into CMEC system to ensure the transactions' transparency, system security, user information privacy, and integrity of the video data.

It can be seen from Figure 5 that the bandwidth cost performance between the BI-CMEC method and CMEC method is much similar. At the beginning, the service cost of the two methods is relatively high, but as the learning process continues to advance, the service cost slowly decreases. Although the BI-CMEC method enabled by blockchain has a higher service cost than the CMEC method in the later stage, overall, the service cost of the BI-CMEC method enabled by the blockchain is very similar to that of the CMEC method. This is because there is no other cost load that is introduced in the BI-CMEC method, except time delay comparing to the CMEC method.

In order to better analyze the experimental effect, we give a comparison of the bandwidth cost and the average QoE when the blockchain is placed in different locations. From Figure 6, we can see that when the blockchain is set in the CDN tip, the bandwidth cost is slightly higher than that of the other two locations, namely, cluster head and cluster element. This is because when many clusters' blockchains are

(1) <b>Section 1:</b> DQN-Based Video Caching and Transcoding Algorithm (DQN-VCT)
(2) <b>Initialization:</b>
(3) Initialize the replay memory $D$ to capacity $N$
(4) Initialize the Q network and the target Q network with random weights
(5) Initialize MEC network service matrix $V$ of requests
(6) <b>for</b> episode = 1, $M$ <b>do</b>
(7) Generate the smart device users' requests data
(8) Observe initial state $s_1$ as illustrated in equation (13)
(9) <b>for</b> $t = 1, T$ <b>do</b>
(10) Give a random probability $\zeta \in [0, 1]$
(11) Choose action $A(t)$ which listed in equation (13) as $A(t) = \begin{cases} a^*(t) = \arg \max_a Q(s, a; \theta), & \zeta > \varepsilon, \\ a(t) \neq a^*(t), \text{ randomly select } a(t), & \text{others.} \end{cases}$
(12) Observe the reward $r(t)$ , state $s(t+1)$
(13) Store the transition $(s(t), A(t), r(t), s(t+1))$ into Buffer pool $D$
(14) Update MEC network service matrix $V$ of requests
(15) Sample random minibatch of transitions $(s(t), A(t), r(t), s(t+1))$ from Buffer pool $D$
(16) Set $y_j = \begin{cases} r_j, & \text{for terminal } s', \\ r_j + \gamma \max_{a'} Q(s', a'; \theta_{i-1})   s, a, & \text{o non-terminal } s'. \end{cases}$
(17) Implement a gradient descent step according to equation (12)
(18) Update the parameters within the Q network
(19) Reset the parameters within the target Q network every $G$ time stages
(20) <b>end for</b>
(21) <b>end for</b>
(22) <b>Section 2:</b> Implementation of Video Caching and Transcoding Decision Smart Contract
(23) <b>Create SC:</b>
(24) Trigger the smart contract according to the user's request based on the action $A(t)$ from Section 1
(25) <b>Execute the SC:</b> carry out the smart contract for managing the transaction process of the intelligent caching and transcoding service
(26) <b>Record the SC:</b> record transactions on the blockchain
(27) <b>Release the SC:</b> the smart contract is released.

ALGORITHM 2: Intelligent video caching and transcoding scheme.

TABLE 2: Summary of major parameter values.

Parameter	Value	Parameter	Value
$D$	10 s	$\omega_3$	0.1
$\beta$	6.5	$\omega_4$	0.1
$\alpha$	1.3	$\omega_5$	0.1
$\mu$	0.6	$B_{\max}$	10 Mbps
$\sigma$	1.2	$B_1$	1 Mbps
$\omega_1$	1.2	$B_2$	2 Mbps
$\lambda$	1.2	$B_3$	4 Mbps
$\omega_2$	0.9	CPU cores	{2, 4, 6, 8}

set in the CDN tip compared to the cluster head and cluster element, it will cause more bandwidth consumption. Then, it can be easily seen in Figure 7 that as the learning process continues, the position of the blockchain has less impact on the video QoE and tends to be similar.

Then, we analyzed the experimental results when there are different numbers of user requests in an episode. In Figure 8, overall, in order to pursue higher video quality at the beginning, the bandwidth cost is higher. When the number of smart device users' requests in a time slot is 100, the bandwidth cost generated is the largest. With continuous learning, the bandwidth cost is slowly reduced and the balance between bandwidth cost and video QoE is desired. Similarly, it can be directly seen in Figure 9 that different numbers of user requests in an episode have almost no effect on the QoE value. Based on the above experimental results,

we can find that the BI-CMEC method we proposed has better robustness to the Internet environment.

Furthermore, the framework and algorithm of this article can withstand the attacks that blockchain technology can withstand, such as encryption cracking, consensus mechanism challenges, 51% attacks, and N@S (nothing at stock) attacks. The discussion of these attacks is well documented in the blockchain theory and technology related literature. Also, the topic of this article is blockchain-enabled intelligent video caching and transcoding in clustered MEC networks, which integrates blockchain technology and applies it into video streaming services based on edge computing to ensure user information privacy and video data security through the distributed storage structure of blockchain. The core point in our paper is how to effectively integrate blockchain technology into intelligent

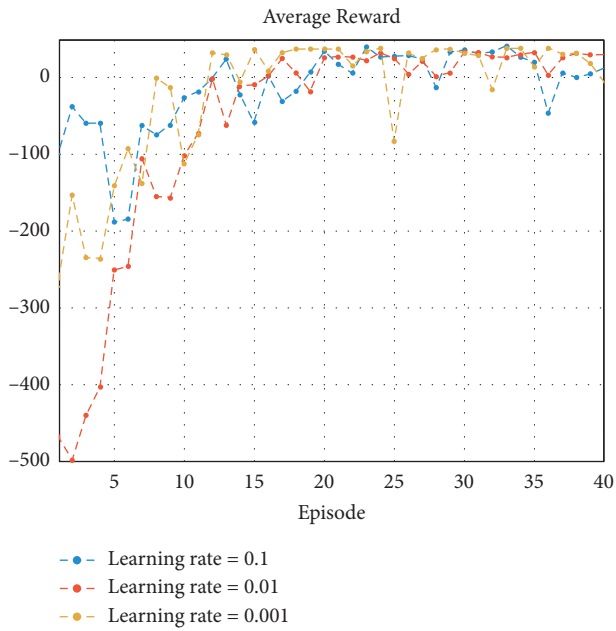


FIGURE 3: The convergence performance of DQN-based joint video caching and transcoding algorithm with the different learning rates.

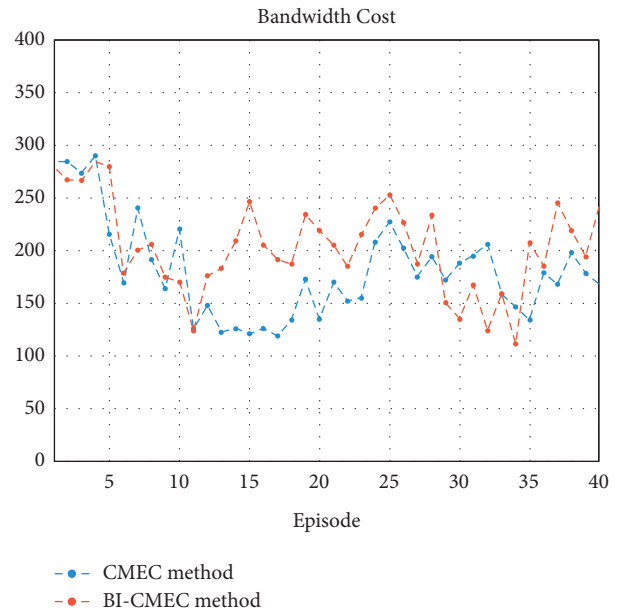


FIGURE 5: The bandwidth cost performance between the BI-CMEC method and CMEC method with blockchain at CDN tip and 50 requests in an episode.

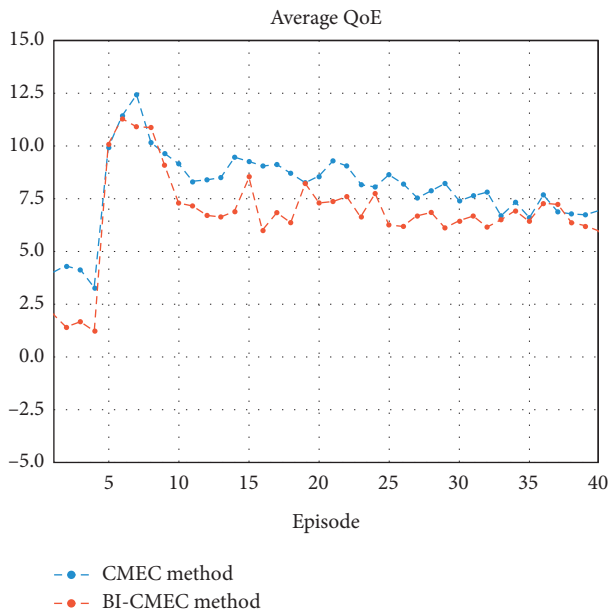


FIGURE 4: The QoE performance between the BI-CMEC method and CMEC method with blockchain at CDN tip and 50 requests in an episode.

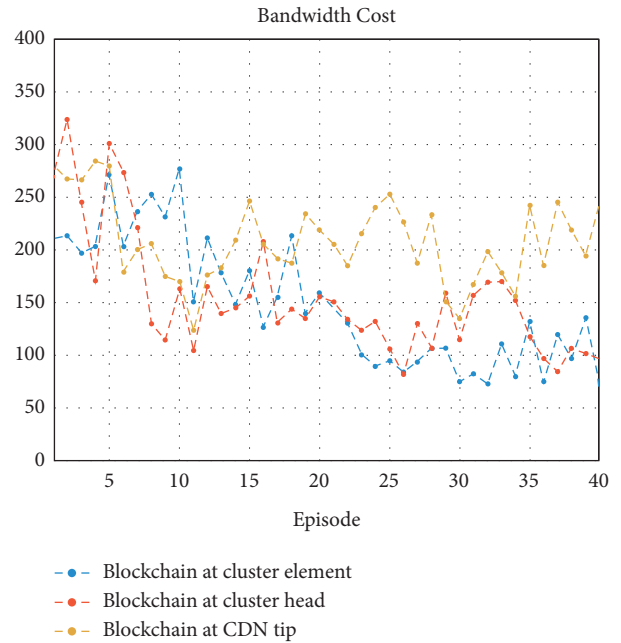


FIGURE 6: The bandwidth cost performance in the BI-CMEC method with blockchain at different positions.

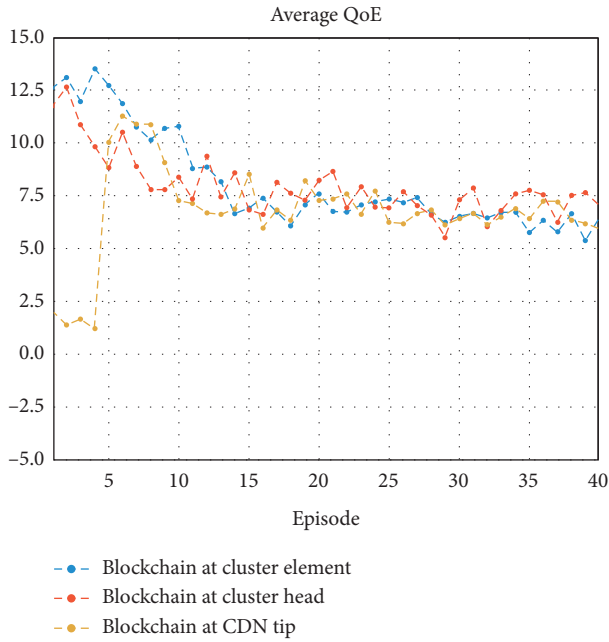


FIGURE 7: The QoE performance in the BI-CMEC method with blockchain at different positions.

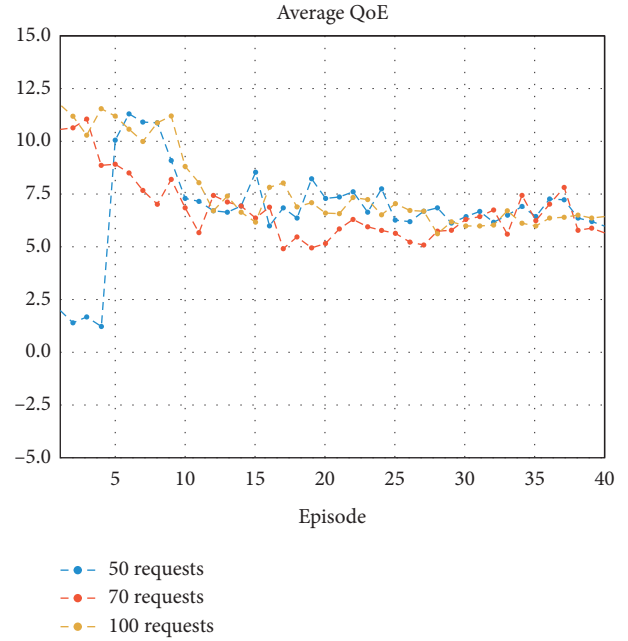


FIGURE 9: The QoE performance of the BI-CMEC method in different requests' numbers at an episode.

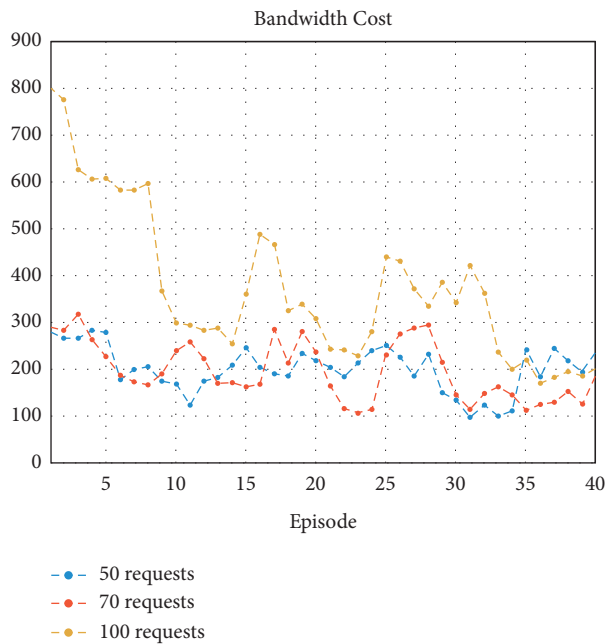


FIGURE 8: The bandwidth cost performance of the BI-CMEC method in different requests' numbers at an episode.

video services in the clustered edge computing environment, on the basis of improving security performance and ensuring the high quality of intelligent video services. Therefore, the discussion of these attacks will not be repeated here.

### 7. Conclusions

In this paper, we firstly proposed a blockchain-enabled CMEC-based video transmission system model (BI-CMEC) that could ensure the transactions' transparency, system security, user information privacy, and integrity of the video data, enhance the ability of servers in actively caching popular video content in the CMEC system, and realize transcoding function at network edge nodes. In addition, we proposed an intelligent video caching and transcoding scheme. A smart contract is specially designed which can acquire self-organization at video caching and transcoding services, especially without a centralized controller. Furthermore, we adopted a DQN-based framework to automatically obtain the intra-cluster joint video caching and transcoding decisions. Finally, the experimental results were presented to validate the effectiveness of the proposed method.

Based on the model of BI-CMEC system, this paper mainly focuses on ensuring video data security and user privacy protection and also encouraging the collaboration among MEC network nodes in one cluster. In this model, the DPoS consensus protocol was used in video transmission application scenarios. In the future work, we will perform intelligent video caching and transcoding by using the deep reinforcement learning method in the federated mode, and more efficient consensus mechanism algorithms need to be proposed to meet the special needs of video transmission application scenarios.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 61961021 and 52165015), Science and Technology Project of Jiangxi Education Department (Nos. GJJ180251 and GJJ171011), and Innovation Special Fund for Individual Graduate Student of Jiangxi University of Finance and Economics (2020 Annual, No. 24).

## References

- [1] Cisco, *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022 White Paper*, 2019, <https://www.cisco.com/c/en/us/solutions/collateral/serviceprovider/visual-networking-index-vni/white-paper-c11-738429.html>.
- [2] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, “Mobile edge computing—a key technology towards 5G,” *ETSI White Paper*, vol. 11, 2015.
- [3] K. Zhang, Y. Mao, S. Leng et al., “Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks—efficient offloading for mobile edge computing in 5G heterogeneous networks,” *IEEE Access*, vol. 4, pp. 5896–5907, 2016.
- [4] A. Ahmed and E. Ahmed, “A survey on mobile edge computing,” in *Proceedings of the IEEE International Conference on Intelligent Systems and Control (ISCO)*, Coimbatore, India, 2016.
- [5] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, “Delay-optimal computation task scheduling for mobile-edge computing systems,” in *Proceedings of the 2016 IEEE International Symposium on Information Theory (ISIT)*, pp. 1451–1455, Barcelona, Spain, July 2016.
- [6] Y. Mao, J. Zhang, and K. B. Letaief, “Dynamic computation offloading for mobile-edge computing with energy harvesting devices,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.
- [7] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, “Collaborative mobile edge computing in 5G networks: new paradigms, scenarios, and challenges,” *IEEE Communications Magazine*, vol. 55, no. 4, pp. 54–61, 2017.
- [8] D. Wang, Y. Peng, X. Ma et al., “Adaptive wireless video streaming based on edge computing: opportunities and approaches,” *IEEE Transactions on Services Computing*, no. 99, p. 1, 2018.
- [9] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, “When mobile blockchain meets edge computing,” *IEEE Communications Magazine*, vol. 56, pp. 8–39, 2018.
- [10] Y. Liu, “2020 Index IEEE communications Surveys&Tutorials vol. 22,” *IEEE Communications Surveys and Tutorials*, vol. 22, no. 4, pp. 1–16, 2020.
- [11] N. Barman, G. C. Deepak, and M. G. Martini, “Blockchain for video streaming: opportunities, challenges and open issues,” *Computer*, vol. 53, p. 7, 2020.
- [12] D. C. Nguyen, P. C. Pathirana, N. Ding, and A. Seneviratne, “Blockchain for 5G and beyond networks: a state of the art survey,” *Journal of Network and Computer Applications*, vol. 166, Article ID 102693, 2020.
- [13] X. Wang, C. Wang, X. Li, V. C. M. Leung, and T. Taleb, “Federated deep reinforcement learning for internet of things with decentralized cooperative edge caching,” *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9441–9455, 2020.
- [14] X. Jiang, F. R. Yu, T. Song, and V. C. M. Leung, “Resource allocation of video streaming over vehicular networks: a survey, some research issues and challenges,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 99, pp. 1–21, 2021.
- [15] Y. Wei, S. Zhou, S. Leng, S. Maharjan, and Y. Zhang, “Federated learning empowered end-edge-cloud cooperation for 5G HetNet security,” *IEEE Network*, vol. 35, no. 2, pp. 88–94, 2021.
- [16] Y. Pan, N. Xiong, and J. Ren, “Data security and privacy protection for cloud storage: a survey,” *IEEE Access*, vol. 8, pp. 131723–131740, 2020.
- [17] N. Zhang, C. Wu, Y. Wu, and N. N. Xiong, “An improved target tracking algorithm and its application in intelligent video surveillance system,” *Multimedia Tools & Applications*, vol. 79, no. 23–24, pp. 15965–15983, 2020.
- [18] Y. Long, Y. Chen, W. Ren, H. Dou, and N. N. Xiong, “DePET: a decentralized privacy-preserving energy trading scheme for vehicular energy network via blockchain and K-anonymity,” *IEEE Access*, vol. 8, pp. 192587–192596, 2020.
- [19] J. Wan, P. Zheng, H. Si, N. N. Xiong, W. Zhang, and A. V. Vasilakos, “An artificial intelligence driven multi-feature extraction scheme for big data detection,” *IEEE Access*, vol. 7, pp. 80122–80132, 2019.
- [20] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, and W. Dou, “Be-Come: blockchain-enabled computation offloading for IoT in mobile edge computing,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4187–4195, 2020.
- [21] J. Feng, F. Richard Yu, Q. Pei, X. Chu, J. Du, and L. Zhu, “Cooperative computation offloading and resource allocation for blockchain-enabled mobile-edge computing: a deep reinforcement learning approach,” *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6214–6228, 2020.
- [22] F. Guo, F. R. Yu, H. Zhang, H. Ji, M. Liu, and V. C. M. Leung, “Adaptive resource allocation in future wireless networks with blockchain and mobile edge computing,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 1689–1703, 2020.
- [23] Y. Dai, D. Xu, and K. Zhang, “Deep reinforcement learning and permissioned blockchain for content caching in vehicular edge computing and networks,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4312–4324, 2020.
- [24] Q. Xu, Z. Su, and Q. Yang, “Blockchain-based trustworthy edge caching scheme for mobile cyber-physical system,” *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1098–1110, 2020.
- [25] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, “Distributed resource allocation in blockchain-based video

- streaming systems with mobile edge computing,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 695–708, 2019.
- [26] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. M. Leung, “Decentralized resource allocation for video transcoding and delivery in blockchain-based system with mobile edge computing,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11169–11185, 2019.
- [27] M. Liu, Y. Teng, F. Yu, V. Leung, and M. Song, “A mobile edge computing (MEC)-Enabled transcoding framework for blockchain-based video streaming,” *IEEE Wireless Communications*, vol. 99, pp. 1–7, 2020.
- [28] R. Zhang, F. R. Yu, J. Liu, T. Huang, and Y. Liu, “Deep reinforcement learning (DRL)-based device-to-device (D2D) caching with blockchain and mobile edge computing,” *IEEE Transactions on Wireless Communications*, no. 99, 2020.
- [29] R. Zhang, F. R. Yu, J. Liu, R. Xie, and T. Huang, “Blockchain-incentivized D2D and mobile edge caching: a deep reinforcement learning approach,” *IEEE Network*, vol. 34, no. 4, pp. 150–157, 2020.
- [30] X. Qiu, L. Liu, W. Chen, Z. Hong, and Z. Zheng, “Online deep reinforcement learning for computation offloading in blockchain-empowered mobile edge computing,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 8050–8062, 2019.
- [31] Z. Zhang, Z. Hong, W. Chen, Z. Zheng, and X. Chen, “Joint computation offloading and coin loaning for blockchain-empowered mobile-edge computing,” *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9934–9950, 2019.
- [32] H. Cheng, Z. Xie, L. Wu, Z. Yu, and R. Li, “Data prediction model in wireless sensor networks based on bidirectional LSTM,” *EURASIP Journal on Wireless Communications and Networking*, pp. 1–12, 2019.
- [33] Y. Yang, N. Xiong, N. Y. Chong, and X. Défago, “A decentralized and adaptive flocking algorithm for autonomous mobile robots,” in *Proceedings of the 3rd International Conference on Grid and Pervasive Computing*, Kuming, China, May 2008.
- [34] Q. Zhang, C. Zhou, N. Xiong, Y. Qin, X. Li, and S. Huang, “Multimodel-based incident prediction and risk assessment in dynamic cybersecurity protection for industrial control systems,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 10, pp. 1429–1444, 2015.
- [35] K. Huang, Q. Zhang, C. Zhou, N. Xiong, and Y. Qin, “An efficient intrusion detection approach for visual sensor networks based on traffic pattern learning,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 10, pp. 2704–2713, 2017.
- [36] A. Shahzad, M. Lee, Y.-K. Lee et al., “Real time MODBUS transmissions and cryptography security designs and enhancements of protocol sensitive information,” *Symmetry*, vol. 7, no. 3, pp. 1176–1210, 2015.
- [37] W. Wu, Y. Gao, T. Zhou et al., “Deep reinforcement learning-based video quality selection and radio bearer control for mobile edge computing supported short video applications,” *IEEE Access*, vol. 7, pp. 181740–181749, 2019.
- [38] L. Lei, X. Xiong, L. Hou, and K. Zheng, “Collaborative edge caching through service function chaining: architecture and challenges,” *IEEE Wireless Communications*, vol. 25, no. 3, pp. 94–102, 2018.
- [39] C. H. Wei, Y. W. Hung, and F. L. Chin, “Q-learning based collaborative cache allocation in mobile edge computing,” *Future Generation Computer Systems*, vol. 102, pp. 603–610, 2020.
- [40] Z. Yang, Y. Liu, Y. Chen, and G. Tyson, “Deep reinforcement learning in cache-aided MEC networks,” in *Proceedings of the ICC 2019-2019 IEEE International Conference on Communications (ICC)*, IEEE, Shanghai, China, 2019.
- [41] C. Zhong, M. C. Gursoy, and S. Velipasalar, “Deep reinforcement learning based edge caching in wireless networks,” *IEEE Transactions on Cognitive Communications and Networking*, 2020.
- [42] M. C. Gursoy, C. Zhong, and S. Velipasalar, “Deep magnet reinforcement learning for cooperative edge caching,” *Machine Learning for Future Wireless Communications*, pp. 439–457, 2020.
- [43] L. Liu, H. Hu, Y. Luo, and Y. Wen, “When wireless video streaming meets AI: a deep learning approach,” *IEEE Wireless Communications*, vol. 27, 2019.
- [44] H. Cheng, Z. Su, N. Xiong, and Y. Xiao, “Energy-efficient node scheduling algorithms for wireless sensor networks using Markov Random Field model,” *Information Sciences*, vol. 329, no. 2, pp. 461–477, 2016.
- [45] H. Cheng, N. Xiong, A. V. Vasilakos, L. Tianruo Yang, G. Chen, and X. Zhuang, “Nodes organization for channel assignment with topology preservation in multi-radio wireless mesh networks,” *Ad Hoc Networks*, vol. 10, no. 5, pp. 760–773, 2012.
- [46] W. Wu, N. Xiong, and C. Wu, “Improved clustering algorithm based on energy consumption in wireless sensor networks,” *IET Networks*, vol. 6, no. 3, pp. 47–53, 2017.
- [47] J. Wang, Y. Ding, N. N. Xiong, W.-C. Yeh, and J. Wang, “GSCS: general secure consensus scheme for decentralized blockchain systems,” *IEEE Access*, vol. 8, pp. 125826–125848, 2020.
- [48] H. Cheng, Z. Xie, Y. Shi, and N. Xiong, “Multi-step data prediction in wireless sensor networks based on one-dimensional CNN and bidirectional LSTM,” *IEEE Access*, vol. 7, pp. 117883–117896, 2019.
- [49] H. Cheng, L. Wu, Y. Zhang, and N. Xiong, “Data recovery in wireless sensor networks using Markov random field model,” *ICACI*, pp. 706–711, 2018.
- [50] N. Xiong, X. Huang, H. Cheng, and Z. Wan, “Energy-efficient algorithm for broadcasting in ad hoc wireless sensor networks,” *Sensors*, vol. 13, no. 4, pp. 4922–4946, 2013.
- [51] C. Zhong, M. C. Gursoy, and S. Velipasalar, “A deep reinforcement learning-based framework for content caching,” in *Proceedings of the 2018 52nd Annual Conference on Information Sciences and Systems (CISS)*, March 2018.
- [52] Y. Zheng, D. Wu, Y. Ke, C. Yang, M. Chen, and G. Zhang, “Online cloud transcoding and distribution for crowdsourced live game video streaming,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 8, pp. 1777–1789, 2017.
- [53] T.-Y. Huang, R. Johari, N. Mckeown, M. Trunnell, and M. Watson, “A buffer-based approach to rate adaptation,” *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 187–198, 2015.
- [54] Y. Guo, F. R. Yu, J. An, K. Yang, G. Yu, and V. C. M. Leung, “Adaptive bitrate streaming in wireless networks with transcoding at network edge using deep reinforcement learning,” *IEEE Transactions on Vehicular Technology*, vol. 99, p. 1, 2020.
- [55] M. Chen, M. Ponc, S. Sengupta, J. Li, and P. A. Chou, “Utility maximization in peer-to-peer systems with applications to video conferencing,” *IEEE/ACM Transactions on Networking*, vol. 20, no. 6, pp. 1681–1694, 2012.
- [56] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, USA, 2018.



- [57] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Playing Atari with deep reinforcement learning," in *Proceedings of the NIPS Deep Learning Workshop*, Lake Tahoe, NV, USA, 2013.
- [58] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [59] W. Zheng and Y. Li, "Deep reinforcement learning-based collaborative video caching and transcoding in clustered and intelligent edge B5G networks," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 6684293, 16 pages, 2020.