Computers, Materials & Continua
DOI:10.32604/cmc.2022.022236
Article

Tech Science Press

Encoder-Decoder Based LSTM Model to Advance User *QoE* in 360-Degree Video

Muhammad Usman Younus[1,*], Rabia Shafi[2], Ammar Rafiq[3], Muhammad Rizwan Anjum[4], Sharjeel Afridi[5], Abdul Aleem Jamali[6] and Zulfiqar Ali Arain[7]

[1]Ecole Doctorale Mathematiques, Informatique, Telecommunication, de Toulouse, University Paul Sabatier, Toulouse, 31330, France
[2]School of Electronics and Information, Northwestern Polytechnical University, Xi'an, 710129, China
[3]Department of Computer Science, NFC Institute of Engineering and Fertilizer Research, Faisalabad, 38000, Pakistan
[4]Department of Electronic Engineering, The Islamia University of Bahawalpur, Bahawalpur, 63100, Pakistan
[5]Department of Electrical Engineering, Sukkur IBA University, Sukkur, 65200, Pakistan
[6]Department of Electronic Engineering, Quaid-e-Awam University of Engineering, Science and Technology (QUEST), Nawabshah, 67450, Pakistan
[7]Department of Telecommunication Engineering, MUET, Jamshoro, 76060, Pakistan
*Corresponding Author: Muhammad Usman Younus. Email: usman1644@gmail.com
Received: 31 July 2021; Accepted: 07 September 2021

Abstract: The development of multimedia content has resulted in a massive increase in network traffic for video streaming. It demands such types of solutions that can be addressed to obtain the user's *Quality-of-Experience* (*QoE*). 360-degree videos have already taken up the user's behavior by storm. However, the users only focus on the part of 360-degree videos, known as a viewport. Despite the immense hype, 360-degree videos convey a loathsome side effect about viewport prediction, making viewers feel uncomfortable because user viewport needs to be pre-fetched in advance. Ideally, we can minimize the bandwidth consumption if we know what the user motion in advance. Looking into the problem definition, we propose an Encoder-Decoder based Long-Short Term Memory (LSTM) model to more accurately capture the non-linear relationship between past and future viewport positions. This model takes the transforming data instead of taking the direct input to predict the future user movement. Then, this prediction model is combined with a rate adaptation approach that assigns the bitrates to various tiles for 360-degree video frames under a given network capacity. Hence, our proposed work aims to facilitate improved system performance when *QoE* parameters are jointly optimized. Some experiments were carried out and compared with existing work to prove the performance of the proposed model. Last but not least, the experiments implementation of our proposed work provides high user's *QoE* than its competitors.

Keywords: Encoder-decoder based lSTM; 360-degree video streaming; LSTM; *QoE*; viewport prediction

1 Introduction

Recently, 360-degree video has achieved a great importance in multimedia streaming. Employing adaptive streaming for 360-degree video content is always being a challenge due to the lack of dedicated streaming and encoding techniques. According to [1], the Compound Annual Growth Rate (CAGR) of 360-degree camera industry is expected to grow by 34% between 2018 to 2024. Therefore, there is no indication of slowing down 360-degree video wearing an Head-Mounted Display (HMD) device in the coming years. However, it becomes difficult to apply an HMD to stream 360-degree video so far. Providing 360-degree videos is challenging for the following reasons:

- The principal challenge in deploying effective 360-degree video streaming technology is the huge data amount than the conventional ones, and thus 360-degree videos are encoded at higher bitrates with higher resolutions. Such types of videos are necessary to offer a genuine immersive experience.
- When 360-degree video is transmitted, its bandwidth consumption is up to 4–6 times that of traditional video. In addition, HMDs need a higher resolution (usually 4K or even 6K) for a good viewing experience.
- HMD cannot be shared with other viewers, so it is possible to have multiple 360-degree video streaming even in a small room.

Although many improvements have been made in video coding, computing, and networking, the community still needs to promote improved solutions to address the issues listed above [2]. It is being challenging to transmit the whole 360-degree video to users because of time-variant features and *QoE* objectives. The former may have an impact on decision-making process of tiles, e.g., network conditions and viewport locations. While the latter one includes the different user's *QoE* factors, i.e., user's perceived quality, rebuffering, temporal and spatial quality variance. As a result, providing a good immersive experience for 360-degree video streaming is always difficult due to their vulnerability to inconsistent and insufficient bandwidth.

A small portion of the video, termed the viewport, is transmitted at the highest resolution in tile-based viewport-adaptive 360-degree video streaming [3], while the rest of the video is provided at lower resolutions. Because 360-degree content is mainly consumed through HMD devices that have limited Field of View (FoV), e.g., 900° vertically and 1100° horizontally. As a result, merely streaming the user's viewport at high resolution is an effective rate-saving strategy. Hence, such solutions adjust the video quality by dynamically selecting regions to minimize the transmitted bitrate while ensuring user's *QoE*.

Following this idea, adaptive streaming for 360-degree video content has to face some challenges, mainly involving viewport prediction and rate adaptation issues. The authors in [4,5] have done great work to improve the prediction accuracy for long-term viewport prediction. Therefore, we tried to develop a rate adaptation technique that considers the prediction errors to optimize video segments bitrates temporally to determine the bitrate allocation for spatial tiles. Furthermore, a trade-off

exists between bandwidth efficiency and video quality to obtain the optimal user's*QoE*. The main contributions are as follow:

1. Studies on viewport prediction in existing literature are still minimal. This paper takes an agnostic machine learning-based prediction model to make future predictions. For viewport prediction, we have proposed Encoder-Decoder based LSTM model where the user's viewport information is examined for the future viewpoint that can vary with buffer occupancy. This model takes the transforming data instead of taking the direct input to predict the future user movements.
2. Based on the proposed long-term viewport prediction model, the client assigns bitrates to each of the tiles as a non-linear optimization issue based on different parameters, namely motion and saliency map, maximizing the user's *QoE.*. Therefore, we propose a rate adaptation algorithm based on predicted viewport using Reinforcement Learning (RL) policy.
3. We have evaluated the experiments of each part of our proposed system separately, for example, viewport prediction and rate adaptation, maximizing the user's *QoE* based on step (1) and step (2). Our experimental results outperform than other comparative schemes.

The paper's layout is arranged as follows: Section 2 defines the related work where Machine Learning (ML) based approaches for viewport prediction and rate adaptation have presented. Section 3 explains the system design, including Encoder-Decoder based LSTM model for viewport prediction and rate adaptation algorithm. However, Section 4 describes the performance evaluation. Section 5 illustrates the discussion about the paper. Finally, Section 6 summarizes the whole paper.

2 Related Work

2.1 ML-Based Techniques for Viewport Prediction

Viewport prediction is one of the challenges of adaptive 360-degree video streaming. Regression-based methods have been studied by [6,7] to estimate the user's future head rotations. However, these studies do not consider any video content and require an existing dataset of user viewports. Another study [8] used an RL-based model to optimize the user's *QoE* by predicting the viewport. But they also do not take into account the video content and different bitrates for the predicted viewport tiles. While [9] proposed an effective technique for viewport prediction to use the user's viewport as video content based on the trajectories of primary objects.

The viewport prediction is always being a vital enabler for 360-degree videos, which improves the prediction accuracy. In near future, the user's head rotation can be predicted with high accuracy but accurate long-term predictions remain elusive. The authors in [10] extract the content-related features from the current frame and predict the next viewport based on the saliency algorithm. Moreover, this model does not work to consider the user's viewing behavior. Also. it fails to capture the properties, i.e., non-linearity and long-term dependency, resulting in undesirable performance regarding the prediction accuracy.

Great efforts have been made on the saliency map concept that shows image characteristics to examine the video content based on their probability distribution function. In [11], PanoSalNet has learned the saliency map from user viewport data by employing a fixation prediction network. They train an image saliency network on their 360-degree video viewing dataset. In [12], the authors combined a viewport prediction model with a rate control strategy to determine a tile probability map using head movement data as input for *QoE.* optimization under given network capacity. Although, a lot of training data is needed to learn the saliency map using user's head tracking data that makes the

model sensitive extending to new videos. Such proposed system does not need to update parameters during streaming and is not suitable for adaptive video streaming. This results in a lack of user dynamic adaptation.

2.2 ML-Based Techniques for Rate Adaptation

This section defines the challenges that need to be addressed by our proposed customized approach for rate adaption of 360-degree video streaming. The efficient delivery of the image through a network is always being a challenge. If the whole 360 image has to be delivered, it demands high network bandwidth for the content provider and the end-user. Though, not all the data is consumed equally. As the viewer faces a specific direction at any given time while watching a 360-degree video. Therefore, the 20% of the transmitted data is consumed by the viewer.

ML has advanced quickly, and its performance when combined with image processing and big data is outstanding. To address rate adaptation issues, the data-driven techniques have recently been developed. The authors in [13] proposed two DRL models to predict head motion considering the motion trajectories and visual frames. Their deep neural network only receives the user's view of interest and decides which direction and viewer's head will move. A saliency-driven model [10] extracts the content-related features from the current frame and also predicts the next viewport depending on the saliency algorithm. But the user's viewing behavior is not considered by this work. Also, it fails to capture the properties, i.e., non-linearity and long-term dependency, resulting in undesirable performance. In [13], DRL-based adaptive approach has been proposed for multiple tiles to minimize the decision space of rate allocation, enabling the rate adaptive algorithm for maximizing the user's *QoE*. Some authors [14,15] use SDN and RL for the improving the network performance.

Moreover, a RL-based rate adaptation algorithm in [16,17] is proposed to determine the user's viewport depending on buffer occupancy to improve the bandwidth efficiency. A rate adaptation issue has been formulated as a non-linear optimization issue in 360-degree video for maximizing the user's *QoE*. A Q-learning-based algorithm in [18] has also been proposed to minimize the decision space by defining the rate allocation strategy for multiple tiles to optimize the user's *QoE*.

3 System Design

In this section, the need of viewport prediction in a 360-degree video streaming system has been discussed. We have used an RL agent to learn a streaming policy to understand the adaptive user's behavior and to adapt the dynamic network behavior. Fig. 1 depicts the proposed architecture of viewport adaptive streaming for 360-degree video. There is no need to stream the entire video at the highest resolution because a user only watches a small portion of the video at one time. While ensuring the user quality, the region is dynamically selected according to the user's head movement and the quality is adjusted to minimize the transmission bitrate that identifies the corresponding viewport of the end users. The core of the proposed system is an RL agent that find the downloading bitrate for each tile. The prediction was modeled as a probability distribution over all possible tiles. The proposed system aims to choose the tiles along with their bitrates to fetch the next segment under the given network capacity. Hence, the agent tries to maximize the user's *QoE* by following the sequential decision process. For better understanding, the detail of each part is explained separately, as follows:

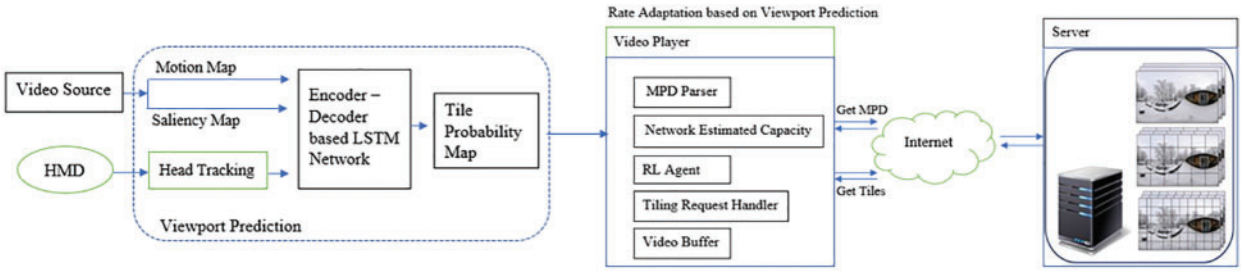

Figure 1: Overview of our proposed design system for 360-degree video streaming

3.1 Viewport Prediction

This section elaborates the need of viewport prediction in 360-degree videos. Our prediction model's output calculates the probability of different tiles to indicate how likely a tile is viewed by a user. A trade-off exists between video resolution and accuracy of viewport prediction that must be integrated in 360-degree video streaming system. Thus, this unique attribute of 360-degree videos saves the network bandwidth significantly. To address the above-mentioned challenges, we need to predict the viewport with high accuracy, otherwise the user's quality declines.

Furthermore, the viewport prediction depends on the fact that users tend to focus on interesting salient features. These characteristics can be revealed by the video analysis for viewport prediction in future. We have used the ML-based approaches for viewport prediction. The main goal of our proposed work is to investigate whether Encoder-Decoder based LSTM model can be leveraged to improve the predictions about user's viewport. This system identifies the content-based features (for example, image saliency detection and motion detection) from a 360-degree video, as well as sensor-based features that provide HMD orientation information. The components are listed below in architecture of proposed viewport prediction model.

1. **Content Combination of Detected Features**: It provides information related to the detected features such as image saliency and motion detection. The saliency network derives those parts of the image that are more attractive to the viewer. This describes the objects that make the distinct differences in the features (e.g., color, texture, etc.). While Lucus-Kanade optical flow [19] detects the consecutive frames, the moving objects may attract the viewers.
2. **Orientation Extractor:** This allows a viewer's orientation data, which includes yaw, pitch, and roll, to pass to the HMD sensor, where it is then concatenated and fed into the network. In our work, we only make predictions based on viewpoint for yaw and pitch angles since, as stated in [20], roll angle is mostly considered zero. Since yaw and pitch angles are treated as independent variables for prediction, it was found that they have a very strong auto-correlation.
3. **Prediction Network:** We use Encoder-Decoder based LSTM model as a predictor. This model makes viewport prediction based on the viewpoint information obtained by users. It captures the spatial and temporal features of 360-degree video. We have used the users' head tracking data so that can be feed into Encoder-Decoder based LSTM model for computing the tile probabilities through modelling the spatial region.

Fig. 2 defines the viewport prediction model based on Encoder-Decoder based LSTM, taking into account the key factors to adapt to users' viewing state and defining the architecture of the 360-degree video streaming system. Our proposed Encoder-Decoder based LSTM model is commonly used model for deep learning and uses the previous output to predict the next output to extract and

learn the features automatically. Our prediction model is very light in computation and predicts the next user's viewport, as it is considered as a temporal function of user head movement.

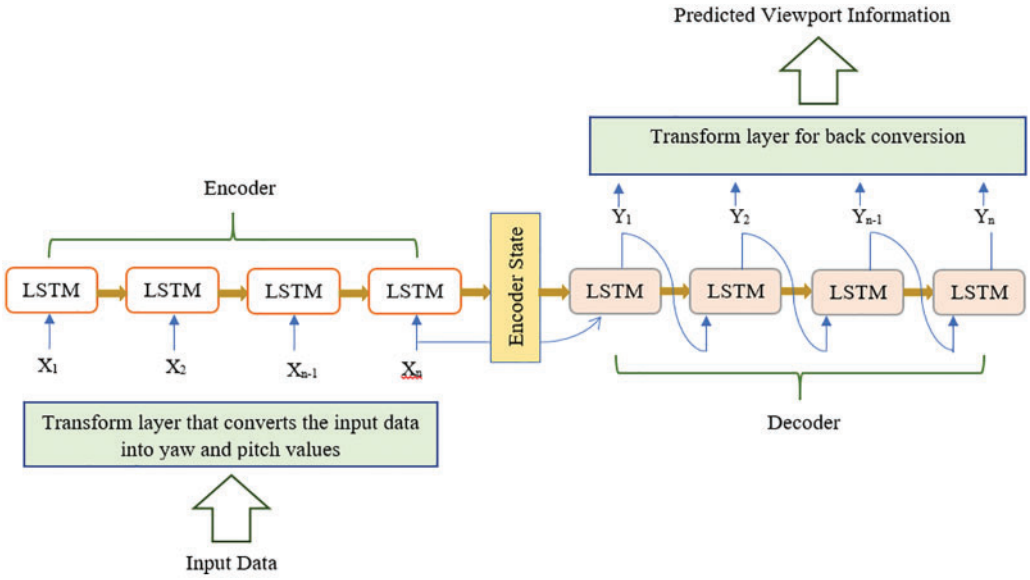

Figure 2: Encoder-Decoder based LSTM model for viewport prediction

The input layer of Encoder-Decoder based LSTM model takes the input data to transform it into yaw and pitch values before inputting into the encoding layer of the proposed model by considering the roll angle to zero, as shown in Eq. (1). The encoder's hidden layer will initialize the decoder, and its initial input is the encoder's general output that generates the hidden states in future times. The proposed model is trained to minimize the prediction error.

$$\begin{bmatrix} yaw \\ pitch \\ roll \end{bmatrix} = \begin{bmatrix} u\tan 2(2(v_3v_0 + v_1v_2), 1 - 2(v_0v_0 + v_1v_1)) \\ u\sin(2(v_2v_0 - v_3v_1)) \\ u\tan 2(2(v_3v_2 + v_0v_1), 1 - 2(v_1v_1 + v_2v_2)) \end{bmatrix}, \tag{1}$$

3.2 Rate Adaptation

There have been numerous rate adaptation strategies for non-360-degree videos while our proposed strategy is inspired by Model Predictive Control (MPC)-based rate adaptation [2] that is high-performance and efficient control method. The principle of the MPC algorithm is to predict the future dynamic of a system based on current information that solves a finite-time optimization problem by applying an optimal solution for each sampling moment. In tile-based 360-degree video transmission, the purpose of bitrate selection is to choose the appropriate bitrate for enhancement layers under user's perspective changes and dynamic network bandwidth, maximizing the user's *QoE* within a period of time. Our work considers the modelling of *QoE* metric of each segment. We presented the following parameters that our proposed rate adaptive algorithm will optimize.

1. **User Perceived Video Quality:** It is defined as the sum of the qualities of tiles that the user actually views. Assume that there are M tiles in a 360-degree video scene, $G_{i,j}$ is the rate selected for j^{th} tile of i^{th} video segment. Mathematically, it can be defined as:

$$E_i = \sum_{j=1}^{M} G_{i,j} O_{i,j}, \tag{2}$$

where $O_{i,j}$ is overlapping ratio of viewport V_{i} and predicted tiles T_{j}. $O_{i,j} = 1$ if T_{j} overlaps with V_{i} and 0 otherwise.

2. **Rebuffering Time:** It happens when video duration is less than the downloading time in the buffer for the i^{th} video segment for j^{th} tile, resulting in decreasing the QoE metric. It is observed that the segment size is based on the bitrate to compute the rebuffering time. Let $S_i\ (G_{i,j})$ represents the size of the i^{th} video segments for bitrate $G_{i,j}$. T_i is the difference between the download time of a video segment and the buffering time that is obtained by the following formula:

$$T_i = \sum_{j=1}^{M} \left(\frac{S_i(G_{i,j})}{C_i} - B_i \right), \tag{3}$$

Here, C_i represents the predicted network bandwidth of downloading the i^{th} video segment, B_i indicates the video buffer duration of downloading the i^{th} video segment, and$(x)_+ = \max(0, x)$.

3. **Temporal Quality Oscillations:** The disparity between two viewports of consecutive segments can reduce the efficiency of 360-degree video streaming. As a result, the changes in viewport quality should not be significant, and can be determined as follows:

$$D_1(i) = |E_i - E_{i-1}|, \tag{4}$$

4. **Spatial Quality Oscillations:** The inconsistent quality levels within the viewport might cause cybersickness and other physiological symptoms, e.g., aversion and nausea. The user's QoE will decrease if the video content is not smooth. This value is calculated using *Coefficient of Variation (CV)* as follows:

$$D_2(i) = \sum_{j=1}^{M} CV \tag{5}$$

The user perceived QoE for each 360-degree video segment can be defined by a weighted summation formulation as follows:

$$QoE = E_i - \mu_1 T_i - \mu_2 D_1(i) - \mu_3 D_2(i), \tag{6}$$

where, μ_1, μ_2, and μ_3 are non-negative constants according to the rebuffering time, temporal and spatial quality oscillations, respectively. $D_1(i)$ *and* $D_2(i)$ have been considered as negative to achieve the chunk smoothness while a relatively small μ_1 means that the user is less sensitive to video freezing.

With QoE metrics defined, our purpose is to assign the rate $G_{i,j}$ for each tile T_j to maximize the problem of rate adaptation algorithm. Our proposed strategy allows for re-downloading of tiles with different encoding qualities under good network conditions. If the tile is not selected to be fetched then

$Zj = 1$ and $G_{i,j} = 0$. As a result, we aim to penalize this undesired behavior by providing an immediate reward of -1. Mathematically, the missing tiles are denoted as follows:

$$U_i = -\sum_{j=1}^{M} P_j r(S_{i,j}(\min)).Z_j, \tag{7}$$

where P_j the probability that indicate how likely the tiles T_j are viewed by a user. Hence, the optimization issue to be solved by defining a reward r_i for i^{th} video segment for j^{th} tile depending on user's QoE considering the missing tiles. Thus, the user perceived QoE for each 360-degree video segment can be defined by a weighted summation formulation:

$$QoE = E_i - \mu_1 T_i - \mu_2 D_1(i) - \mu_3 D_2(i) + U_i \tag{8}$$

In the proposed rate adaptation algorithm, the 360-degree video is divided into a number of segments. Note that the feasible bitrate of video segments can be chosen by selecting the predicted tiles Tj under the network capacity C. Algorithm 1 provides the pseudo-code.

The algorithm considers both viewport and estimated network capacity C signals, which are used for the next segment's video rate selection on viewport basis. The quality level of i^{th} video segment is decided after downloading $(i-1)th$ segment. Initially, we do not know any kind of information about the network conditions.

The proposed rate adaptation algorithm tries to find the user's viewport Vi to maximize the video quality under network conditions. We have selected $Rates$ as best rate selection. The quality of the overlapping tiles G_j for each viewport is increased and predicted tiles overlap with viewport if $O_{i,j} > 0$ (line 4).

If QoE metric is higher than the available network capacity (line 5) and is not the minimum then it will continue the previous bitrate aggressively. If condition in line 6 does not lie and rate selection is defined under given network capacity C, then QoE metric (Eq. (5)) is evaluated and checked. We reward G to $Rates$ if a new rate selection has the highest quality, indicating that the optimal rate must be modified (line 10). The same procedure is repeated to examine the next viewport.

Algorithm 1: Rate Adaptation based on Viewport Prediction.

Algorithm 1
Input:
Estimate network capacity C, P_j probability of predicted T_j tiles.
Output:
Rates, Selected bitrate for T_j tiles
Initialization:
Rates = 0
max $\leftarrow -\infty$
1 ***While*** $G_{temp} \leftarrow Rates$ //* ***Update rates****//
2 ***for*** *i* $\leftarrow$*1 to viewport* v_i ***do***
3 $G \leftarrow G_{temp}$

(Continued)

Algorithm 1: Continued

If $O_{i,j} > 0$ then $G_j^{\uparrow}$

If

$$\text{Pr}\,ecision = \frac{Number\ of\ \ tiles\ predicted\ correctly}{Number\ of\ \ tiles\ predicted\ correctly + Number\ of\ \ tiles\ predicted\ \ \ Incorrectly}$$

then

Continue the previous bitrates

else

$QoE \leftarrow Equation(5)$

If $QoE > \max$ ***then***

$Rate \leftarrow G$

4 Performance Evaluation

This section details the several experiments we conducted to demonstrate the effectiveness of our proposed technique. The server uses MPEG-DASH streaming system for modelling and evaluating the proposed system by modifying the Python VR client. The player has been written in C^{++} using Android NDK and in Java using Android SDK for tile scheduling and rate adaptation, and tracking head movement, respectively. A trace-driven simulation is created by an open source dataset to employ the real head movement traces collected from 50 users watching 10 different 360-degree videos [21]. Each trace consists of the user's head position, such as yaw, pitch, and roll angles, of which roll angle is considered negligible [22]. Each video segment is 1min long with resolution of 3840 × 1920 in Equi-Rectangular Projection (ERP) format. Each video is divided into small video segments of 2s and encoded with different bitrates such as {300, 700, 1500, 3700, 8500, 20000} kbps using an open source encoder Kvazar[1] encoder. For head tracking data of each pair of videos, we conducted different experiments to evaluate the *QoE* metric. The detail of all hyper parameters has given in Tab. 1.

Table 1: Setting parameters

Parameters	Characteristics
Segment duration	2 s
Resolution	3840 × 1920
Representation set	{300, 700, 1500, 3700, 8500, 20000} kbps
Video segment length	1 min
QoE parameters	$\mu_1 = 1, \mu_2 = 1, \mu_3 = 1$
	$\mu_1 = 1, \mu_2 = 2, \mu_3 = 1$
	$\mu_1 = 1, \mu_2 = 0.3, \mu_3 = 0.3$
Viewport size	100°
Batch size	32
Learning rate	0.002

[1] https://github.com/ultravideo/kvazaar

4.1 Viewport Dataset

We made the viewport prediction based on Encoder-Decoder based LSTM model using PyCharm environment for the same dataset [21]. The prediction model gives the output of the tile probability of next video frames by giving the inputs such as saliency map and head tracking data. The following hyper parameters are used to train our network: batch size (32), Adam optimizer [23], and learning rate (0.002). In the training process, the network was trained for 50 normalized epochs with the ADAM optimizer that corrects the deviations and updates the weights to speed up the convergence during the model training. Our training model has been deployed for all users and is generalizable. We randomly choose 80% processed files for all the videos as training and 20% for validation.

4.2 Network Setup

We have used MP4Box[2] for streaming client environment, adding an interface to rate adaptation and prediction system to examine *QoE* metric. The client connects to the server over the Internet at a speed of up to 50 Mbps and downloads all predicted tiles from the first video segment in order to decode it.

4.3 Evaluation Results for Viewport Prediction

Firstly, the viewport prediction of our proposed Encoder-Decoder based LSTM model is evaluated by comparing it with other methods such as Linear Regression (LR) [24], which predicts the future viewport by fitting all the data points in the sliding window. ATRTIA [25] predicts the future user's viewport depending on 3D-Convolutional Neural Network (3DCNN) by extracting the spatial and temporal characteristics of 360-degree video. Mosaic [12] predicts the future viewpoint information on the basis of CNN and the Recurrent Neural Network (RNN) model, where CNN is used to extract the spatial features. While Flare [26] uses the IBR-approved user study to predict the future viewport instead of downloading the entire panoramic scene. Moreover, we consider precision metric that can be defined as ratio of correctly predicted tiles to both correctly and incorrectly predicted tiles as follows;

$$\text{Pr}\,ecision = \frac{Number\ of\ \ tiles\ predicted\ \ correctly}{Number\ of\ \ tiles\ predicted\ \ correctly + Number\ of\ \ tiles\ predicted\ \ \ Incorrectly} \tag{9}$$

It is noted that the client needs to prefetch some video segments to minimize interruptions in the playback. We will also demonstrate how the proposed work performs for different prediction windows. In the following experiments, we set the prediction window to 1 s, 3 s, and 5 s for evaluating the performance of our proposed Encoder-Decoder-based LSTM model to predict a user's viewpoint information.

All head movement traces collected from the given dataset are applied to the above methods. Fig. 3 shows the distributions of the viewport prediction precision for three prediction window sizes: 1s, 3s, and 5s, across all traces. Fig. 3 shows that each data point on a Cumulative Distribution Function (CDF) curve is the viewport prediction precision for head movement traces. It has found that our proposed scheme performs well in all scenarios as compared to other alternatives. Few tiles are incorrectly predicted to be viewed when precision is higher. Therefore, higher precision is needed for higher bitrates to view the tiles under constrained bandwidth. Besides, various ML algorithms represent the diverse performance behavior for different prediction windows. For instance,

[2] https://gpac.wp.imt.fr/mp4box/

LR performs quite well in Fig. 3a as compared to Fig. 3c, indicating that it is good for short prediction windows but declines in prediction precision as the prediction window grows due to overfitting. As a result, our proposed model outperforms LR in terms of robustness. As far Flare suffers from high stalls due to imperfect viewport prediction and does not perform well for long-term viewport prediction. In Figs. 3a–3c, it can be seen that as prediction time increases, the prediction precision decreases and this degradation can be seen in Flare. However, if we talk about Mosaic, we again find that our proposed work gives better performance because it does not run a prediction system at run-time, putting atypical users at risk of poor video quality. Although, our proposed work also outperforms ATRITA in terms of increasing smoothness within the viewport.

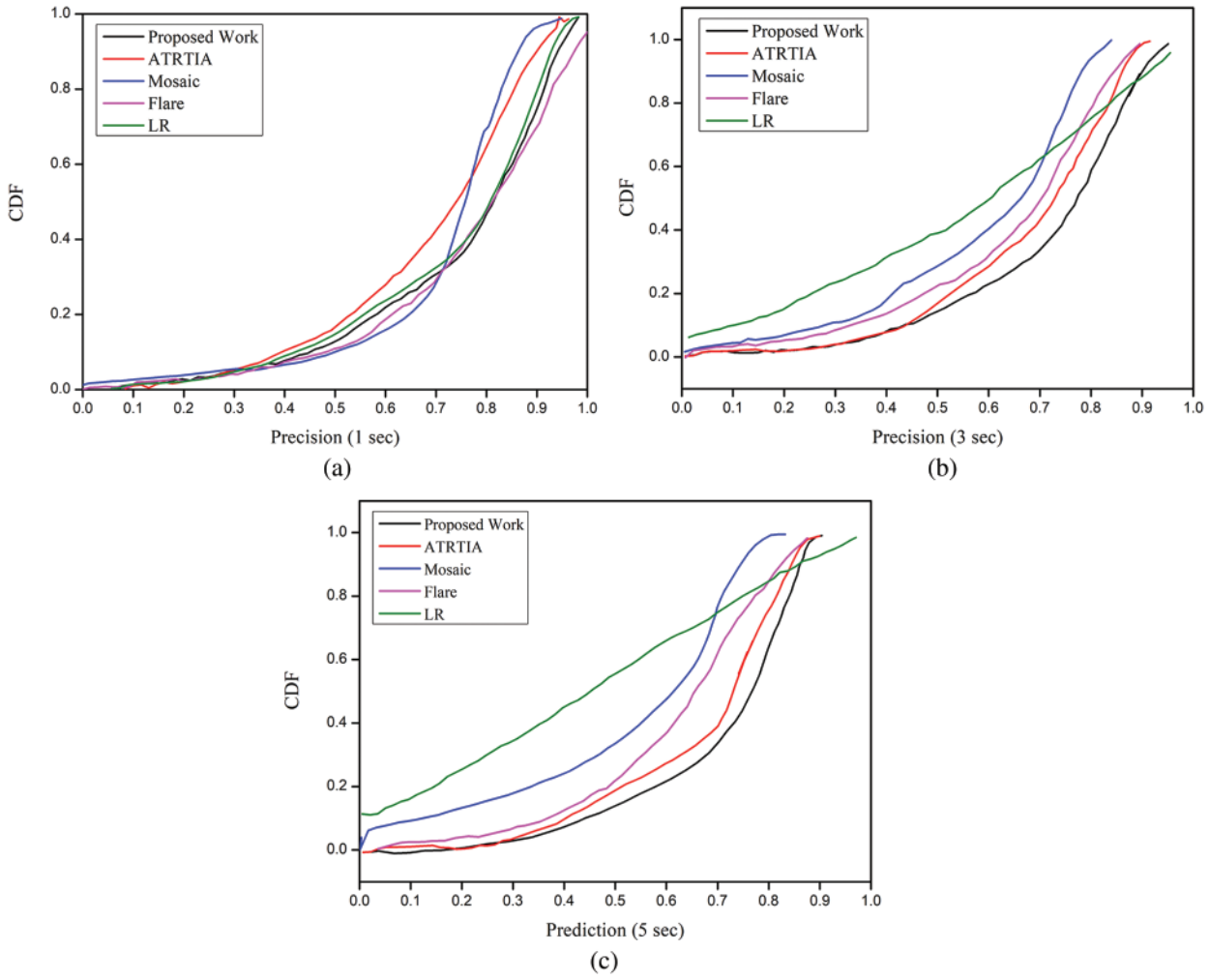

Figure 3: CDF of viewport prediction precision for different prediction horizons where (a) Represents the prediction window precision at 1 sec, (b) Shows the prediction window precision at 3 sec, and (c) Denotes the prediction window precision at 5 sec

4.4 Evaluation Results for Streaming Performance

In this section, we performed experiments on the base of different *QoE* parameters settings against CDF. In multimedia streaming, it has been challenging to quantify the *QoE* metric. We performed experiments with different *QoE* parameters as defined in Eq. (8). Particularly, we set the different values of these parameters such as (1, 1, 1), (1, 2, 1), and (1, 0.3, 0.3) for coefficients. We did experiments for *QoE* evaluation of our proposed Encoder-Decoder based LSTM model by comparing it with other methods such as Flare [26], 360ProbDASH [27] and Tile-VR [28].

Fig. 4 represents the *QoE* distributions of different approaches for three sets of different *QoE* parameters and defines the detailed results under all three QoE metric values against CDFs. On each *QoE* metric over the test traces, our proposed rate adaptation technique based on the Encoder-Decoder LSTM model outperforms the best of existing algorithms. It provides the improved viewport quality in the *QoE* objective, as shown in Fig. 4. Comparing to 360ProbDash, the proposed work improves viewport quality when trying to maximize average quality in *QoE* metrics, such as Fig. 4c. Furthermore, when we focus on viewport spatial and temporal variations in QoE metric, as shown in Figs. 4a and 4b, our proposed work harmonizes priorities among three metrics and has higher *QoE* than its competitors for all parameters in consideration. Flare improves the value of the *QoE* metric in different subsequent segments because of built-in characteristics to mitigate the incurred higher quality changes. It enables the video streaming system perform equally on different *QoE* metrics while the optimum can scarcely be attained. These limitations are overcome by the proposed work without complex tuning to learn the long-term policy by benefiting from prediction. Again, noted that the proposed work outperforms than Tile-VR.

5 Discussion

Virtual Reality (VR) has recently gained tremendous popularity as a result of significant advancements in multimedia technologies. 360-degree video is one of the key elements of VR, where a scene is captured using omnidirectional cameras. It can offer an immersive user viewing experience that makes the user feel like "being there" in the scene. Advanced HMDs have become more popular by enabling a plethora of innovative 360-degree video applications, allowing new media content for the unique immersive video experience to be streamed. Because of this, the community still needs to provide improved solutions. Therefore, it is difficult to transmit the whole 360-degree video to the user due to time-variant characteristics and *QoE* metric. The former can affect the decision-making process for tiles, such as viewport locations. While the latter one includes the different user's *QoE* factors, i.e., user's perceived quality, rebuffering, temporal and spatial quality variance.

A key challenge of 360-degree video streaming is viewport prediction [29,30]. For this purpose, an Encoder-Decoder based LSTM model has been presented, which predicts the future position of a specific user. It captures the non-linear relationship between past and future viewport positions more accurately. This model took the transforming data instead of taking the direct input to predict the future user movements.

Another concern is its incurred video quality variations. We also tried to design a rate adaptation algorithm based on a viewport prediction model that considers the prediction errors to improve the user's *QoE*. It optimizes video segments bitrates temporally and finds bitrate allocation for spatial tiles. Our rate adaptation strategy was inspired by MPC-based rate adaptation that is an efficient control method. The principle of the MPC algorithm is to solve a finite-time optimization problem by applying an optimal solution for each sampling moment.

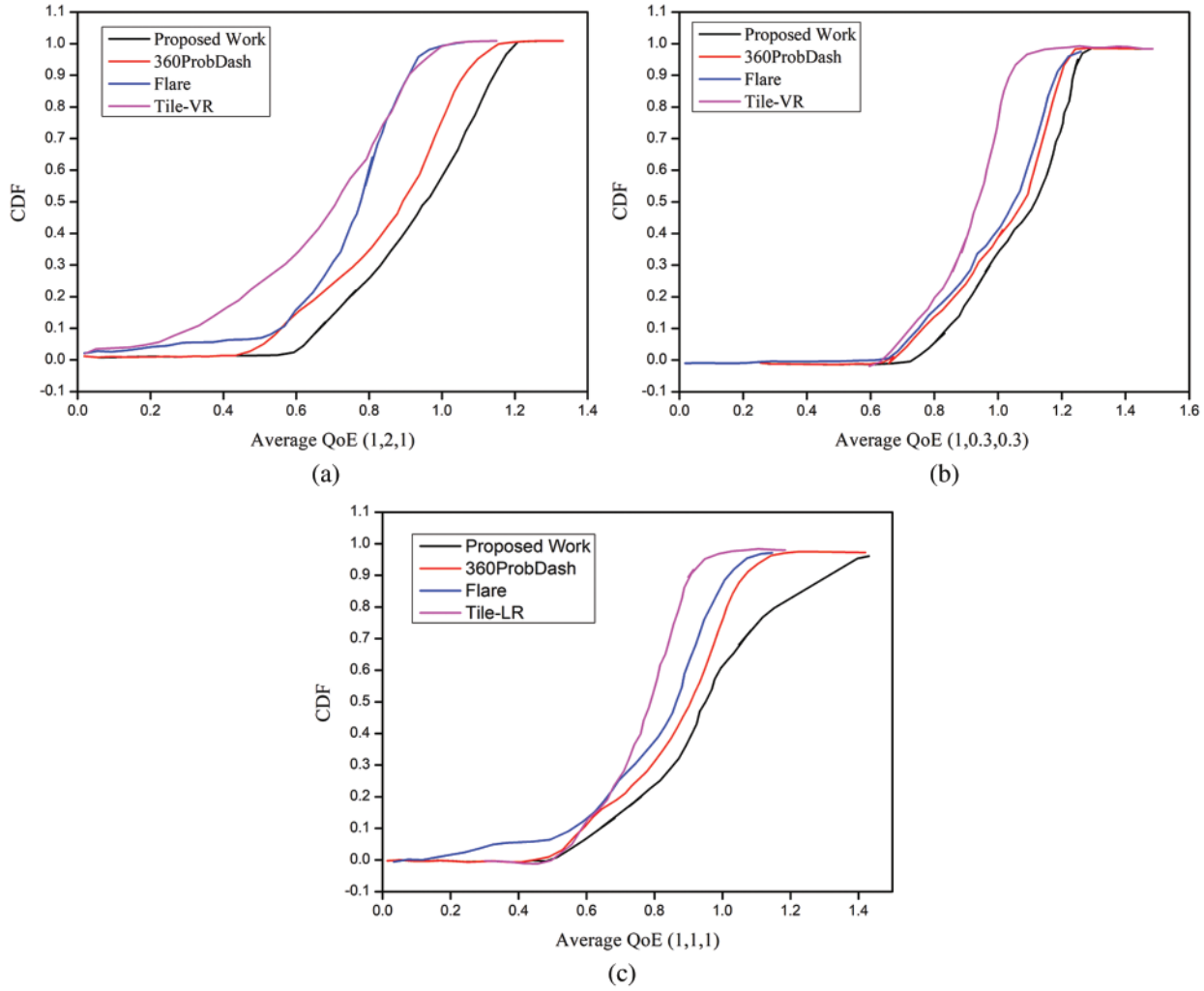

Figure 4: *QoE* with different parameters

We have evaluated the experiments of each part of our proposed system separately, for example, viewport prediction and rate adaptation, maximizing the user's *QoE* based on different parameters. Our experimental results perform well than other comparative schemes.

6 Conclusion

This paper describes a novel Encoder-Decoder based LSTM model for viewport prediction, which takes into account the user head tracking data, saliency and motion maps. The prediction model takes the transforming data instead of taking the direct input from encoder. Moreover, prediction model is then combined with rate adaptation approach that assigns the bitrates to different tiles of video. We tried to optimize the *QoE* metric to achieve the maximum user's quality and evaluated each part of our proposed work separately under different scenarios, by comparing them with other methods. Our evaluations in realistic network settings reveals that the proposed architecture outperforms compared to other existing methods.

In future, we would like to extend our work by performing a subjective user study to enhance the smoothness within the viewport. Also, we intend to incorporate the audio channel by including the different challenges considering a supplemental representation of 360-degree video content.

Acknowledgement: We would like to thank all reviewers for reviewing and giving valuable comments to improve the manuscript's quality.

Funding Statement: There are no funding sources for this project.

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

References

[1] "360-degree camera market analysis–global forecast by 2018–2024." 2020. [Online]. Available: https://www.marketwatch.com/press-release/360-degree-camera-market-size-volume-share-demand-growth-business-opportunity-by-2028-2021-06-21?tesla=y.

[2] J. He, M. A. Qureshi, L. Qiu, J. Li, F. Li *et al.,* "Rubiks: Practical 360-degree streaming for smartphones," in *Proceedings of the 16th Annual Int. Conf. on Mobile Systems, Applications, and Services*, Munich, Germany, pp. 482–494, 2018.

[3] J. V. Hooft, M. T. Vega, S. Petrangeli, T. Wauters and F. D. Turck, "Tile-based adaptive streaming for virtual reality video," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM),* vol. 15, no. 4, pp. 1–24, 2019.

[4] J. Tang, Y. Huo, S. Yang and J. Jiang, "A viewport prediction framework for panoramic videos," in *Int. Joint Conf. on Neural Networks (IJCNN)*, Glasgow, UK, pp. 1–8, 2020.

[5] X. Jiang, Y. H. Chiang, Y. Zhao and Y. Ji, "Plato: Learning-based adaptive streaming of 360-degree videos," in *IEEE 43rd Conf. on Local Computer Networks (LCN)*, Chicago, USA, pp. 393–400, 2018.

[6] L. Xie, Z. Xu, Y. Ban, X. Zhang and Z. Guo, "360probdash: Improving lSTM of 360 video streaming using tile-based http adaptive streaming," in *Proceedings of the 25th ACM Int. Conf. on Multimedia*, New York, USA, pp. 315–323, 2017.

[7] A. T. Nasrabadi, A. Mahzari, J. D. Beshay and R. Prakash, "Adaptive 360-degree video streaming using scalable video coding," in *Proc. of the 25th ACM Int. Conf. on Multimedia*, New York, USA, pp. 1689–1697, 2017.

[8] Y. Zhang, P. Zhao, K. Bian, Y. Liu and L. Song, "DRL360: 360-degree video streaming with deep reinforcement learning," in *IEEE INFOCOM Conf. on Computer Communications*, Paris, France, pp. 1252–1260, 2019.

[9] L. Chopra, S. Chakraborty, A. Mondal and S. Chakraborty, "PARIMA: Viewport adaptive 360-degree video streaming," in *Proceedings of the Web Conf.*, New York, USA, pp. 2379–2391, 2021.

[10] A. D. Aladagli, E. Ekmekcioglu, D. Jarnikov and A. Kondoz, "Predicting head trajectories in 360 virtual reality videos," in *Int. Conf. on 3D Immersion (IC3D)*, Brussels, Belgium, pp. 1–6, 2017.

[11] A. Nguyen, Z. Yan and K. Nahrstedt, "Your attention is unique: Detecting 360-degree video saliency in head-mounted display for head movement prediction," in *Proceedings of the 26th ACM Int. Conf. on Multimedia*, New York, USA, pp. 1190–1198, 2018.

[12] S. Park, A. Bhattacharya, Z. Yang, S. R. Das and D. Samaras, "Mosaic: Advancing user quality of experience in 360-degree video streaming with machine learning," *IEEE Transactions on Network and Service Management,* vol. 18, no. 1, pp. 1000–1015, 2021.

[13] M. Xu, Y. Song, J. Wang, M. Qiao, L. Huo *et al.,* "Predicting head movement in panoramic video: A deep reinforcement learning approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 41, no. 11, pp. 2693–2708, 2018.

[14] M. U. Younus, M. K. Khan, M. R. Anjum, S. Afridi, Z. A. Arain *et al.* "Optimizing the lifetime of software defined wireless sensor network via reinforcement learning," *IEEE Access,* vol. 9, pp. 259–272, 2020.

[15] M. U. Younus, M. K. Khan and A. R. Bhatti, "Improving the software defined wireless sensor networks routing performance using reinforcement learning," *IEEE Internet of Things Journal,* vol. 10*,* pp. 14546–1460, 2021.
[16] N. Kan, J. Zou, C. Li, W. Dai and H. Xiong, "RAPT360: Reinforcement learning-based rate adaptation for 360-degree video streaming with adaptive prediction and tiling," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 31*,* pp. 3631–3677, 2021.
[17] M. U. Younus, "*Contribution to energy optimization in WSN: Routing based on RL and SDN oriented routing*." *PhD diss.*, University of Toulouse 3, Toulouse, France, 2020.
[18] F. Jun, C. Xiaoming, Z. Zhizheng, W. Shilin and C. Zhibo, "360SRL: A sequential reinforcement learning approach for ABR tile-based 360 video streaming," in *IEEE Int. Conf. on Multimedia and Expo (ICME)*, Shanghai, China, pp. 290–295, 2019.
[19] N. Kan, J. Zou, K. Tang, C. Li and H. Xiong, "Deep reinforcement learning-based rate adaptation for adaptive 360-degree video streaming," in *ICASSP IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom, pp. 4030–4034, 2019.
[20] Z. Wu, L. Su, Q. Huang, B. Wu, J. Li *et al.,* "Video saliency prediction with optimized optical flow and gravity center bias," in *IEEE Int. Conf. on Multimedia and Expo (ICME)*, Seattle, USA, pp. 1–6, 2016.
[21] W. C. Lo, C. L. Fan, J. Lee, C. Y. Huang and C. H. Hsu, "360 video viewing dataset in head-mounted virtual reality," in *Proceedings of the ACM MMSys*, Taipei, Taiwan, pp. 211–216, 2017.
[22] Y. Bao, H. Wu, T. Zhang, A. A. Ramli and X. Liu, "Shooting a moving target: Motion-prediction-based transmission for 360-degree videos," in *IEEE Int. Conf. on Big Data (Big Data)*, Washington, USA, pp. 1161–1170, 2016.
[23] Y. Jiang and F. Han, "A hybrid algorithm of adaptive particle swarm optimization based on adaptive moment estimation method," in *Int. Conf. on Intelligent Computing*, Springer, Nanchang, China, pp. 658–667, 2017.
[24] I. Goodfellow, Y. Bengio, A. Courville and Y. Bengio, "*Deep Learning,*" MIT press Cambridge, vol. 1*,* 2016.
[25] S. Park, M. Hoai, A. Hoai and S. R. Das, "Adaptive streaming of 360-degree videos with reinforcement learning," in *Proceedings of the IEEE/CVF Winter Conf. on Applications of Computer Vision, Virtual*, USA, pp. 1839–1848, 2021.
[26] F. Qian, B. Han, Q. Xiao and V. Gopalakrishnan, "Flare: Practical viewport-adaptive 360-degree video streaming for mobile devices," in *Proc. of the 24th Annual Int. Conf. on Mobile Computing and Networking*, New York, USA, pp. 99–114, 2018.
[27] L. Xie, X. Zhimin, B. Yixuan, Z. Xinggong and G. Zongming, "360 probdash: Improving QoE of 360 video streaming using tile-based http adaptive streaming," in *Proceedings of the ACM MM. ACM*, New York, USA, pp. 315–323, 2017.
[28] F. Qian, J. Lusheng, H. Bo and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, ACM, New York, USA, pp. 1–6, 2015.
[29] R. Shafi, W. Shuai and M. U. Younus, "360-degree video streaming: A survey of the state of the art," *Symmetry,* vol. 12*,* no. 9*,* pp. 1–31, 2020.
[30] R. Shafi, W. Shuai and M. U. Younus, "MTC360: A multi-tiles configuration for viewport-dependent 360-gegree video streaming," in *IEEE 6th Int. Conf. on Computer and Communications (ICCC)*, Chendgu, China, pp. 1868–1873, 2020.