

Deep reinforcement learning based quality of experience aware for multimedia video streaming

Manjunatha Peddareddygari Bayya Reddy, Sheshappa Shagathur Narayanappa

Department of Information Science and Engineering, Sir M Visvesvaraya Institute of Technology, Bengaluru, India

Article Info

Article history:

Received Dec 5, 2023

Revised Jul 4, 2024

Accepted Jul 9, 2024

Keywords:

Deep reinforcement learning

Mean opinion score

Quality of experience

Rebuffering

Video streaming

ABSTRACT

Video streaming involves the continuous delivery of video files from a server to a client, where multimedia streaming is employed for playback through an online or offline media player. Video streaming uses live broadcasts to enhance direct communication with community partners and customers. The existing methods have less video streaming quality and are unable to efficiently adapt to the dynamic conditions of the network. In this research, an adaptive bit rate (ABR) method depending on dynamic adaptive video streaming over hypertext transfer protocol or HTTP (DASH) based deep reinforcement learning (DRL) named DASH-based DRL is proposed to determine the following segment's quality in DASH video streaming with wireless networks. The proposed algorithm significantly improves the quality of experience (QoE) performance by providing a highly stable video quality, reducing the distance factor, and enduring smooth streaming sessions. The performance of the proposed method is analyzed based on performance measures of performance improvement, QoE metrics, mean opinion score, normalized value of QoE, average of normalized value of QoE, switching quality, and rebuffering time. The suggested algorithm obtains a high average normalized QoE of 0.72, average switching quality of 0.15, and an average rebuffering time of 0.16 sec, which is comparatively superior to other algorithms like real-time streaming protocol (RTSP), HTTP live streaming (HLS) and reinforcement learning (RL).

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Manjunatha Peddareddygari Bayya Reddy

Department of Information Science and Engineering, Sir M Visvesvaraya Institute of Technology

Bengaluru, India

Email: manjupb.reddy@gmail.com

1. INTRODUCTION

Multimedia streaming has gained popularity among consumers worldwide [1]. The availability of streaming content has increased along with the number of streaming services such as Amazon, Home Box Office (HBO), Prime, and Netflix [2]. The growing multimedia surrounding including the internet of things (IoT), big data processing, and cloud computing has a huge impact on the lifestyle of internet users [3], [4]. multimedia (M-IoT) is a significant network technology that enables communication and interaction between businesses, people, and things like vehicles, sensors, and cameras [5]. Long term evolution (LTE) provides spatial multiplexing, higher-level modulation, and large bandwidths by utilizing orthogonal frequency division multiple access (OFDMA) downlink transfer technology with various radio resource management (RRM) procedures [6], [7]. High downlink data rates support high data rates, spectral efficiency, and reduced packet delays [8]. Viewport-adaptive streaming is a simple approach for reducing the transmission bandwidth in which the quality of scenes is constantly modified in real time, based on the user's field-of-view (FoV) [9]. Tile-based adaptive streaming is now a commonly adaptive approach among viewport-based solutions for

sending quality-variable tiles without compromising the quality of visuals in response to user interaction [10], [11]. Quality of experience (QoE) is described by the International Telecommunication Union (ITU) as an essential metric in multimedia research. QoE refers to a service or overall acceptability of an application, as evaluated subjectively by the end user [12]. An influencing factor (IF) is any feature of a user, application, context, or system that affects the QoE in its present state. Quality prediction is an essential procedure for every service provider to ensure customer satisfaction [13], [14]. Subjective testing is an essential component of any multimedia services evaluation, it seeks objective approaches for forecasting the video quality because subjective experiments are time-consuming and costly [15]. Signal-based video and image quality measures are categorized into three groups depending on the presence of the reference signal: full reference (FR), no-reference (NR), and reduced-reference (RR) [16]. Since the reference signal is frequently unavailable to network operators or service providers, NR metrics are of great significance to service providers for QoE evaluation of cloud gaming and passive video streaming services [17]–[20]. The existing methods in dynamic adaptive video streaming over hypertext transfer protocol or HTTP (DASH) video streaming on heuristics and machine learning (ML) methods are utilized for making a selection of bitrate decisions. Though, these methods contain less performance and are unable for adapting the dynamic conditions of the network efficiently.

Choi *et al.* [21] implemented an adaptive bit rate (ABR) algorithm named ABRaider to improve QoE in video streaming. In the implemented method, multiple stage reinforcement learning (RL) that has offline and online stages is utilized. In the offline stage, the method combined ABR methods' strengths and created policies matched to different environments. In the online stage, the method concentrated on the particular environment of individual users by leveraging the client's execution power. The implemented method provided high QoE to every user in different environments with multiple strategies, including all the environments of the user. The implemented method enhanced the QoS of video information. However, because of bandwidth constraints, poor quality images were transmitted. Kang and Chung [22] introduced a DASH which was named DASH-based HTTP method to improve QoE in video streaming. The introduced method utilized an online RL for QoE analysis, at the same time, the client updated the ABR method for performing the video streaming to adapt change in-network status of client. By classification of network strategy, network traces were utilized as a dataset, while the present user's network environment was classified by characteristics. The introduced method provided high video quality. However, the method minimized the rate of performance improvement by regulating count of threads utilized for training. Zeng *et al.* [23] suggested field-of-view (FoV) aware–multi agent soft actor critic (FA-MASAC) method to enhance QoE in video streaming. The suggested method utilized multiple agent deep reinforcement learning (MADRL) based on joint edge caching and selection bitrate. The MADRL method was used for multiple category 3600 video streaming to enhance the mean QoE of the user. Then, the FA-MASAC method was implemented to support the agents' optimum edge caching and selection of bitrate decisions in a distributed way, wherein every video category was taken as the agent. The suggested method minimized time and enhanced QoE. Nonetheless, the method had difficulties during training because of insufficient data.

Wang *et al.* [24] presented a learning based online qoe optimization method named multi-agent policy gradient for finite time horizon (MAPG-finite) to enhance QoE in video streaming. The presented method developed a policy gradient algorithm for multiple agent learning issues with nonlinear objectives. The presented method allowed for optimization distributions of bandwidth between multi-agents, increased the QoE and objective of fairness on rewards of video streaming. The presented method reduced rebuffering and improved video quality. Nonetheless, this method covered a small area but faced the problem of redundancy. Li *et al.* [25] developed a prediction optimization transmission (POT) method to enhance QoE in video streaming. The developed method made predictions of network bandwidth and FoV in every little rolling window to minimize the prediction error. The developed method considered the upper bounded QoE contribution of the next point cloud video for improving the model performance. Additionally, the deep reinforce learning-based real-time resolver was developed for making decisions on fixed architecture optimization issues in every roll. This method improved the QoE of streaming video but did not consider the dynamic behavior of wireless networks. Tang *et al.* [26] implemented an attention-based multi-agent reinforcement learning (AMARL) method to enhance QoE in video streaming. The implemented method combined user allocation and bitrate adaptation to a single optimization objective. Then, a multiple-agent reinforcement learning technique integrated with an attention mechanism for solving the issue of multiple edge servers was deployed. The implemented method enhanced the compression rate of video frames. Nonetheless, poor quality images were transmitted because of bandwidth constraints. From the overall analysis, it is drawn that the previous research have limitations of bandwidth constraints, transmission of poor-quality images, minimized rate of performance improvement by regulating number of threads utilized for training, and difficulties in training because of insufficient data. This research suggests an ABR method that depends on deep reinforcement learning (DRL) to determine the following segment's quality in video

streaming of DASH with wireless networks. The DRL algorithm has three major parameters namely, conditions of the network, buffer state, and distance factor among successive segments of a video. The main aim of the proposed algorithm to give a high quality user experience. The major contributions of this research are given below: i) DRL method for DASH video streaming concentrates on managing a distance of quality among consecutive segments, ii) Markov decision process (MDP) is developed in DASH video streaming along a learning method to enable optimal solution determination by RL, and iii) The reward function is developed, taking into consideration the perceived quality change rate, rebuffering and a quality of every video segment.

The remaining section of paper is organized in following format. Section 2 describes the proposed methodology, while detailed explanation of deep reinforcement learning is described in section 3. Section 4 describes results, comparative analysis and discussion of proposed methodology, and the conclusion of this paper is given in section 5.

2. PROPOSED METHOD

In this section, DRL algorithm is proposed for providing a high-quality video experience. The suggested DRL approach focuses on how agents make decisions to optimize a particular objective across multiple steps in intricate environments, aiming to accrue significant collective rewards [27]. The proposed algorithm makes consecutive decisions within distinct time intervals, which makes it significant in solving complicated problems. In RL, there are two fundamental learning methods, MDP and Q-learning. The MDP is utilized to derive solutions of RL in video streaming DASH. The RL algorithm has five components which are agent, environment, state, action and rewards. The agent considers sequential actions in series depending on the present environment state. Afterward, every action and environment is moved to new state, and an agent received a reward in response. A main aim of an agent is to increase an entire collective reward along a particular count of stages. The elements of this model are described below:

- Agent: An agent of RL utilizes the present state (like bandwidth, buffer state, and quality of segment downloaded previously) for choosing the following action to be considered (following segment of a video to be downloaded).
- Environment: In RL, the agent processes included network conditions and video stream.
- Reward: Reward of an RL is where the agents are received for taking a specific action.

The agent utilizes a present state to choose an action that is implemented in an environment, and then the environment returns a reward to agent, after that updates its policy. Figure 1 represents the processing of the proposed DRL.

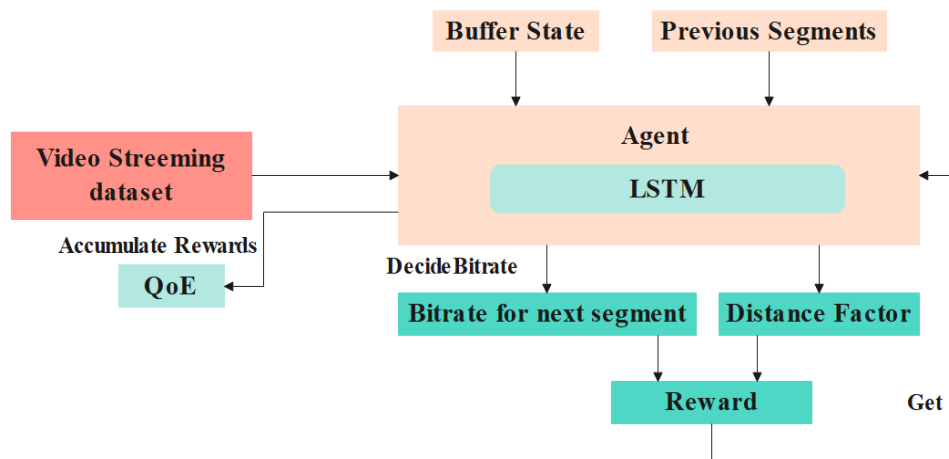


Figure 1. Process of proposed DRL

In DRL algorithm which concentrates on DASH, the DASH server considers gathering of videos as $V\{v_1, v_2, \dots, v_N\}$. Every video in the server is segmented to different bitrates or quality levels, represented as $Q\{q_1, q_2, q_3, \dots, q_M\}$ with every segment having a constant duration [28]. Additionally, every video is attended through media presentation description (MPD) record which has comprehensive data on these segments. At a beginning of video streaming, a DASH client procedure is implemented through requesting

and then identifying MPD record. By this technique, client-side adaptation agent is tasked with defining much matchable bitrate for every segment depending on optimum policy. By downloading a initials segment, the agent monitors the environment and includes complex parameters like measurements of bandwidth, status of buffer, and quality (i.e., bitrate) of the segment that is previously downloaded. Agent influences environmental data to made informed decisions around optimum bitrate for a following segment. The iterative procedure endures as a agent considers actions and receives rewards, while collecting the entire segments has downloaded.

The MDP method formulates the strategy of optimum bitrate adaptation, supporting the factor of distance among two successive segments [29]. The aim is to ensure a huge level satisfaction for users represented as $S = \{s_1, s_2, s_3, \dots\}$ that is described through the elimination of interruptions of playback and the establishment of reliably huge-quality video streaming experience. Here, S represents the function of eliminating interruptions in playback and maintaining stable, high-quality streaming. The mathematical formula is represented as (1),

$$S = f(\text{playback}_{\text{interruption}}, \text{stable_high_quality}) \quad (1)$$

3. DEEP REINFORCEMENT LEARNING

The research endeavors for ensuring consistently huge perceptual quality and increase user satisfaction while removing rebuffering. Here, R is framed as a reward function where, $R = f(r, rb, q)$ which is considered to account a rate of perceived quality change (r), rebuffering occurrences (rb), and quality of every video segment (q). The video sequences are tested on various devices across different conditions of the internet, within the session of the ongoing streaming. Additionally, the bitrates are switched in the format of $\{(144p, 240p), (144p, 360p), (144p, 720p) \dots\}$.

Based on a findings, qualities are classified into 3 distinct categories as Q_{high} , Q_{medium} and Q_{poor} . To describe these categories, the quality sets $Q_{\text{high}} = \{1080p, \dots, 2160p\}$ which represents high quality, $Q_{\text{medium}} = \{360p, \dots, 720p\}$ which represents medium quality and $Q_{\text{poor}} = \{144p, \dots, 240p\}$ which represents poor quality are created. The q_t represents the quality level and s_i represents the segment that is classified to the quality set $Q = \{Q_{\text{high}}, Q_{\text{medium}}, Q_{\text{poor}}\}$, and the bandwidth is classified to the set $Bw = \{Bw_{\text{low}}, Bw_{\text{medium}}, Bw_{\text{high}}\}$.

3.1. Perceived quality change

The perceived quality change is implemented on a rate of quality change. In this research, perceived quality change (PQC) is considered while a user detects the switch, suggesting which the shift in quality along a similar quality set is not taken as PQC. Hence, the identify changes when the quality level q_t changes from poor to high-quality set q_{t+1} . To count these changes, the distance factor represented as $\Delta fact$ that measures the difference between two quality levels is implemented. Table 1 describes the distance factor values among different quality levels. In accordance with the Table 1, $\Delta fact_{q_t}^{q_{t+1}} = |1|$, where q_t represents medium quality and q_{t+1} represents high quality.

High quality	Medium quality	Poor quality
0	-1	-2
1	0	-1
2	1	0

3.2. Rebuffering

The Rb represented as rebuffering which happens subsequently to download every segment is calculated through comparing a playback period of the download segment sp , and download completion time represented as sd . Rb happens when sd exceeds sp and suggests a pause in the video playback. Therefore, Rb is referred to as a collected count of video pauses. The mathematical formula of rebuffering is represented as (2),

$$Rb = \sum_{k=1}^N Rb_k \quad (2)$$

3.3. Segment quality

The segment quality shows a quality state of every downloaded segment and is represented as q_i . The average quality (Avg_q) is calculated by the sum of whole segment quality (q_k) separated by a whole number of segments (N), and its numerical formula is expressed as (3).

$$Avg_q = \frac{\sum_{k=1}^N q_k}{N} \quad (3)$$

where N represents a whole number of segments.

3.4. QoE function

The QoE function depends on 3 major parameters: quality of every segment, perceptual change in quality and rebuffering pauses. Agent receives a reward for every decision caused and its mathematical formula is given in (4). The mathematical formula for calculating QoE is represented as (5). Here, μ represents penalties given to rebuffering and λ represents the quality variations. Then, the QoE_{value} is normalized, and the mathematical formula is given in (6). Where, N represents the total number of video segments.

$$r_t = f(q_t, q_{t-1}) \quad (4)$$

$$QoE_{max} = \sum_{k=1}^k (r_k) - \sum_{k=1}^k |\mu Rb_k| - \sum_{k=1}^{k-1} |\lambda (\Delta f act_k)| \quad (5)$$

$$QoE = \frac{QoE_{max}}{N} * 100 \quad (6)$$

3.5. Markov decision process

The requirements from the client side for making successive decisions over time are represented as t . Markov decision process (MDP) is utilized to validate the process of adaptation. MDP is classified by five major components namely, agent, environment, states, actions and rewards, and it is illustrated in Figure 2.

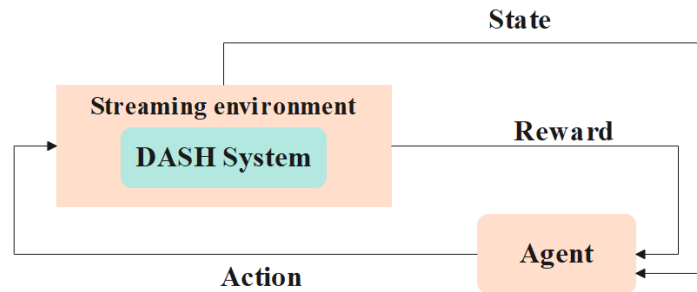


Figure 2. Architecture of RL

Here, action determines the optimum bit rate for the following segment. Agent represents the responsibilities from the client-side, along with their primary task to choose an action represented as at . The action involves the choice of relevant bitrate or quality q_t for the following segment S_i download. The decision of the agent is based on current bandwidth calculations Bw_t , usual buffer state $buff_state_t$ and quality q_{t-1} of preceding segment s_{i-1} . The bitrate selection of agent results in receiving the rewards.

In adaptive video streaming, the segments are determined as fixed set states $S = \{s_0, s_1, \dots, s_N\}$ and the video is separated into N segments. At the same time, a group of actions is created as $\{a_t\}$ with t considering values from $\{0, 1, 2, \dots, N\}$, thereby representing the decision at stage t . The duration of every stage t is resolute through the segment download time. Action set for the particular state is represented as $A(s) = \{A_{s1}, A_{s2}, A_{s3}\}$. Where, A_{s1} represents a scenario while q_t and q_{t-1} exist in a similar quality set, as determined earlier. A_{s2} relates to where q_t and q_{t-1} do not have a similar quality set, but are closest in levels of quality. A_{s3} describes occurrences where q_t falls within the poor quality set. The transitions of the set at every stage are described as $s_t = \{q_{t-1}, q_t, buff_state_t, Bw_t\}$, q_{t-1} represents the previous segment's quality, q_t represents the downloaded segment's quality, Bw_t represents present bandwidth and $buff_state_t$ represents the present buffer state. To determine the following segment's quality, the agent uses policy π as

an estimation approach. In this research, the deep neural network (DNN) is implemented for estimating policy $\pi(a|s; \theta)$ to a map representation in state S to the following state S' . The agent is summarized within DNN that provides policy for decision making when θ represents weights within the DNN. State transition at time t is represented as $S = (Bw, buff_state_t, q_t)$ after introducing $\pi(a|s; \theta)$ policy. The chosen action is $a_t = A_{s1}$ and its numerical formula is represented as (7),

$$S' = (Bw', buff_state'_t, q'_t) \quad (7)$$

where, $q'_t = \Delta q_t$ when, (q_t, q_{t-1}) is in similar quality set, Bw' is equal to ΔBw_t , $buff_state'_t$ is equal to $\Delta buff_state_t$. The Reward function $R_i(S_t)$ is received after downloading every segment S_t . The numerical formula for the reward is represented as (8),

$$r_t = R(S_t = S) = f(q_t, q_{t-1}) \quad (8)$$

3.6. Agent design

The agent is created by utilizing long short-term memory (LSTM) that made decisions depending on a present bandwidth, buffer state and the quality of a previous downloaded segment. LSTM considers input as a state $S_t = (q_t, Bw_t, Buf_t)$ and chooses an action that relates to the state $S_{t+1} = (q_{t+1}, Bw_t, Buf_t)$. Then, agent receives a reward depending on the selected action. Whether an agent received positive reward, it goes to a following state and makes decisions. But whether an agent received negative reward, an alert message is sent to agent, permitting potential corrective metrics. Then, a agent detects environment for determining whether a correction is made without badly affecting a QoE. Figure 3 represents a agent architecture where a result at time stage $t - 1$ supports as input to a following decision at time step t .

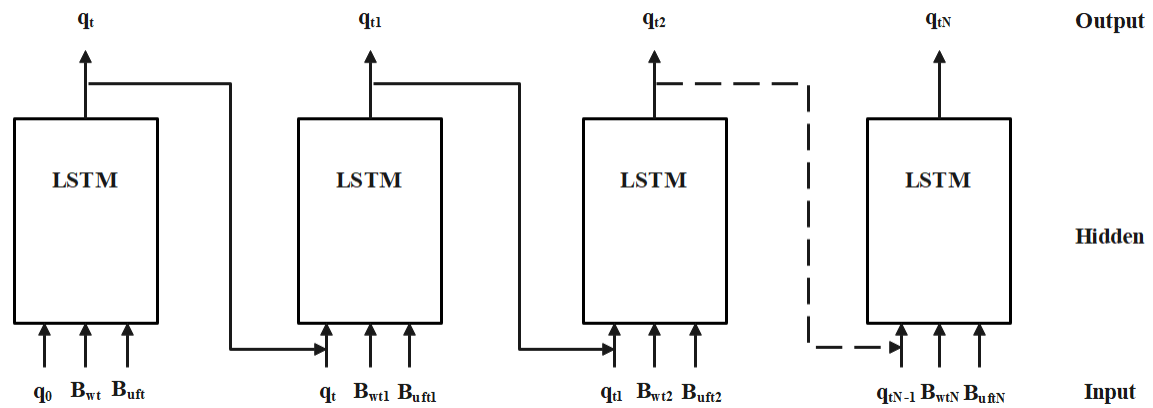


Figure 3. Agent architecture

3.6.1. Agent training

The training process of the proposed DRL is explained in this section. Agent of RL is trained in varied environments which includes statistics of bandwidth and videos encoded at various levels of quality along various counts of segments. In the offline stage, training is performed through two identical agents, major and secondary agents. The rewards and penalties are utilized for training the agents and the major aim is to reduce factor distance among the following segments, protect rebuffering events, and choose huge quality segments to download. In the following training stage, two agents are utilized for streaming data in various environments. When a streaming session happens, the main agent makes a decision and receives rewards but secondary agent stays in passive mode. Then, secondary agent alone reacts while that receives an alert message through a significant agent which represents a need for correcting the decision. In this research, the reinforce policy gradient algorithm is deployed for optimizing the policy of the agent and updating the parameter θ of policy. Its mathematical formula is represented in (9).

$$\pi_\theta(s, a) = P[a|s; \theta] \quad (9)$$

where, $\pi_\theta(s, a)$ describes a policy function, here π_θ represents a policy, s represents a state and a represents an action. $P[a|s; \theta]$ represents the possibility of taking action a in a given state s below policy π_θ . A significant objective is to enhance a collected reward that is represented as R_t and mathematical formula is shown in (10),

$$R_t = \sum_t \gamma^t * r_t \quad (10)$$

where, R_t represents the collected reward which is calculated through summing of rewards acquired at t time step where every reward is discounted through γ^t . That permits an agent to reflect long-term rewards of their actions, providing much weight to rewards and slowly minimizing an effect of further rewards as t rises. The reward gradient with regards to θ parameter is represented as ∇_θ with its expression given in (11),

$$R_t = \nabla_\theta E[\sum_t \gamma^t * r_t] = E \sum_t \gamma^t * r_t \nabla_\theta \log \pi_\theta(a_t | s_t) \quad (11)$$

The equation (11) represents a gradient of discounted reward through regard to policy parameter θ in RL. It is crucial in policy gradient techniques that are utilized for optimizing a policy of the agent to enhance expected rewards. Afterward every episode, a parameter θ of policy is updated by utilizing α learning rate, is expressed numerically (12),

$$\theta \leftarrow \theta + \alpha \sum_t \nabla_\theta \log(\pi_\theta(s_t, a_t)) (\sum_t r_t) \quad (12)$$

The equation (12) represents how a parameter θ is updated afterwards every episode in RL setting and describes the updated rule utilized in a policy gradient technique. The amount of iterations relates to a whole amount of video segments. Initially, method takes input as quality of the initial downloaded segment, present bandwidth, and buffer state. Next, execution and production of the following segment's quality takes place depending on the inputs. The procedure iteratively continues with past quality influencing a process of decision-making till the segment of the last video is reached. The proposed DASH based DRL remarkably improves the QoE and provides a highly stable video quality.

4. RESULTS AND DISCUSSION

The proposed DASH-based DRL algorithm improves the QoE performance, alongside providing highly stable video quality, is simulated in Python environment with system requirements: random access memory (RAM) 16 GB, processor Intel core i7 and operating system Windows 10 (64 bit). The performance of the proposed technique is analyzed on the basis of performance measures of performance improvement (%), QoE metrics, mean opinion score (MOS), normalized value of QoE, average of normalized value of QoE, switching quality and rebuffering time (sec). The simulation parameters and its values are given in Table 2.

Table 2. Values of simulation parameter

Simulation parameter	Values	Description
γ	0.99	Discount factor
λ	-1	Rebuffering penalty
μ	-1	Change of quality penalties

4.1. Quantitative and qualitative analysis

The performance of the suggested technique is analyzed on the basis of performance measures: performance improvement (%), QoE metrics, mean opinion score (MOS), normalized value of QoE, average of normalized value of QoE, switching quality and rebuffering time (sec). Various graphical representations of results are provided in this section to evaluate the suggested algorithm. Figure 4 illustrates the outcomes concerning time with performance improvement. The graph represents a growth in performance as a training procedure presented on x-axis with time in hours. The process is stopped in 60 hours once the required amount of information is attained, wherein higher training time does not enhance the performance. Figure 5 represents the degree of performance with the agents and their impact on performance enhancement. This proves the significance of the collective methods in improving the capabilities of the whole decision-making system and their efficiency in optimizing.

Figure 6 represents the results of QoE metrics of rebuffering time for the suggested DASH-based DRL algorithm. These metrics are essential to evaluate the performance of the proposed algorithm. From

Figure 6, it is evident that the number of rebuffering events reduced by experiments enhance the whole QoE values using video sequences. Figure 7 represents the results of MOS methods, where is it seen that a proposed algorithm attains an mean MOS value of 5 which is a superior range. This score expresses a huge level of user satisfaction through their experience of video streaming. Figure 8 illustrates a results of normalized QoE value, wherein the proposed algorithm attains high values of the normalized QoE value.

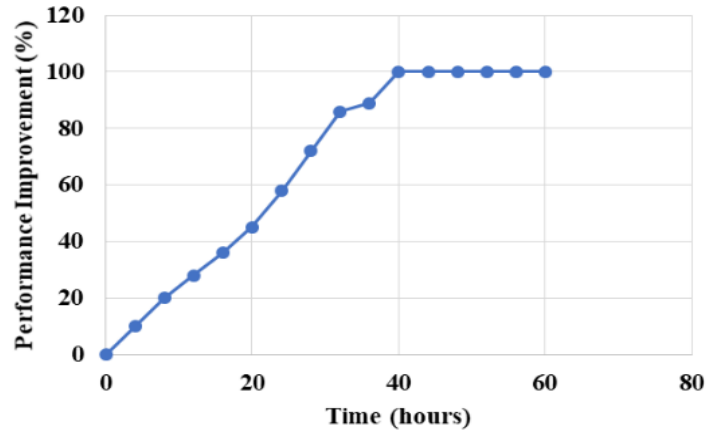


Figure 4. Performance Improvement with respect to time

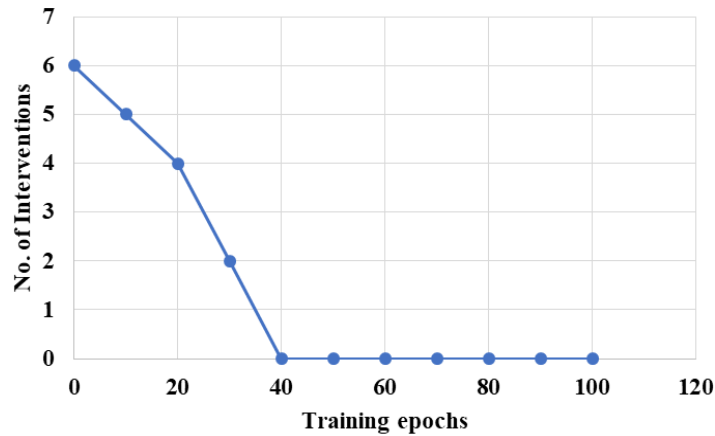


Figure 5. Performance improvement with an agent

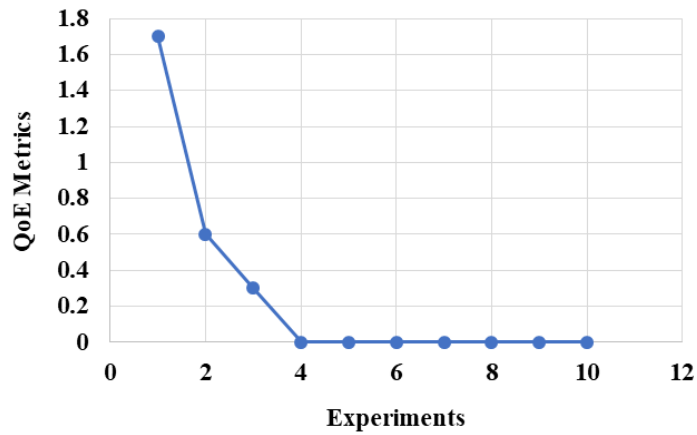


Figure 6. Results of QoE metrics

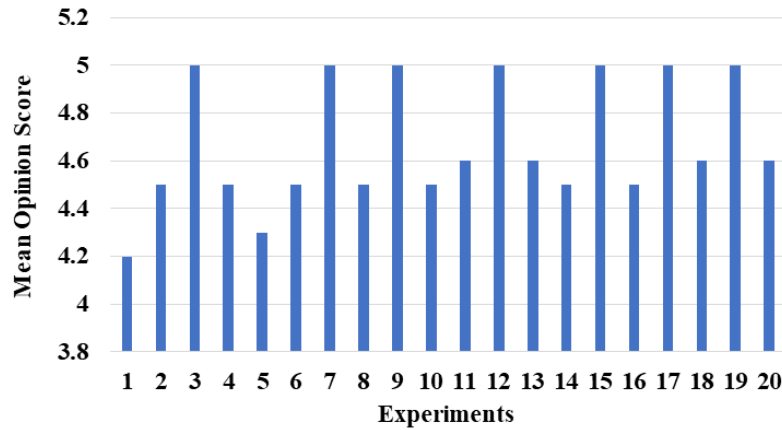


Figure 7. Results of mean opinion score

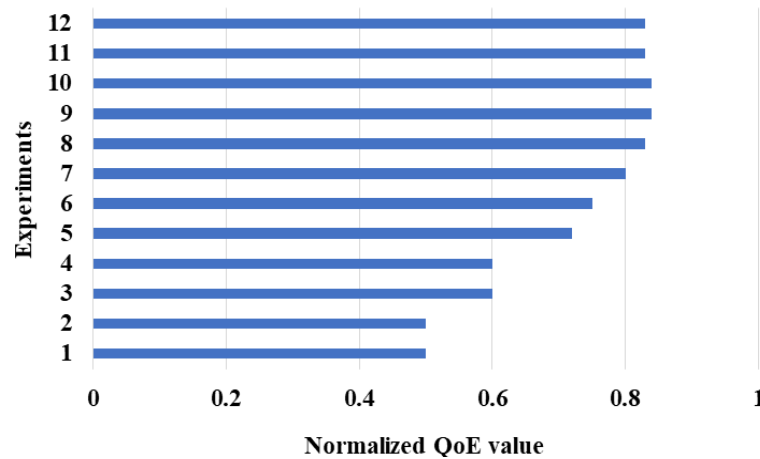


Figure 8. Results of normalized QoE value

Table 3 and Figure 9 present the performance of the introduced DASH-based DRL algorithm evaluated against the existing algorithms namely, real-time streaming protocol (RTSP), HTTP live streaming (HLS) and reinforcement learning (RL). The performance metrics utilized for comparison are average normalized value of QoE, switching quality, and buffering time in seconds. The DASH-based DRL algorithm attains a high normalized QoE value of 0.72 with less switching quality of 0.15 and an average buffering time of 0.16 seconds, which is superior when compared to the existing algorithms.

4.2. Comparative analysis

The proposed DASH based DRL is compared with other existing methods namely, ABRaider [18], DASH based HTTP [19] and FA-MASAC [20] which is presented in Table 4. The performance metrics of normalized average QoE and Performance Improvement over time are taken for the assessment of the introduced algorithm. From Table 4, it is evident that the proposed algorithm attains an improved normalized average QoE of 0.72 and 58% of performance improvement, with respect to time in 24 hours. The introduced algorithm performs preferably in contrast to the previous methods. Here, the ABRaider [18] attains a normalized average QoE of 0.6, while that of the DASH-based HTTP and FA-MASAC [20] are 0.6 and 0.65, respectively.

Table 3. Performance of proposed DASH based DRL

Methods	Average normalized QoE value	Average switching quality	Average buffering time (s)
RTSP	0.49	0.45	0.42
HLS	0.54	0.38	0.36
RL	0.65	0.22	0.28
DASH based DRL	0.72	0.15	0.16

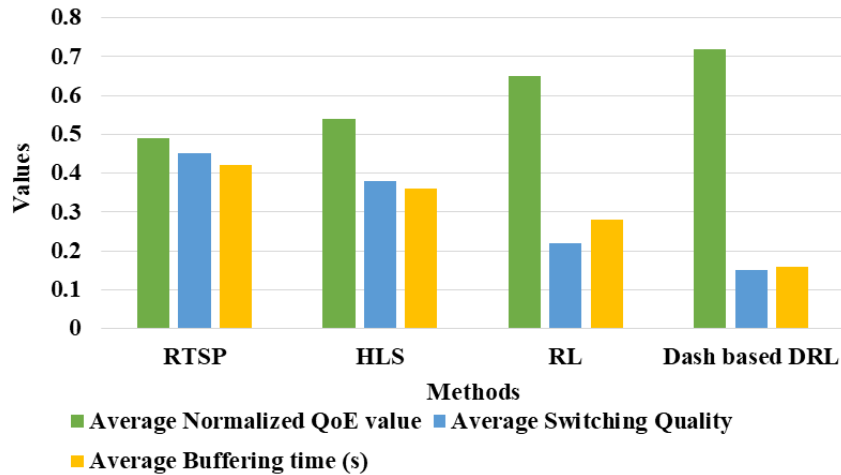


Figure 9. Performance of proposed DASH based DRL

Table 4. Comparative analysis of DASH based DRL

Author	Method	Normalized Average QoE	Performance Improvement over time (%)
Budati <i>et al.</i> [18]	ABRaider	0.6	N/A
Li <i>et al.</i> [19]	DASH based HTTP	0.6	20
Choi <i>et al.</i> [20]	FA-MASAC	0.65	N/A
Proposed	Dash based DRL	0.72	58

4.3. Discussion

The existing methods in DASH video streaming on heuristics and ML methods are utilized for making a selection of bitrate decisions. Though, these methods contain less performance and are unable to employ the dynamic conditions of the network efficiently. To overcome these limitations, the DRL algorithm is proposed in this research that enhances the video quality and QoE performance. Initially, the video quality is classified into 3 different categories and the distance factors are described among two successive video segments. Next, LSTM is utilized to determine the optimum quality for the following video segment where rewards are received continuously till every segment is downloaded. The proposed algorithm significantly enhances the performance of QoE, providing a highly stable video quality, and reduces the distance factor with an ensured smooth streaming session. In this section, findings of the introduced algorithm along with the disadvantages of previous algorithms are described. The ABRaider [18] method has the drawback of transmitting poor-quality images due to bandwidth constraints. The DASH-based HTTP [19] method has difficulties in training because of insufficient data, and the FA-MASAC [20] method covers a small area but faces a problem of redundancy. From the Table 4, it is clear that the proposed method performed well than other existing methods compared like ABRaider [18], DASH-based HTTP [19] and FA-MASAC [20].

5. CONCLUSION




This research proposes an ABR method that depends on deep reinforcement learning (DRL) named DASH-based DRL to determine the segment quality in DASH video streaming with wireless networks. The main objective of the introduced algorithm to give a high quality user experience. Proposed algorithm significantly improves the QoE performance, aside from providing highly stable video quality with a reduced distance factor and endured smooth streaming sessions. The performance of proposed technique is analyzed with performance measures of Performance improvement (%), QoE metrics, MOS, normalized value of QoE, average of normalized value of QoE, switching quality, and rebuffering time (sec). The proposed algorithm accomplishes a high average normalized QoE of 0.72, average switching quality of 0.15, and average rebuffering time of 0.16 seconds, which are all preferable than other algorithms like RTSP, HLS and RL.

Furthermore, the proposed methodology has limitations like the user experience may be degraded if wrong actions are chosen in the initial phase. In future, in order to maximize the convergence speed of the method, the sharing parameters among updated methods in various network environments will have been developed.




REFERENCES

- [1] M. W. A. Ashraf, C. Huang, A. Raza, K. Sharif, M. M. Karim, and S. Huang, "Forwarding and caching in video streaming over ICSDN: A clean-slate publish-subscribe approach," *Computer Networks*, vol. 219, p. 109433, Dec. 2022, doi: 10.1016/j.comnet.2022.109433.
- [2] A. del del Río, J. Serrano, D. Jimenez, L. M. Contreras, and F. Alvarez, "A deep reinforcement learning quality optimization framework for multimedia streaming over 5G networks," *Applied Sciences*, vol. 12, no. 20, p. 10343, Oct. 2022, doi: 10.3390/app122010343.
- [3] R. Farahani, M. Shojafar, C. Timmerer, F. Tashtarian, M. Ghanbari, and H. Hellwagner, "ARARAT: A collaborative edge-assisted framework for HTTP adaptive video streaming," *IEEE Transactions on Network and Service Management*, vol. 20, no. 1, pp. 625–643, Mar. 2023, doi: 10.1109/tnsm.2022.3210595.
- [4] M. O. Elbasheer, A. Aldegheshem, N. Alrajeh, and J. Lloret, "Video streaming adaptive qos routing with resource reservation (VQoSRR) model for SDN networks," *Electronics*, vol. 11, no. 8, p. 1252, Apr. 2022, doi: 10.3390/electronics11081252.
- [5] S. Lee, J.-B. Jeong, and E.-S. Ryu, "Group-based adaptive rendering system for 6DOF immersive video streaming," *IEEE Access*, vol. 10, pp. 102691–102700, 2022, doi: 10.1109/access.2022.3208599.
- [6] J. Li, C. Zhang, Z. Liu, R. Hong, and H. Hu, "Optimal volumetric video streaming with hybrid saliency based tiling," *IEEE Transactions on Multimedia*, vol. 25, pp. 2939–2953, 2023, doi: 10.1109/tmm.2022.3153208.
- [7] M. T. Sultan and H. El Sayed, "QoE-aware analysis and management of multimedia services in 5G and beyond heterogeneous networks," *IEEE Access*, vol. 11, pp. 77679–77688, 2023, doi: 10.1109/access.2023.3298556.
- [8] P. Lebreton and K. Yamagishi, "Quitting ratio-based bitrate ladder selection mechanism for adaptive bitrate video streaming," *IEEE Transactions on Multimedia*, vol. 25, pp. 8418–8431, 2023, doi: 10.1109/tmm.2023.3237168.
- [9] C.-H. Lin, G.-H. Hu, J.-S. Chen, J.-J. Yan, and K.-H. Tang, "Novel design of cryptosystems for video/audio streaming via dynamic synchronized chaos-based random keys," *Multimedia Systems*, vol. 28, no. 5, pp. 1793–1808, May 2022, doi: 10.1007/s00530-022-00950-6.
- [10] M. Utke, S. Zadootaghaj, S. Schmidt, S. Bosse, and S. Möller, "NDNetGaming - development of a no-reference deep CNN for gaming video quality prediction," *Multimedia Tools and Applications*, vol. 81, no. 3, pp. 3181–3203, Jul. 2020, doi: 10.1007/s11042-020-09144-6.
- [11] C. Lee, S.-G. Kang, and A. Nayyar, "Location-proximity-based clustering method for peer-to-peer multimedia streaming services with multiple sources," *Multimedia Tools and Applications*, vol. 81, no. 16, pp. 23051–23090, May 2021, doi: 10.1007/s11042-021-10985-y.
- [12] M. S. Coelho, C. A. V. Melo, and N. L. S. da Fonseca, "An encoding-aware bitrate adaptation mechanism for video streaming over HTTP," *Multimedia Tools and Applications*, vol. 81, no. 19, pp. 27423–27451, Mar. 2022, doi: 10.1007/s11042-022-12520-z.
- [13] M. Taha, "An efficient software defined network controller based routing adaptation for enhancing QoE of multimedia streaming service," *Multimedia Tools and Applications*, vol. 82, no. 22, pp. 33865–33888, Mar. 2023, doi: 10.1007/s11042-023-14938-5.
- [14] S. Lee and J. Yoo, "Reinforcement learning based multipath QUIC scheduler for multimedia streaming," *Sensors*, vol. 22, no. 17, p. 6333, Aug. 2022, doi: 10.3390/s22176333.
- [15] P. M. Ashok Kumar, L. N. Arun Raj, B. Jyothi, N. F. Soliman, M. Bajaj, and W. El-Shafai, "A novel dynamic bit rate analysis technique for adaptive video streaming over HTTP support," *Sensors*, vol. 22, no. 23, p. 9307, Nov. 2022, doi: 10.3390/s22239307.
- [16] D. Samiyya, J. Ramasamy, and M. Gunasekar, "An efficient congestion control in multimedia streaming using adaptive BRR and fuzzy butterfly optimization," *Transactions on Emerging Telecommunications Technologies*, vol. 34, no. 3, Jan. 2023, doi: 10.1002/ett.4707.
- [17] G. Xiong, X. Qin, B. Li, R. Singh, and J. Li, "Index-aware reinforcement learning for adaptive video streaming at the wireless edge," in *Proceedings of the Twenty-Third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, Oct. 2022, pp. 81–90, doi: 10.1145/3492866.3549726.
- [18] A. K. Budati, S. Islam, M. K. Hasan, N. Safie, N. Bahar, and T. M. Ghazal, "Optimized visual internet of things for video streaming enhancement in 5G sensor network devices," *Sensors*, vol. 23, no. 11, p. 5072, May 2023, doi: 10.3390/s23115072.
- [19] J. Li, H. Zhang, and H. Ma, "DRL-based transmission control for QoE guaranteed transmission efficiency optimization in tile-based panoramic video streaming," *Multimedia Systems*, vol. 29, no. 5, pp. 2761–2777, Jun. 2023, doi: 10.1007/s00530-023-01129-3.
- [20] M. H. Jofri, M. F. Md Fudzee, M. N. Ismail, S. KASIM, and J. Abawajy, "Quality of experience (QOE) aware video attributes determination for mobile streaming using hybrid profiling," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 8, no. 3, pp. 597–609, Dec. 2017, doi: 10.11591/ijeecs.v8.i3.pp597-609.
- [21] W. Choi, J. Chen, and J. Yoon, "ABRaider: Multiphase reinforcement learning for environment-adaptive video streaming," *IEEE Access*, vol. 10, pp. 53108–53123, 2022, doi: 10.1109/access.2022.3175209.
- [22] J. Kang and K. Chung, "HTTP adaptive streaming framework with online reinforcement learning," *Applied Sciences*, vol. 12, no. 15, p. 7423, Jul. 2022, doi: 10.3390/app12157423.
- [23] J. Zeng, X. Zhou, and K. Li, "MADRL-based joint edge caching and bitrate selection for multicategory 360° video streaming," *IEEE Internet of Things Journal*, vol. 11, no. 1, pp. 584–596, Jan. 2024, doi: 10.1109/jiot.2023.3287187.
- [24] Y. Wang, M. Agarwal, T. Lan, and V. Aggarwal, "Learning-based online QoE optimization in multi-agent video streaming," *Algorithms*, vol. 15, no. 7, p. 227, Jun. 2022, doi: 10.3390/a15070227.
- [25] J. Li *et al.*, "Toward optimal real-time volumetric video streaming: A rolling optimization and deep reinforcement learning based approach," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 12, pp. 7870–7883, Dec. 2023, doi: 10.1109/tcsvt.2023.3277893.
- [26] X. Tang, F. Chen, and Y. He, "Intelligent video streaming at network edge: An attention-based multiagent reinforcement learning solution," *Future Internet*, vol. 15, no. 7, p. 234, Jul. 2023, doi: 10.3390/fi15070234.
- [27] E. I. Iroegbu and D. Madhavi, "Accelerating the training of deep reinforcement learning in autonomous driving," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 3, p. 649, Sep. 2021, doi: 10.11591/ijai.v10.i3.pp649-656.
- [28] I. Onyegbadue, C. Ogbuka, and T. Madueme, "Robust least square approach for optimal development of quadratic fuel quantity function for steam power stations," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 2, pp. 732–740, Feb. 2022, doi: 10.11591/ijeecs.v25.i2.pp732-740.
- [29] B. Talafha, A. Abuammar, and M. Al-Ayyoub, "Atar: Attention-based LSTM for Arabizi transliteration," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 3, pp. 2327–2334, Jun. 2021, doi: 10.11591/ijece.v11i3.pp2327-2334.

BIOGRAPHIES OF AUTHORS

Manjunatha Peddareddygari Bayya Reddy    is currently pursuing Ph.D in Visvesvaraya Technological University (VTU). He received M.E degree from Bangalore University (UVCE) and Bachelor of Engineering (B.E.) degree from Visvesvaraya Technological University (VTU). He presently holds the position of assistant professor in the Department of Artificial Intelligence and Machine Learning, Bangalore Institute of Technology. He has 13 years of experience in teaching and research. His research interests include networking, machine learning, and data analytics. He can be contacted at email: manjubb.reddy@gmail.com.



Sheshappa Shagathur Narayanappa    is an associate professor in the Department of Information Science and Engineering, Sir M Visvesvaraya Institute of Technology, Bangalore, Karnataka, India. The author has completed Ph.D from GITAM University, Andhra Pradesh. He has 23 years of experience in teaching and research. His research interests include computer networking, machine learning, and data analytics. He has many publications and is presently working on many more papers. He can be contacted at email: sheshappa_is@sirmvit.edu.