

Research Article

Quality Evaluation of Online Mental Health Education Based on Reinforcement Learning in the Pandemic

Weifeng Zhang 

Xinxiang University, No. 191, Jinsui Road, Xinxiang City, Henan Province 453003, China

Correspondence should be addressed to Weifeng Zhang; zhangweifeng02@xxu.edu.cn

Received 8 November 2021; Revised 3 December 2021; Accepted 6 December 2021; Published 17 December 2021

Academic Editor: Lianbo Ma

Copyright © 2021 Weifeng Zhang. 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.

The COVID-19 pandemic has become one of the biggest major health crises reported due its massive impact on many countries. From mental health experts, we know that we cannot lose sight of an equally alarming issue which is the long-term mental health impact the pandemic is going to leave on the society. The rapid spread of the pandemic gives little chance to prepare for or even process all that has happened in terms of job losses and the complete uprooting of everyday life and relationships. It is understandable that students may feel irritable, frustrated, or sad sometimes. Loneliness, confusion, and anxiety are also common, but the issue is how we can know if students' emotions are a normal reaction to an abnormal situation. Therefore, online mental health education has become pretty important for students during the pandemic. Furthermore, it is important to evaluate the quality of online mental health education through microlessons. In this paper, based on Q-learning algorithm, the real-time adaptive bitrate (ABR) configuration parameters mechanism is proposed to detect the changes of network state constantly and select the optimal precalculated configuration according to the current network state. The simulation results show that the proposed algorithm based on Q-learning outperforms other baselines in average latency, average bitrate, and Mean Opinion Score (MOS) on Chrome DevTools and Clumsy. Meanwhile, the experimental results also reveal that the average number of identified mental health problems of the proposed mechanism has always been the best with the bandwidth from 10 Mbit/s to 500 Mbit/s.

1. Introduction

The current challenges surrounding the COVID-19 pandemic have caused a huge amount of fear, anxiety, and uncertainty for all of us. While social distancing guidelines help us prevent the spread of the virus, it has also led to determination of social gatherings. The COVID-19 pandemic has hit the entire world hard, but it has especially impacted the mental and physical well-being of many members of the population. Studies show that COVID-19 results in symptoms of anxiety, stress, and depression and may be associated with disturbed sleep [1,2]. We are sure that many of us are already aware of the physical symptoms of COVID-19, but it is equally important to discuss its less known effects which are its adverse mental health consequences [3]. Decreased perceived physical activity due to the pandemic was linked to a higher stress and anxiety response in individuals more or less [4]. Therefore, it is necessary to educate the public about the physical

symptoms of COVID-19, discussing the mental health impacts on members of the population and preventive measures the general population can take to protect their mental and physical well-being from COVID-19 [5, 6]. The target audience for this paper is the students who are more likely to suffer from the mental health problems.

In early 2020, schools and colleges across China started online learning affected by COVID-19. Efforts have been made to find offline resources and carry out online mental health education under the pandemic [7]. Although mental health education emphasizes the subjectivity of students and the activity of the course, in addition to the usual new media approach, the playing of mental health clips and microlessons can arouse students' hearts more than single teaching mode in teaching practice [8, 9]. Online mental health education advocates to touch students' hearts through the microlessons and the introduction of psychological technology, fully respect the subjectivity of mental health

education, and pay attention to the knowledge of psychological discipline and the diversity of mental health education approaches. The quality evaluation of online mental health education is involved with whether students can avoid the influence of the pandemic and stay in a good state of mind during the pandemic.

The mental health-related microlessons should ensure that all students have excellent experience of video playing. Adaptive bitrate (ABR) algorithm allows us to have multiple bitrates to play mental-health-related videos [10]. An encoder sends a single stream into a transcoder, but a transcoder then takes that first single stream and then breaks it up into multiple streams at different quality levels, different resolutions, and different bitrates. Even as the bandwidth may get a little worse or suddenly the Wi-Fi drops off, the video would keep playing and just degrade a little in quality with a seamless way. In order to transmit high-quality videos under different network conditions, most video transmission will use the ABR algorithm. There are two steps from ABR algorithm. (1) Divide the video into blocks, and each block is encoded at a certain bitrate (or quality). (2) Select which bitrate level of video block to obtain according to the number of videos buffered by the client and the throughput recently realized by the client [11–13].

The introduction of ABR algorithm into reinforcement learning can better improve the video quality of microlessons. Reinforcement learning is considered at the sweet spot between control theory and machine learning [14]. Reinforcement learning is essentially a branch of machine learning that deals with how to learn control strategies to interact with a complex environment. It starts with an agent interacting with an environment. The agent is trying to achieve a multistep goal within the environment. For example, a self-driving car might be trying to drive on the roads in the real world [15]. The process of reinforcement learning is shown in Figure 1. In reinforcement learning, the agent selects an action for the environment by observing the state from the environment, and the environment receives the action to update the state and generates a reinforcement signal (reward or punishment) back to the agent. If an agent's action policy leads to a positive reward in the environment, then the agent's tendency to produce that action policy is reinforced. The goal of the agent is to find the optimal policy in each discrete state in order to maximize the desired discount reward and select the next action according to the reinforcement signal and the current state [16, 17]. The selected action not only affects the current reinforcement value, but also affects the state of the environment at the next moment and the final reinforcement value.

Given the above, it is necessary to apply reinforcement learning for quality evaluation of online mental health education. Accordingly, the main contributions of this paper are summarized as follows. (i) The real-time ABR configuration parameters mechanism is proposed to solve the ABR parameter configuration over the network with changing bandwidth. (ii) A learning method based on Q-learning is applied to generate the mapping table, and the network state is mapped to the optimal configuration parameters for the ABR-based mechanism to run in the related network state.

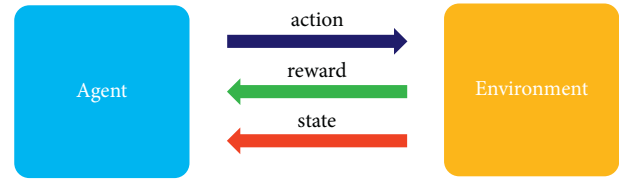


FIGURE 1: Reinforcement learning diagram.

The rest of the paper is organized as follows: Section 2 reviews related work. In Section 3, the real-time ABR configuration parameters mechanism is studied. In Section 4, Q-learning-based offline learning method is proposed. The experimental results are shown in Section 5. Section 6 concludes this paper.

2. Related Work

The ABR algorithm has different bitrate selection strategies, and these decisions affect metrics such as average bitrate or latency, which directly affect the quality of experience (QoE). In recent years, the scholars put more and more emphasis on how to efficiently select the bitrate for users. In [18], based on synchronized video streaming and the state information of next channel, a joint resource allocation and adaptive bitrate optimization algorithm was proposed to maximize the user's quality. Video streaming was particularly important in Internet of Things, in [19], a secure and network state cognitive solution was proposed to solve the wireless channel vulnerable to interference and malicious attacks. In [20], the cooperative and processing framework of ABR at mobile edge computing supported network was proposed to minimize the latency in video searching. In [11], an enhanced mobile-edge-computing-based ABR video transmission scheme was proposed to adjust the bitrate version of video transmission flexibly. In [21], a flexible transcoding strategy was proposed to realize the high-quality ABR video in cellular network. In [22], a secret data embedded video host scheme was proposed to ensure that the value of peak signal-to-noise ratio was high after data embedding. In [12], a joint video transcoding and quality adaptive framework was proposed to maximize the average return. In [23], a rebuffering probability-based optimal algorithm was proposed to avoid download pausing because the buffer of the client was full. In [24], a mobile edge computing-based environment intelligent dynamic adaption algorithm was proposed to help user select the optimal streaming in order to reduce the latency. In [25], an online ABR algorithm was proposed to supply good quality cellular network. In [26], a video quality-aware learning-based ABR approach was presented. In [27], a system that generated ABR for video was presented. In [28], an online control algorithm was designed to minimize rebuffering and maximize video quality.

The reinforcement learning is used extensively in various fields, and scholars have achieved many meaningful results, which are aimed at directly optimizing users' QoE. In [29], an extended QoE-considered fairness rate adaption method was proposed to minimize the number of actions in

balancing the server state. In [30], a reinforcement learning-based DASH technique was constructed to solve the user's QoE problem. In [31], a novel resource allocation model was proposed to allocate resources and reduce latency. In [32], a reinforcement learning-based quality-of-service cognitive adaptive online orchestration method was proposed to adjust the real change of network. In [33], a novel reinforcement learning optimal framework was proposed to ensure good experience of playing. In [34], an enhanced deep Q-learning-based adaption framework was proposed to obtain faster convergence speed and higher average reward.

As far as we know, there are few studies on the combination of Q-learning and ABR for online mental health education, but some strategies have been proposed for online mental health education. Based on artificial intelligence, in [35], a mental health education and consultant expert system was designed to relieve the work pressure and learning pressure. In [36], a mental health problem was studied to evaluate the impact of COVID-19 pandemic on mental health and the effectiveness and attitudes towards online education among Chinese children aged 7–15 years.

3. Real-Time ABR Configuration Parameters Mechanism

The real-time ABR configuration parameters mechanism proposed in this paper applies the segmental stability of network connection to solve the key challenge of sensitivity of ABR parameter configuration to the network environment. The mechanism is mainly divided into two stages. (1) In the offline phase, the system precalculates the optimal configuration selection of each fixed network state. (2) In the online phase, during the mental health-related microlessons video playing, the system detects the changes of network state constantly and selects the optimal precalculated configuration according to the current network state.

3.1. Offline Phase. To map network state to the optimal ABR configuration parameters, this paper uses a learning pipeline consisting of three components: environment parameter estimator, virtual player, and configuration parameters selector. The environmental parameter estimator takes a throughput trace record as input to represent a particular network state and explore different ABR configurations on this throughput record. It is achieved by using a virtual player that simulates the dynamics of a real video player and is able to output the performance of the ABR algorithm when different configurations are applied. The learning process is approximately divided into three steps:

- (i) Analyzing the client throughput sequence set: the mean and mean square deviation of throughput in the phase are calculated by environmental parameter estimator to identify the network state. Unlike traditional algorithms, this representation method fully considers the fluctuation of network state.
- (ii) The virtual player takes the obtained network state as input and explores different configuration

parameters of the ABR mechanism on the network state.

- (iii) The parameter selector compares the performance of different configurations and builds an optimal configuration, which maps a given network state to the optimal configuration.

3.2. Online Phase. The system is mainly composed of client, cloud video server, and so on. Considering that the network state change-point detection algorithm is a computation-based intensive algorithm, it will inevitably bring unnecessary resource consumption when running on the client side. Therefore, the edge computing is introduced to make the system have better performance and lower coupling degree and is conducive to development and maintenance [37]. The detailed process of the online phase can be divided into the following steps:

- (i) During the mental-health-related microlessons video playing, the client player collects playback status information continuously (e.g., buffer length, video playback status, and throughput measurement) and uploads this information to edge computing nodes periodically.
- (ii) The edge node analyzes the network status according to the player status information uploaded by the client and activates the parameter selector when the network status changes.
- (iii) According to the optimal configuration generated by offline learning phase, the parameter selector will select the parameter P that can run best under the current network state and return parameter P to the ABR module of the client player.
- (iv) After receiving the new parameter set, the ABR module replaces the parameters in the model with the new parameters and then selects the appropriate bitrate for the client player parameters from the new model.

4. Q-Learning-Based Offline Learning Method

The ultimate goal of real-time ABR configuration parameters mechanism is to improve users' QoE and achieve higher user participation, while the goal of this paper is to provide a new more flexible model than a fixed model. The real-time ABR configuration parameters mechanism adjusts system parameters in real-time through the perception of network state and then selects the appropriate bitrate by using the mapping relationship between network state and video bitrate, which can be expressed as $V_n = f(\cdot)$. The performance of the selected bitrate V_n can be judged from the following aspects:

- (i) Average video quality, that is, the average quality of each video block, which is defined as follows:

$$\frac{1}{N} \sum_{n=1}^N q(V_n), \quad (1)$$

where n is the number of video blocks and $q(V_n)$ is the quality of each video block.

- (ii) Average quality range, which shows how much video quality changes from one block to another, which is defined as follows:

$$\frac{1}{N-1} \sum_{n=1}^N |q(V_{n+1}) - q(V_n)|. \quad (2)$$

- (iii) Latency: for each video block, the latency happens when the download time $dl_n(V_n)$ is higher than that of the playing time of the buffered video block buf_n , so the total latency is defined as follows:

$$\sum_{n=1}^N dl_n(V_n) - buf_n. \quad (3)$$

Since users may have different preferences for which of the above-mentioned three aspects is more important, the QoE of video clips from 1 to N by weighted sum of the above factors is defined as follows:

$$Q = \sum_{n=1}^N q(V_n) - \rho \sum_{n=1}^N |q(V_{n+1}) - q(V_n)| - \delta \sum_{n=1}^N dl_n(V_n) - buf_n, \quad (4)$$

where ρ and δ are nonnegative weighted parameters corresponding to the video quality change and the latency recovery time respectively. A relatively small ρ indicates that users are not particularly concerned about changes in video quality. On the contrary, a large variable indicates that users care more about this metric than any other parameter.

Therefore, the objective function of real-time ABR configuration parameters problem can be defined as follows:

$$\max \sum_{n=1}^N Q(V_n). \quad (5)$$

Furthermore, the real-time ABR configuration parameters mechanism needs to meet the constraint, which is defined as follows:

$$V_n = f(N, p), \quad \forall n = 1, 2, \dots, N. p \in P. \quad (6)$$

Equation (6) indicates that the relation of real-time ABR configuration parameters mechanism maps the network state N to the bitrate, where p ($p \in P$) is the configuration parameter in real-time ABR configuration parameters mechanism. We take the given network state N as the input and the configuration parameter p that can maximize the video $Q(\cdot)$ as the output to optimize the model, so as to improve the performance of real-time ABR configuration parameters mechanism in a given network state.

In order to achieve the above goal of real-time ABR-based algorithm, it is necessary to calculate in advance which parameters of ABR-based algorithm have the best effect under different network states and then form the mapping

table of optimal configuration parameters of network state and ABR-based algorithm. In this paper, a learning method based on Q-learning is used to generate the mapping table, and the network state is mapped to the optimal configuration parameters for the ABR-based algorithm to run in the related network state.

4.1. Environment Modeling. The system state S_n is mainly determined by two factors, which are the client network state N_n and the client video buffer filling state buf_n . Therefore, the system state is defined as follows:

$$S_n = (buf_n, N_n). \quad (7)$$

This paper uses two-tuple μ_n, φ_n to define the network state N_n [38], where μ_n is the mean value of user-perceived throughput, and φ_n is the standard deviation of user-perceived throughput in this period. This method not only considers the mean value of network bandwidth, but also fully considers the details of network throughput changes, such as the network fluctuation. The throughput recorded in the statistical period is $\{x_1, x_2, \dots, x_n\}$; then μ_n and φ_n can be calculated as follows, respectively:

$$\mu_n = \sum_{n=1}^N \frac{x_n}{n}, \quad (8)$$

$$\varphi_n = \sqrt{\frac{1}{N} \sum_{n=1}^N x_n - \mu_n}.$$

Since the mean value and standard deviation value range of throughput perceived by users are continuous intervals, which are discretized to model state variables in order to facilitate problem definition and reduce the dimension of system state, the value range is shown in Table 1. Here, μ_{\max} represents the maximum of user-perceived throughput and buf_{\max} represents the maximum buffer size. buf_n is defined as follows:

$$buf_n = buf_{n-1} + t_s - dl_n * t_s, \quad (9)$$

where t_s is the starvation time, representing latency has happened. dl_n represents the number of video clips extracted from the client buffer at state n , and since only a complete video clip is allowed to be deleted from the client buffer, dl_n is an integer. The video clips extracted from the client buffer are decoded to produce the original video clips, which are placed in another buffer called the presentation buffer. When the buffer is empty, a new video clip is added from the client buffer. Therefore, the calculation of dl_n should consider the following two aspects: the download time of the last request segment Δt_n and the current display buffer occupancy $Cbuf_{n-1}$, which represents the video clip that is not displayed in the previous stage. The dl_n can be defined as follows:

$$dl_n = \lceil \frac{\Delta t_n - Cbuf_{n-1}}{t_s} \rceil. \quad (10)$$

TABLE 1: The relationship between video buffer filling and network bandwidth discretization.

| The element of network state | Range |
|------------------------------|-------------------------------------|
| μ_n | $[0, \mu_{\max}] \text{sec}$ |
| buf_n | $[0, \text{buf}_{\max}] \text{sec}$ |

The client buffer state buf_n can be summarized as follows. When buf_n is equal to 0, the client buffer becomes empty and the downloaded video clip is directly added to the display buffer, so the video latency has already happened during the downloading of the previous video clip. On the contrary, when buf_n is equal to the maximum buffer size buf_{\max} , the client needs to wait for the display buffer to be empty before re-requesting a new video clip.

4.2. Construction of Reward Function. The reward function rew reflects the quality of a decision. In this paper, there are three main factors that affect users' QoE when video clips are played: the quality level of video clips, the quality variation range of continuous video clips, and the latency during video playing. There is a quasilinear relationship between the Peak Signal-to-Noise Ratio (PSNR) reflecting objective video quality and the Mean Opinion Score (MOS) reflecting subjective video quality [39]. Therefore, a simple linear function is used to model the reward function rew , which is defined as follows:

$$\text{rew} = \text{rew}_c - \gamma \text{rew}_{cc} - \delta \text{rew}_l, \quad (11)$$

where rew_c is the reward function of the video clip quality level, rew_{cc} is the reward function of the continuous video clip quality change frequency, and rew_l is the reward function of video play latency.

Supposing that there are N complete segments in a video sequence, and each segment sustains t seconds, the client player requests a video block with an appropriate quality level according to the available bandwidth. The requested video quality level is $\{q_1, q_2, \dots, q_n\}$, and this set represents the requested video quality sequence. Therefore, the video clip quality reward rew_c can be expressed as the average value of all requested video quality levels when using the current parameters, which is defined as follows:

$$\text{rew}_c = \frac{1}{N} \sum_{n=1}^N q(\text{rew}_n), \quad (12)$$

where rew_{cc} is the reward function of the continuous video clip quality change frequency, which can be evaluated as the average number of quality changes between adjacent sections and the magnitude of the changes, which is defined as follows:

$$\text{rew}_{cc} = \frac{1}{N} \sum_{n=1}^N |q(\text{rew}_{n+1}) - q(\text{rew}_n)|, \quad (13)$$

where rew_{cc} aims to keep the variation range of video clip quality as small as possible. When the quality of two

consecutive video clips suddenly changes greatly, a large penalty will be obtained.

The reward function of video playing latency rew_l can be measured by the ratio of cache starvation event to the total play time. The ratio of total starvation time t_s to total display time t_d is θ , which is defined as follows:

$$\theta = \frac{t_s}{t_d}, \quad (14)$$

$$t_d = N * t + t_s.$$

A starvation event happens when the buffer becomes empty, which is also called latency event. Let r_n , b_n , w_n , and L_{n-1} denote the requested n th video block, bitrate, network bandwidth during the downloading, and the length of buffer reserved before downloading. So the starvation time is defined as follows:

$$t_n^s = \max\left(\frac{b_n \times t}{w_n} - L_{n-1}, 0\right). \quad (15)$$

Therefore, the starvation will not happen if there is sufficient buffer before downloading a new video block; that is, $L_{n-1} \geq b_n * t / w_n$. Otherwise, the starvation time is the difference between the downloading time and the length of buffer reserved. Finally, the total starvation time can be calculated as follows:

$$t_s = \sum_{n=1}^N t_n^s. \quad (16)$$

4.3. Exploration Policy. When agent gets the reward function value and updates the Q-table, there are two main policies for selecting an action in a certain state: one is the exploration, which randomly selects an action. This will have a better effect on exploring unknown actions, and is more conducive to updating the Q-value, which has an important impact on the performance and convergence speed of the algorithm [40]. Another is exploitation, which uses the strategy of past experience to select the action with the highest Q value in the current state in order to get the most expected reward. Epsilon-Greedy (ϵ -greedy) algorithm is explored by probability ϵ ($0 \leq \epsilon \leq 1$) to randomly select the next action. The selection policy for ϵ -greedy algorithm $\pi(a|s)$ is defined as follows:

$$\pi(a|s) \leftarrow \begin{cases} A(s), & \text{if } \lambda < \epsilon, \\ \operatorname{argmax}_{a \in A(s)} Q(s, a), & \text{if } \lambda \geq \epsilon, \end{cases} \quad (17)$$

where λ ($0 \leq \lambda \leq 1$) is a uniformly distributed random number, and values are randomly assigned to λ before each decision. If the $\lambda < \epsilon$, the agent randomly selects an action from action set A to execute. If $\lambda \geq \epsilon$, the agent iterates through all the actions in the action set A and finally performs the specific action that can obtain the maximum value of Q .

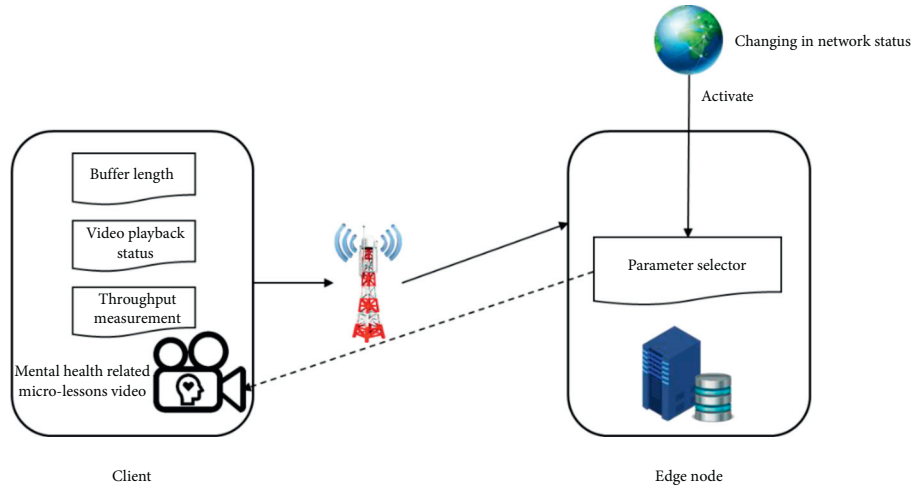


FIGURE 2: The diagram of simulation scenario.

5. Experiment and Results Analysis

5.1. Setup

5.1.1. Environment Construction. The client application is a player based on Dash.js. We have modified Dash.js to a certain extent, and the Dash player will periodically report the throughput changing information, buffer length, video playback state, and other playback information [41]. We deploy the real-time ABR configuration parameters mechanism based on Q-learning algorithm proposed in this paper on the edge node, which is responsible for returning the corresponding ABR algorithm parameters according to the player data reported by the client.

The terminal used in experiment is Qualcomm Snapdragon 888 CPU, 2.84 GHz and 12 GB RAM, and the OS of the terminal is Android 11. The configuration of video server is Intel Xeon processor E5-2697 v4 (45M Cache, 2.30 GHz) and 128 GB RAM, and the OS of the video server is Ubuntu 20.10. The configuration of edge computing node is Intel i9-12900 KF, 3.20 GHz and 32 GB RAM, as shown in Figure 2.

During the experiment, we use Chrome DevTools and Clumsy to simulate different network conditions. In this way, we can use Chrome remote interface based on throughput tracking to control upload and download throughput and latency, which is convenient for us to simulate real application scenarios. All experimental results are reported via the average value form with 10000 times, and the selecting data are randomly spaced 100 times.

5.1.2. Stimulation Dataset. The simulation dataset is all from the throughput tracking data of real students playing mental-health-related microlessons video collected in one month. Each datum contains the video block size and the download time over a period of time. We obtain the throughput by dividing the block size by the download time. The dataset includes both the throughput data for video playback on PC with wired connections and the throughput data on mobile devices with Wi-Fi or cellular connections.

5.1.3. Video Quality Evaluation Parameters. We use the above-mentioned reward function as a way to measure the quality of decisions for individual video clips. To evaluate the effectiveness of different methods, we also need to use a parameter that measures the quality of the video throughout the playing. Generally, QoE can only be measured by subjective evaluation tests or by using objective user perception models. The latency of video depends on the number of latency times and the average length of latency.

5.1.4. Parameters Setting. In real-time ABR configuration parameters mechanism based on Q-learning algorithm, one parameter that needs to be noted is the client reporting period D , which represents the interval between the player reporting information. Generally, it is equal to the number of video blocks downloaded in a period of time. When the decision period is too short, the decision result will be affected by the instantaneous state to a large extent and cannot reflect the change of the state truly. When the decision period is too long, the algorithm cannot make a positive response to the change of the environment state, resulting in the decline of the overall performance. Therefore, the parameter that can obtain the best result is chosen as $D = 15$. In this way, the reward function only considers important measurements to reduce the influence of instantaneous state and suboptimal decision transfer on the result. The parameters and the estimation are shown in Table 2.

We conduct experiments on different parameter combinations, and the statistical results are shown in Table 3. It can be seen from Table 3 that most of the average video quality estimations are close, which indicates that the real-time ABR configuration parameters mechanism based on Q-learning algorithm proposed in this paper has good robustness. When the learning rate is 0.1 and the discount rate is 0.1, higher average video quality evaluation value can be obtained. So, in the following experiment, α and γ are set as 0.1.

TABLE 2: Parameters and the estimation.

| Parameter | Estimation |
|-----------|---------------|
| D | 5, 10, 15, 20 |
| α | 0.1, 0.5, 0.9 |
| γ | 0.1, 0.5, 0.9 |

TABLE 3: Average video quality estimation under different α and γ .

| α | γ | MOS |
|----------|----------|-------|
| 0.1 | 0.1 | 1.331 |
| 0.1 | 0.5 | 1.294 |
| 0.1 | 0.9 | 1.228 |
| 0.5 | 0.1 | 1.217 |
| 0.5 | 0.5 | 1.209 |
| 0.5 | 0.9 | 1.113 |
| 0.9 | 0.1 | 1.095 |
| 0.9 | 0.5 | 0.998 |
| 0.9 | 0.9 | 0.896 |

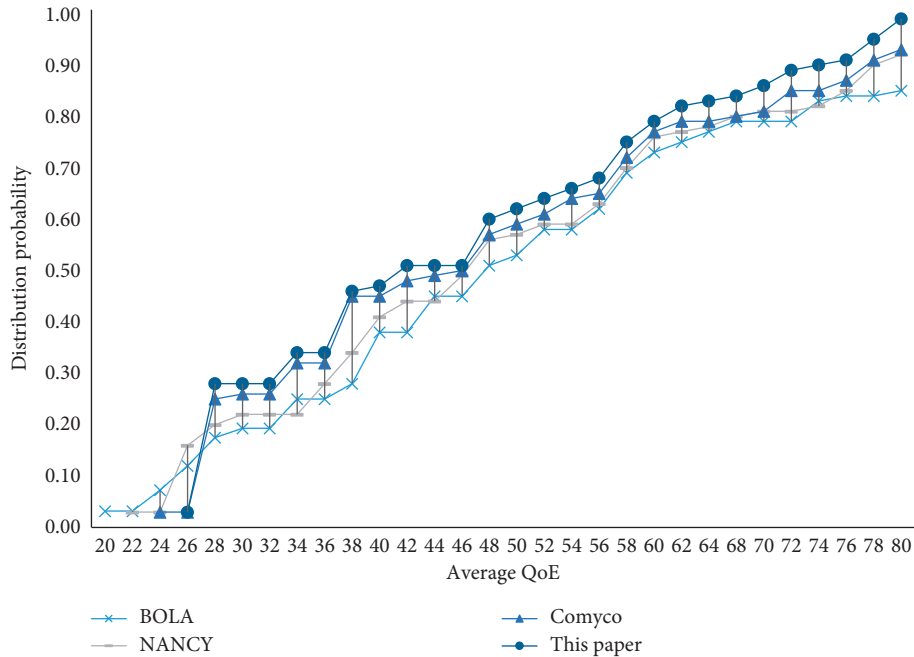


FIGURE 3: The comparison of average QoE.

5.2. *Comparison Analysis.* In this section, the proposed real-time ABR configuration parameters mechanism based on Q-learning algorithm will be verified by comparing with three benchmarks, which are Comyco [26], NANCY [27], and BOLA [28]. Comyco is a video quality cognitive learning-based ABR method, NANCY is a system that generates ABR for video using reinforcement learning, and BOLA is an online control algorithm. This paper uses average latency, average bitrate, and average MOS to verify the effectiveness of the proposed algorithm.

Figure 3 shows the Cumulative Distribution Function (CDF) of the average QoE of the proposed algorithm, Comyco, NANCY, and BOLA, where the abscissa is the average QoE

and the ordinate is CDF. Figure 3 perfectly reflects the distribution probability of average QoE in different intervals. It can be seen from Figure 3 that when the average QoE is large, the probability distribution of the algorithm proposed in this paper remains at a high level and nearly closes to 1.0, which indicates that the real-time ABR configuration parameters mechanism based on Q-learning algorithm proposed in this paper adopts the QoE model as the reward function. The accuracy of the QoE model will directly affect the performance of the algorithm and then improve QoE.

Figure 4 further shows the comparative experimental results of average QoE, average latency, and average bitrate on Clumsy. In order to facilitate comparison, each metric is

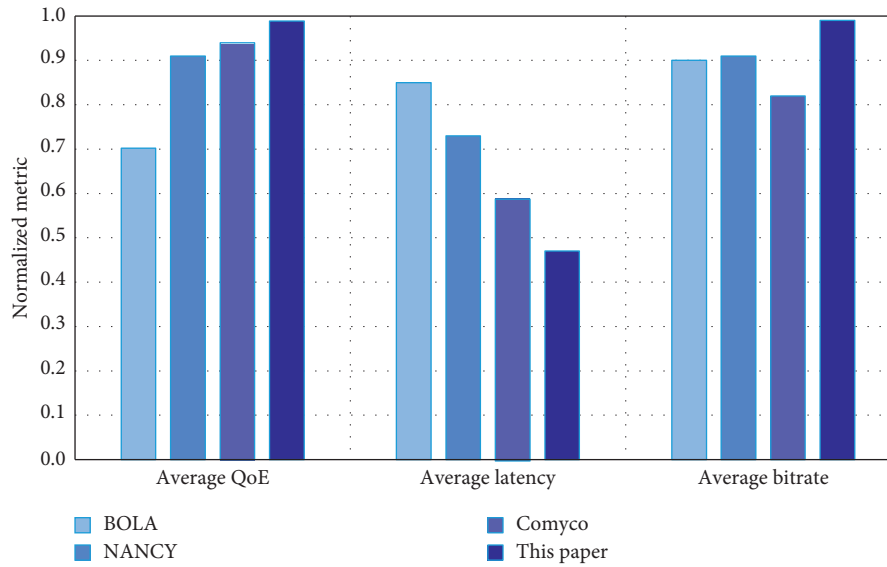


FIGURE 4: The comparison of performance on clumsy.

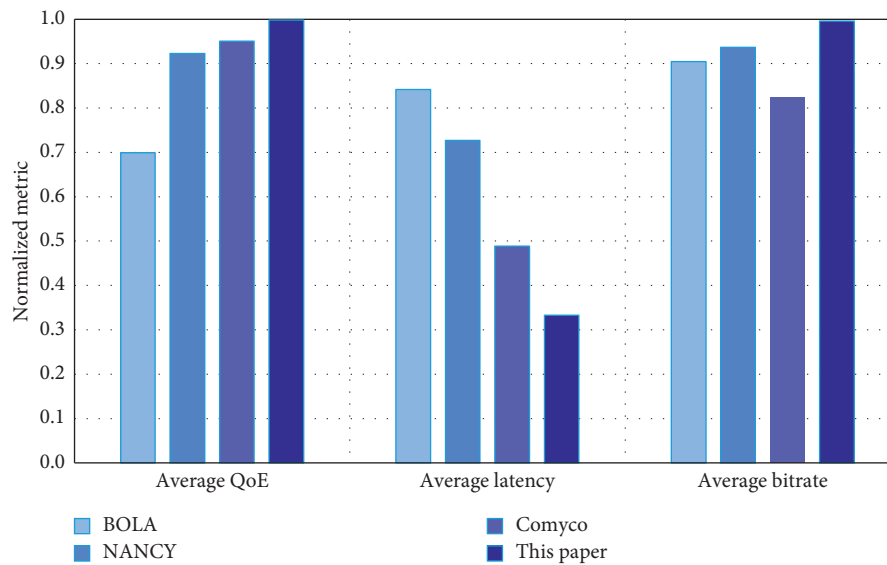


FIGURE 5: The comparison of performance on Chrome DevTools.

normalized, and the calculation method is the ratio of the actual value to the maximum value, and the ordinate represents its normalized value. As can be seen from Figure 4, the performance of real-time ABR configuration parameters mechanism based on Q-learning algorithm is much higher than that of other heuristic algorithms.

Figure 5 shows the algorithm and three benchmarks tested on Chrome DevTools. As can be seen from Figure 5, the performance of the proposed algorithm is significantly better than that of the other three heuristic algorithms. The average QoE, average latency, and average bitrate of four algorithms on Chrome DevTools are generally higher than those on Clumsy.

The calculation method of MOS to measure the QoE is very effective in the scene without latency. As can be seen from Figure 6, the real-time ABR configuration parameters mechanism based on Q-learning algorithm can improve the average video quality estimation compared with the three benchmarks. Meanwhile, it is not hard to find from Figure 6 that with the increasing of the maximum buffer space, the average video quality estimation will also increase. With the increase of buffer space, in poor network environment, more video data will be buffered at the initial stage of buffering, which reduces the probability of rebuffering during video playing, and thus reduces the duration and frequency of latency.

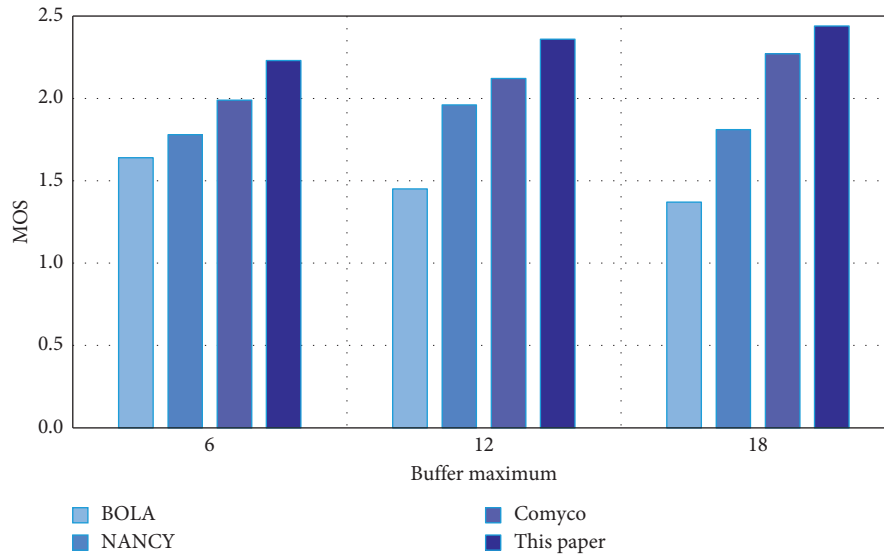


FIGURE 6: The comparison of average video quality estimation with different buffer maximums.

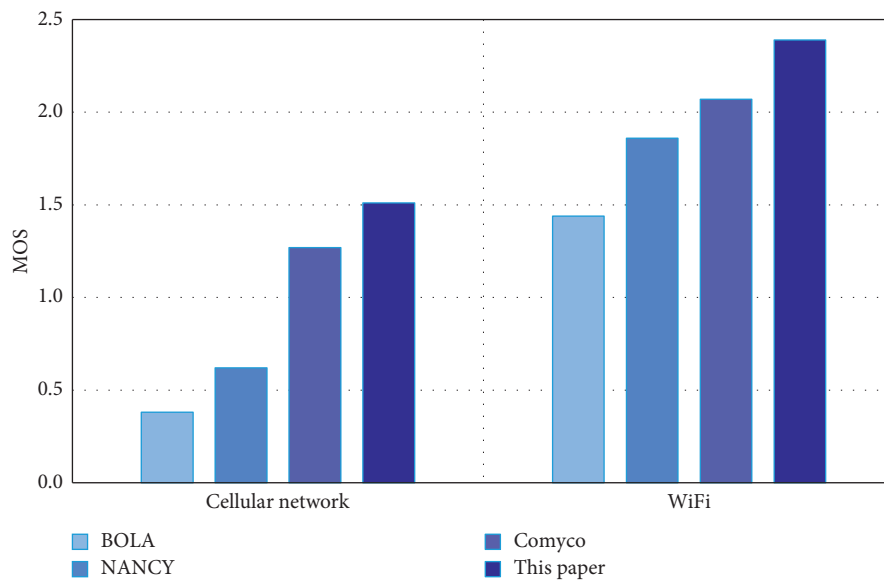


FIGURE 7: The comparison of average video quality estimation with different connecting methods of the network.

As shown in Figure 7, the real-time ABR configuration parameters based on Q-learning algorithm can obtain better video service quality under different network environments compared with three benchmarks. This further shows that the video bitrate adjustment mechanism proposed in this paper has a good adaptability in actual network environment. At the same time, the comparative analysis will find that Wi-Fi connecting method can obtain better video service quality since cellular network is complex and changeable. That is prone to network mutation, making the prediction of network bandwidth not accurate.

High-quality mental-health-related microlessons videos can improve and evaluate the quality of online mental health education more effectively, so as to identify mental health problems more effectively. The proposed mechanism plays well in average latency, average bitrate, and average MOS, and it also has a good performance in identifying the mental health problem. As can be seen from Figure 8, the average number of mental health problems identified by the proposed mechanism is consistently the highest in the bandwidth range of 10 Mbit/s to 500 Mbit/s.

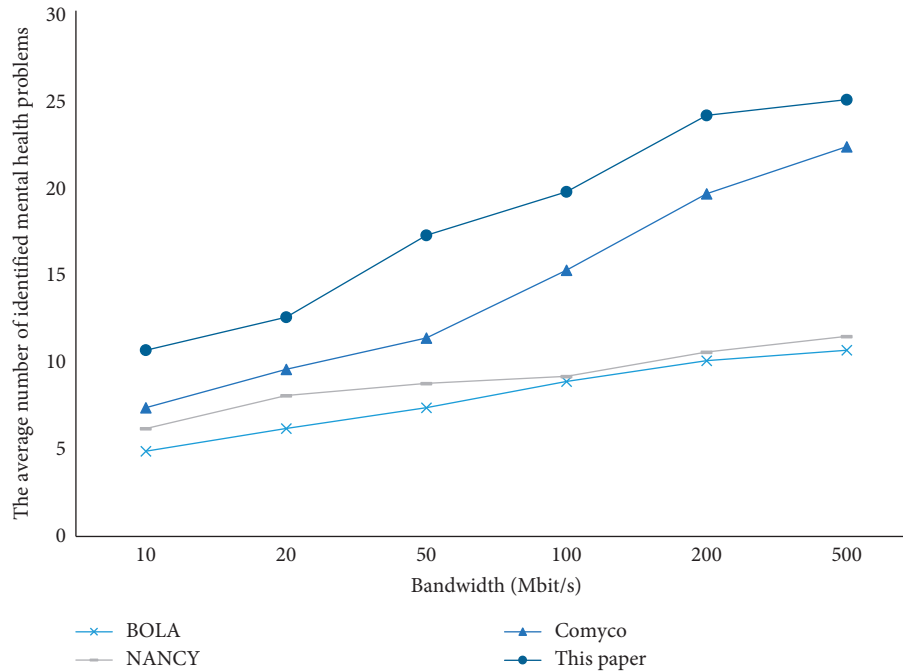


FIGURE 8: The comparison of identifying the mental health problems.

6. Conclusions

The pandemic has affected our lives in numerous ways, but it is imperative that we focus on our mental health, especially for the students who are more susceptible to the surroundings. While countries around the world continue to mobilize to contain the spread of COVID-19. A large body of scientific studies show that there is a close relationship between indicators such as unemployment, mental health, and suicide. So, during the pandemic, the online mental health education is particularly important for impressionable students. In this study, at first, the real-time ABR configuration parameters mechanism is proposed to solve the configuration over the network with changing bandwidth. Then, a learning method based on Q-learning is applied to generate the mapping table, and the network status is mapped to the optimal configuration parameters for the ABR-based mechanism to run in the related network status. Finally, the simulation results demonstrate that the proposed algorithm in this paper improves upon average latency, average bitrate, and MOS on Chrome DevTools and Clumsy. Besides, the proposed mechanism also plays well in identifying the mental health problems.

For the moment, the pandemic will continue for a while. In addition to video microclasses, interactive live video streaming is also a good choice for students' mental health education. In future, we will study the codec technique and transmission protocol in live. In addition, the introduction of imitation learning, super-resolution reconstruction, scalable coding, and other methods into ABR algorithm can effectively solve the problems of low sample efficiency and slow convergence of ABR algorithm based on deep reinforcement learning, which is also a relatively meaningful research direction in the future.

Data Availability

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

Conflicts of Interest

The author declares no conflicts of interest.

References

- [1] S. Gas, H. E. Ozsoy, and K. C. Aydin, "The association between sleep quality, depression, anxiety and stress levels, and temporomandibular joint disorders among Turkish dental students during the COVID-19 pandemic," *Cranio-the Journal of Craniomandibular & Sleep Practice*, 2021.
- [2] G. Tasdemir Yigitoglu, A. Yilmaz, and H. Yilmaz, "The effect of Covid-19 on sleep quality, anxiety and depression on healthcare staff at a tertiary hospital in Turkey," *Archives of Psychiatric Nursing*, vol. 35, no. 5, pp. 504–510, 2021.
- [3] S. Sockalingam, C. Clarkin, E. Serhal, C. Pereira, and A. Crawford, "Responding to health care professionals' mental health needs during COVID-19 through the rapid implementation of project ECHO," *Journal of Continuing Education in the Health Professions*, vol. 40, no. 3, pp. 211–214, 2020.
- [4] I. P. Albery, M. M. Spada, and A. V. Nikcevic, "The COVID-19 anxiety syndrome and selective attentional bias towards COVID-19-related stimuli in UK residents during the 2020-2021 pandemic," *Clinical Psychology & Psychotherapy*, 2021.
- [5] X. Li, L. M. Vanderloo, J. L. Maguire et al., "Public health preventive measures and child health behaviours during COVID-19: a cohort study," *Canadian Journal of Public Health = Revue Canadienne de Sante Publique*, vol. 112, no. 5, pp. 831–842, 2021.

- [6] P. A. Apanga and M. T. Kumbeni, "Adherence to COVID-19 preventive measures and associated factors among pregnant women in Ghana," *Tropical Medicine & International Health*, vol. 26, no. 6, pp. 656–663, 2021.
- [7] M. Maqableh and M. Alia, "Evaluation online learning of undergraduate students under lockdown amidst COVID-19 pandemic: the online learning experience and students' satisfaction," *Children and Youth Services Review*, vol. 128, 2021.
- [8] C. P. van der Westhuizen and A. Golightly, "Video annotation software application for thorough collaborative assessment of and feedback on microteaching lessons in geography education," *Journal of Geography in Higher Education*, vol. 39, no. 3, pp. 420–436, 2015.
- [9] H. S. Tuluze and S. Cecen, "Scrutinizing practicum for a more powerful teacher education: a longitudinal study with pre-service teachers," *Educational Sciences-Theory & Practice*, vol. 16, no. 1, pp. 127–151, 2016.
- [10] Z. Meng, Y. Guo, Y. Shen et al., "Practically deploying heavyweight adaptive bitrate algorithms with teacher-student learning," *IEEE/ACM Transactions on Networking*, vol. 29, no. 2, pp. 723–736, 2021.
- [11] X. Xu, J. Liu, and X. Tao, "Mobile edge computing enhanced adaptive bitrate video delivery with joint cache and radio resource allocation," *IEEE Access*, vol. 5, pp. 16406–16415, 2017.
- [12] Y. Guo, F. R. Yu, J. An, K. Yang, C. 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. 69, no. 4, pp. 3879–3892, 2020.
- [13] L. Gao, M. Tang, H. Pang, J. Huang, and L. Sun, "Multi-user cooperative mobile video streaming: performance analysis and online mechanism design," *IEEE Transactions on Mobile Computing*, vol. 18, no. 2, pp. 376–389, 2019.
- [14] C. You, J. Lu, D. Filev, and P. Tsiotras, "Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning," *Robotics and Autonomous Systems*, vol. 114, pp. 1–18, 2019.
- [15] J. Yang, S. Harish, C. Li, H. Zhao, B. Antous, and P. Acar, "Deep reinforcement learning for multi-phase microstructure design," *Computers, Materials & Continua*, vol. 68, no. 1, pp. 1285–1302, 2021.
- [16] N. Bougie and R. Ichise, "Skill-based curiosity for intrinsically motivated reinforcement learning," *Machine Learning*, vol. 109, no. 3, pp. 493–512, 2020.
- [17] S. Lü, S. Han, W. Zhou, and J. Zhang, "Recruitment-imitation mechanism for evolutionary reinforcement learning," *Information Sciences*, vol. 553, pp. 172–188, 2021.
- [18] J. Zhang, H. Wu, X. Tao, and X. Zhang, "Adaptive bitrate video streaming in non-orthogonal multiple access networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 3980–3993, 2020.
- [19] Z. Zeng, H. Che, W. Miao, J. Huang, and S. Zhang, "Towards secure and network state aware bitrate adaptation at IoT edge," *Journal of Cloud Computing-Advances Systems and Applications*, vol. 9, no. 1, 2020.
- [20] T. X. Tran and D. Pompili, "Adaptive bitrate video caching and processing in mobile-edge computing networks," *IEEE Transactions on Mobile Computing*, vol. 18, no. 9, pp. 1965–1978, 2019.
- [21] C. Liu, H. Zhang, H. Ji, and X. Li, "MEC-assisted flexible transcoding strategy for adaptive bitrate video streaming in small cell networks," *China Communications*, vol. 18, no. 2, pp. 200–214, 2021.
- [22] Y. Chen, H. Wang, H. Wu, and Y. Liu, "An adaptive data hiding algorithm with low bitrate growth for H.264/AVC video stream," *Multimedia Tools and Applications*, vol. 77, no. 15, pp. 20157–20175, 2018.
- [23] P. Wisniewski, J. M. Batalla, A. Beben, P. Krawiec, and A. Chydzinski, "On optimizing adaptive algorithms based on rebuffering probability," *ACM Transactions on Multimedia Computing Communications and Applications*, vol. 13, no. 3, 2017.
- [24] J. Kim, Y. Won, C. Yoon et al., "A context-aware adaptive algorithm for ambient intelligence DASH at mobile edge computing," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 4, pp. 1377–1385, 2020.
- [25] B. Wang, F. Ren, J. Yang, and C. Zhou, "Improving the performance of online bitrate adaptation with multi-step prediction over cellular networks," *IEEE Transactions on Mobile Computing*, vol. 20, no. 1, pp. 174–187, 2021.
- [26] T. Huang, C. Zhou, X. Yao et al., "Quality-aware neural adaptive video streaming with lifelong imitation learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2324–2342, 2020.
- [27] P. Saxena, M. Naresh, and M. Gupta, "NANCY: neural adaptive network coding methodology for video distribution over wireless networks," in *Proceedings of the 2020 IEEE Global Communications Conference (GLOBECOM)*, Taipei, Taiwan, December 2020.
- [28] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: near-optimal bitrate adaptation for online videos," *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1698–1711, 2020.
- [29] S. Altamimi and S. Shirmohammadi, "QoE-fair dash video streaming using server-side reinforcement learning," *ACM Transactions on Multimedia Computing Communications and Applications*, vol. 16, no. 2, 2020.
- [30] X. Wei, M. Zhou, S. Kwong et al., "Reinforcement learning-based QoE-oriented dynamic adaptive streaming framework," *Information Sciences*, vol. 569, pp. 786–803, 2021.
- [31] P. Zhou, Y. Xie, B. Niu et al., "QoE-aware 3D video streaming via deep reinforcement learning in software defined networking enabled mobile edge computing," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 1, pp. 419–433, 2021.
- [32] J. Chen, J. Chen, and H. Zhang, "DRL-QOR: deep reinforcement learning-based QoS/QoE-aware adaptive online orchestration in NFV-enabled networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1758–1774, 2021.
- [33] A. Bentaleb, A. C. Begen, and R. Zimmermann, "ORL-SDN: online reinforcement learning for SDN-enabled HTTP adaptive streaming," *ACM Transactions on Multimedia Computing Communications and Applications*, vol. 14, no. 3, 2018.
- [34] J. Liu, X. Tao, and J. Lu, "QoE-oriented rate adaptation for DASH with enhanced deep Q-learning," *IEEE Access*, vol. 7, pp. 8454–8469, 2019.
- [35] L. Lilingling, "Online mental health education teaching mode and empirical research based on artificial intelligence," *Journal of Intelligent & Fuzzy Systems*, vol. 40, no. 2, pp. 3467–3476, 2021.
- [36] Z. Ma, S. Idris, and Y. Zhang, "The impact of COVID-19 pandemic outbreak on education and mental health of Chinese children aged 7–15 years: an online survey," *BMC Pediatrics*, vol. 21, no. 1, 2021.

- [37] D. Wang, Y. Peng, X. Ma et al., “Adaptive wireless video streaming based on edge computing: opportunities and approaches,” *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 685–697, 2019.
- [38] J. Liu, Y. Xu, and Z. Li, “Resource allocation for performance enhancement in mobile ad hoc networks,” *IEEE Access*, vol. 7, pp. 73790–73803, 2019.
- [39] M. Barazzetta, M. Colombo, L. Bastianelli et al., “Testing of VoLTE mean opinion score in reverberation chambers,” *IET Science, Measurement & Technology*, vol. 14, no. 8, pp. 949–954, 2020.
- [40] L. Ma, S. Cheng, and Y. Shi, “Enhancing learning efficiency of brain storm optimization via orthogonal learning design,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 11, pp. 6723–6742, 2021.
- [41] K. Spiteri, R. Sitaraman, and D. Sparacio, “From theory to practice: improving bitrate adaptation in the DASH reference player,” *ACM Transactions on Multimedia Computing Communications and Applications*, vol. 15, no. 2, 2019.