Overall Quality Prediction for HTTP Adaptive Streaming Using LSTM Network

Huyen T. T. Tran, *Member, IEEE*, Duc V. Nguyen, *Member, IEEE*, Nam Pham Ngoc, *Member, IEEE*, and Truong Cong Thang, *Senior Member, IEEE*

***Abstract*— HTTP Adaptive Streaming has become a popular solution for multimedia delivery nowadays. However, due to network bandwidth fluctuations, video quality strongly varies during streaming. Therefore, a key challenge in HTTP Adaptive Streaming is how to evaluate the overall quality of a streaming session. In this article, a machine learning approach is proposed for overall quality prediction, where each segment in a streaming session is represented by a set of features. Two options of the feature set are investigated. In the first option, we use four features, namely segment quality, content characteristics, stalling duration, and padding. The second option consists of three features, namely bitstream-level parameters, stalling duration, and padding. The features are fed into a Long Short Term Memory (LSTM) network that is capable of exploring temporal relations between impairment events of quality variations and stalling events. The overall quality is predicted from the outputs of the LSTM network using a linear regression module. Through experimental results, it is shown that the proposed approach achieves a high prediction performance and outperforms seven existing approaches. Especially, the second option is found to be both efficient and effective. The source code of the proposed approach has been made available to the public.**

***Index Terms*— Quality of experience, video adaptive streaming, subjective test, machine learning approach, long short term memory.**

I. Introduction

HTTP Adaptive Streaming (HAS) has become a cost-effective means for multimedia delivery nowadays. In HAS, a video is encoded into multiple versions with different quality levels, each is divided into a sequence of short segments [1]. Segments are hosted on common Web servers. Based on network status, a client decides suitable versions of segments and sends HTTP requests for the versions to the Web server. Due to network bandwidth fluctuations, the selected versions may vary strongly during a streaming session, causing quality variations. Also, stalling events may occur if segments cannot arrive at the client before their playback deadlines as illustrated in Fig. 1. A main challenge in HAS is how to evaluate the overall quality of a streaming session given impacts of quality variations and stalling events [2].

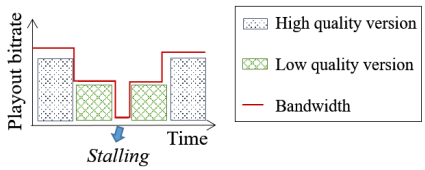


Fig. 1. An illustration of quality variations and stalling events.

Generally, there are two quality measures of a streaming session, namely *instantaneous quality* and *overall quality*. Note that both measures are mentioned in Recommendations ITU-R BT.500-13 [3] and ITU-T P.880 [4].

- *Instantaneous quality* means the momentary quality perceived at any instant during a streaming session.
- *Overall quality* means the perceived quality of a whole session which is rated at the end of the streaming session.

For service providers, it is important to measure both the instantaneous and the overall quality for providing the highest possible quality of experience for users. The instantaneous quality measure is meaningful for understanding immediate reactions of users to impairment events, i.e., quality variations and stalling events [5]–[7]. So far, there have been some studies related to the instantaneous quality assessment and modeling [7]–[9]. Meanwhile, the overall quality measure is important in understanding the quality of service they are providing [5], [10]. In this study, we focus on the overall quality measure.

Most existing approaches for overall quality prediction are analytical model-based approaches [11]–[13]. In these approaches, the impacts of quality variations are modeled using some statistics such as the number of quality switches, the average, the minimum, and the histogram of segment quality values. As for the impacts of stalling events, the number of stalling events, the sum and the histogram of stalling durations are commonly used. In order to predict the overall quality, these statistics are pooled using analytical functions such as linear function [11], [12] and logarithm function [13]. Among the existing approaches, some have jointly taken into account the impacts of quality variations and stalling events [11], [14].

In the literature, there exist only a few advanced machine learning approaches for overall quality prediction such

Manuscript received July 3, 2020; revised September 8, 2020; accepted November 1, 2020. Date of publication November 4, 2020; date of current version August 4, 2021. This article was recommended by Associate Editor X. Wang. *(Corresponding author: Huyen T. T. Tran.)*

Huyen T. T. Tran, Duc V. Nguyen, and Truong Cong Thang are with the Department of Computer and Information Systems, The University of Aizu, Aizuwakamatsu 965-8580, Japan (e-mail: tranhuyen1191@gmail.com; nvduc712@gmail.com; thang@u-aizu.ac.jp).

Nam Pham Ngoc is with the College of Engineering and Computer Science, VinUniversity, Hanoi 100000, Vietnam (e-mail: v.nampn3@vingroup.net).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2020.3035824

as [15], [16]. This is partly because of the lack of sufficient datasets for training. In particular, most publicly available datasets suffer from two key problems. First, the sizes of these datasets are rather small (e.g., 15 sessions in [17]), which is likely to result in over-fitting when training in machine learning approaches. Second, the existing datasets do not reflect realistic streaming scenarios because 1) the included sessions are very short, typically less than 20 seconds (e.g., [18], [19]), and 2) only either quality variations [17] or stalling events [20] are considered.

To the best of our knowledge, the study in [15] presents the first advanced machine learning approach for overall quality prediction, in which a random neural network is employed. The inputs of the approach consist of the average of quantization parameter values over all macro blocks of all video frames, the number of stalling events, the average and the maximum of stalling durations. The approach is evaluated using a dataset which contains 118 streaming sessions with a length of 16 seconds.

In [16], the authors propose another advanced machine learning approach, which uses a regression model to make overall quality prediction. In their proposed approach, each streaming session is characterized by 5 statistics, namely the average of segment quality values, the time over which segment quality decreases took place, the time since the last impairment event (i.e., either a stalling event or a segment quality decrease), the total stalling duration, and the number of stalling events. Three regression models of linear regression, Support Vector Regression (SVR), and ensemble methods are considered. By using a dataset consisting of 112 sessions with a length of approximately 72 seconds, it is found that SVR achieves the highest average prediction performance. Note that the sessions in the dataset are generated from 8 hand-crafted quality variation patterns, each contains only one segment quality decrease and two stalling events at most.

In the latest stage of ITU-T P.1203.3 standardization [21], an advanced machine learning approach is proposed for predicting the overall quality of streaming sessions. The approach uses a random forest model whose inputs include various statistics such as the first, the fifth, and the tenth percentiles of segment quality values, the sum of stalling durations, and the number of stalling events. In [22], the approach is validated using sixty 60-second long sessions and fifteen 240-second long sessions.

In most existing approaches [11]–[16], [21], both analytical model-based and advanced machine learning approaches, their inputs are some statistics of a session as described above. However, such statistics cannot fully reflect the impairment events (i.e., quality variations and stalling events) occurring in a streaming session since the temporal relations between impairment events are lost. For instance, with the same number of stalling events and the same sum of stalling durations, consecutive stalling events may cause more negative impacts than intermittent ones. Therefore, it is necessary to develop an approach that can exploit the temporal relations between impairment events in a streaming session. This is the main motivation of our study.

In this study, we propose a new machine learning approach for overall quality prediction of HAS sessions. In the proposed approach, the inputs are taken on a segment basis instead of statistics on a session basis. In addition, we employ Long Short Term Memory (LSTM) networks for pooling the inputs because it can exploit the temporal relations between impairment events by using a memory [23]. In the literature, LSTM networks are applied for some tasks related to quality prediction such as instantaneous quality prediction [7] and response time prediction of services [24]. However, so far, there exists no study that employs LSTM networks to predict the overall quality of HAS sessions.

Our main contributions in this study are summarized as follows.

- First, we propose a new machine learning approach, which uses an LSTM network to predict the overall quality of HAS sessions. Each segment is attributed by some segment features. To the best of our knowledge, this is the first study using inputs on a segment basis and pooling them using LSTM network for overall quality prediction.
- Second, we investigate two different LSTM network types, namely basic and advanced LSTM networks. Especially, we examine the impact of padding on the performance of the proposed approach. Also, two different options (namely complex and simple) of input features are proposed and analyzed. The complex option is composed of segment quality, content characteristics, stalling duration, and padding. The simple option consists of bitstream-level parameters, stalling duration, and padding.
- Third, the proposed approach is evaluated using three datasets. The first consists of 515 sessions, 332 of which are obtained from a real streaming testbed. The sessions have lengths from 60 to 76 seconds. The second is composed of 120 sessions with the duration of 1 minute. The third dataset contains 588 sessions, each is 8 seconds in length. Experimental results show that the proposed approach achieves high prediction performance. In addition, we conduct a comparison of prediction performance between the proposed approach and seven existing approaches. It is found that the proposed approach outperforms the reference approaches.

A part of this work has been presented in [25]. Compared to the previous work, this study has the following new points. First, we improve the proposed approach by using an advanced LSTM network. Second, a new (simple) option of the feature set is proposed, which is found to be both efficient and effective. Third, we include five additional metrics, which are used to represent the segment quality feature. Fourth, we investigate whether or not the addition of the switching frequency feature to the inputs of the LSTM network is able to improve the performance of the proposed approach. Fifth, an extensive evaluation of LSTM-network-related issues is carried out, including padding options and the number of hidden units. In addition, a comparison is conducted between the LSTM networks and a simple regression model of Support Vector Regression (SVR). Sixth, three more existing approaches,

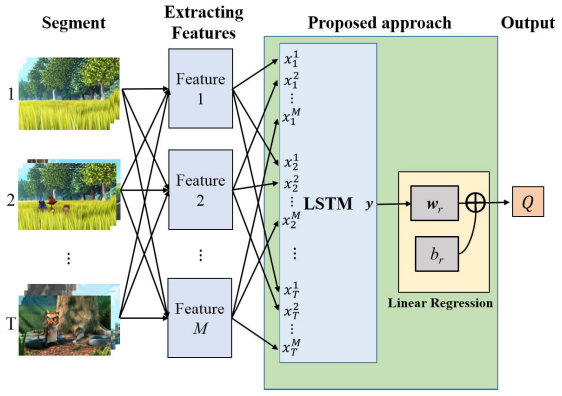


Fig. 2. Architecture of the proposed approach.

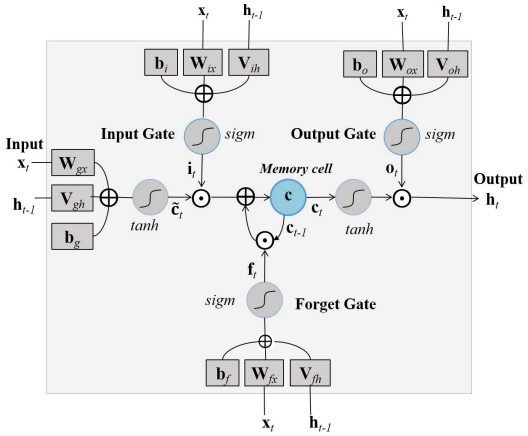


Fig. 3. LSTM unit architecture.

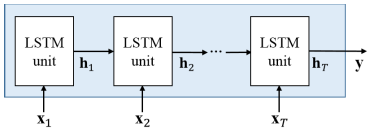


Fig. 4. *baLSTM* network.

which are proposed in [7], [12], [14], are implemented and compared to the proposed approach. Seventh, two datasets are additionally used to evaluate the performances of the approaches.

The rest of this article is organized as follows. Section II presents the architecture of the proposed approach. The options of the feature set are described in Section III. Section IV presents datasets and experiment settings to evaluate the performance of the proposed approach. An analysis of input feature options is presented in Section V. The performances of LSTM network types and their related settings are provided in Section VI. Section VII presents a comparison of performance between the proposed approach and seven existing approaches. In Section VIII, some discussions on the proposed approach are presented. Finally, conclusions are given in Section IX.

II. Architecture of the Proposed Approach

In this section, we first present the general architecture of the proposed approach. Then the two LSTM network types used in this study are described in detail.

A. General Architecture

Fig. 2 shows the architecture of the proposed approach. In particular, a streaming session is considered as a series of segments, each is attributed by a set of features. The features are then fed into an LSTM network. The outputs of the LSTM network are used to predict the overall quality of the streaming session through a linear regression module.

Let bold capital letters (e.g., $\mathbf{X}$), bold lowercase letters (e.g., $\mathbf{x}$), and italic letters (e.g., X) denote matrices, vectors, and scalars, respectively. T denotes the number of segments in the streaming session. Let

$$\mathbf{x}_t = \begin{bmatrix} x_t^1 \\ x_t^2 \\ \vdots \\ x_t^M \end{bmatrix} \tag{1}$$

be the feature vector of segment t $(1 \leq t \leq T)$ with M being the number of features per segment.

Let $\mathbf{y} \in \mathbb{R}^d$ denote the output of the LSTM network. The overall quality Q is calculated by

$$Q = \mathbf{w}_r \mathbf{y} + b_r, \tag{2}$$

where $\mathbf{w}_r$ and b_r are parameters to be learned.

In this study, we use two LSTM network types, called basic (*baLSTM*) and advanced (*adLSTM*) networks. In the two following subsections, the relationships between the inputs and the outputs of these network types are presented in detail.

B. Basic LSTM Network (baLSTM)

In the *baLSTM* network, each vector $\mathbf{x}_t$ is connected to the corresponding hidden state $\mathbf{h}_t$ via an LSTM unit [26] as shown in Fig. 4. Note that the LSTM unit is shared for all the segments. Fig. 3 shows the architecture of the LSTM unit. Specifically, the hidden state $\mathbf{h}_t$ is calculated using the following equations.

$$\tilde{\mathbf{c}}_t = tanh(\mathbf{W}_{gx}\mathbf{x}_t + \mathbf{V}_{gh}\mathbf{h}_{t-1} + \mathbf{b}_g), \tag{3}$$
$$\mathbf{i}_t = sigm(\mathbf{W}_{ix}\mathbf{x}_t + \mathbf{V}_{ih}\mathbf{h}_{t-1} + \mathbf{b}_i), \tag{4}$$
$$\mathbf{f}_t = sigm(\mathbf{W}_{fx}\mathbf{x}_t + \mathbf{V}_{fh}\mathbf{h}_{t-1} + \mathbf{b}_f), \tag{5}$$
$$\mathbf{o}_t = sigm(\mathbf{W}_{ox}\mathbf{x}_t + \mathbf{V}_{oh}\mathbf{h}_{t-1} + \mathbf{b}_o), \tag{6}$$
$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \tag{7}$$
$$\mathbf{h}_t = \mathbf{o}_t \odot tanh(\mathbf{c}_t), \tag{8}$$

where the parameters of $\mathbf{W} \in \mathbb{R}^{d \times M}$, $\mathbf{V} \in \mathbb{R}^{d \times d}$, and $\mathbf{b} \in \mathbb{R}^d$ are learned during the training process, d is the number of hidden units, and $\odot$ denotes the element-wise product. $\mathbf{i}_t, \mathbf{f}_t, \mathbf{o}_t$, and $\mathbf{c}_t$ are respectively the output vectors of the input gate, the forget gate, the output gate, and the memory cell. They are important components to enable the LSTM unit to exploit the temporal relations between impairment events.

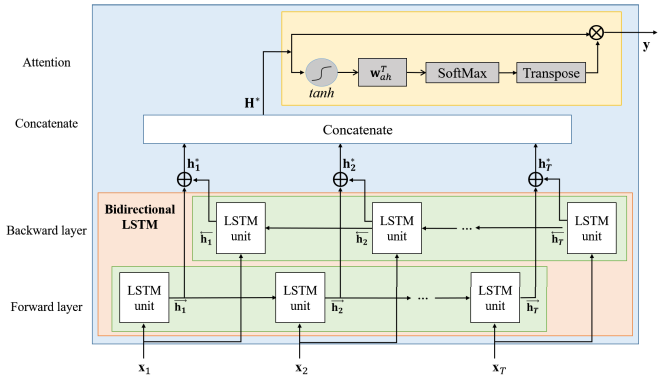


Fig. 5. *adLSTM* network.

In particular, the input gate $\mathbf{i}_t$ chooses whether or not to add new information $\tilde{\mathbf{c}}_t$ from the current inputs to the memory cell $\mathbf{c}_t$. The forget gate $\mathbf{f}_t$ selects and removes old information $\mathbf{c}_{t-1}$ from the memory cell. The output gate $\mathbf{o}_t$ selects useful information from the memory cell $\mathbf{c}_t$ to update the hidden state $\mathbf{h}_t$. The output of the *baLSTM* network is the hidden state $\mathbf{h}_T$ corresponding to the last segment.

$$\mathbf{y} = \mathbf{h}_T. \tag{9}$$

C. Advanced LSTM Network (adLSTM)

It is well known that the overall quality strongly depends on the temporal relations between impairment events [27]. Therefore, the bidirectional [28] and attention mechanisms [29] are applied to enhance the learning capability of the temporal relations. Different from the *baLSTM* network that consists of only the forward layer, the bidirectional LSTM network includes both the forward (i.e., positive time direction) and backward (i.e., negative time direction) layers as shown in Fig. 5. This enables to determine the impact of the current event in relation to both past and future events. Regarding the attention mechanism, it is responsible to decide key events of high attention by computing the weights of hidden states.

In particular, each vector $\mathbf{x}_t$ is first fed into two LSTM units to compute two corresponding hidden states, denoted $\overrightarrow{\mathbf{h}_t}$ for the forward layer and $\overleftarrow{\mathbf{h}_t}$ for the backward layer. The architectures of these LSTM units are similar to that used in the *baLSTM* network. In particular, $\overrightarrow{\mathbf{h}_t}$ and $\overleftarrow{\mathbf{h}_t}$ are calculated by the following equations.

$$\overrightarrow{\tilde{\mathbf{c}}_t} = tanh(\overrightarrow{\mathbf{W}_{gx}}\mathbf{x}_t + \overrightarrow{\mathbf{V}_{gh}}\overrightarrow{\mathbf{h}_{t-1}} + \overrightarrow{\mathbf{b}_g}), \tag{10}$$
$$\overrightarrow{\mathbf{i}_t} = sigm(\overrightarrow{\mathbf{W}_{ix}}\mathbf{x}_t + \overrightarrow{\mathbf{V}_{ih}}\overrightarrow{\mathbf{h}_{t-1}} + \overrightarrow{\mathbf{b}_i}), \tag{11}$$
$$\overrightarrow{\mathbf{f}_t} = sigm(\overrightarrow{\mathbf{W}_{fx}}\mathbf{x}_t + \overrightarrow{\mathbf{V}_{fh}}\overrightarrow{\mathbf{h}_{t-1}} + \overrightarrow{\mathbf{b}_f}), \tag{12}$$
$$\overrightarrow{\mathbf{o}_t} = sigm(\overrightarrow{\mathbf{W}_{ox}}\mathbf{x}_t + \overrightarrow{\mathbf{V}_{oh}}\overrightarrow{\mathbf{h}_{t-1}} + \overrightarrow{\mathbf{b}_o}), \tag{13}$$
$$\overrightarrow{\mathbf{c}_t} = \overrightarrow{\mathbf{f}_t} \odot \overrightarrow{\mathbf{c}_{t-1}} + \overrightarrow{\mathbf{i}_t} \odot \overrightarrow{\tilde{\mathbf{c}}_t}, \tag{14}$$
$$\overrightarrow{\mathbf{h}_t} = \overrightarrow{\mathbf{o}_t} \odot tanh(\overrightarrow{\mathbf{c}_t}), \tag{15}$$
$$\overleftarrow{\tilde{\mathbf{c}}_t} = tanh(\overleftarrow{\mathbf{W}_{gx}}\mathbf{x}_t + \overleftarrow{\mathbf{V}_{gh}}\overleftarrow{\mathbf{h}_{t-1}} + \overleftarrow{\mathbf{b}_g}), \tag{16}$$
$$\overleftarrow{\mathbf{i}_t} = sigm(\overleftarrow{\mathbf{W}_{ix}}\mathbf{x}_t + \overleftarrow{\mathbf{V}_{ih}}\overleftarrow{\mathbf{h}_{t-1}} + \overleftarrow{\mathbf{b}_i}), \tag{17}$$
$$\overleftarrow{\mathbf{f}_t} = sigm(\overleftarrow{\mathbf{W}_{fx}}\mathbf{x}_t + \overleftarrow{\mathbf{V}_{fh}}\overleftarrow{\mathbf{h}_{t-1}} + \overleftarrow{\mathbf{b}_f}), \tag{18}$$
$$\overleftarrow{\mathbf{o}_t} = sigm(\overleftarrow{\mathbf{W}_{ox}}\mathbf{x}_t + \overleftarrow{\mathbf{V}_{oh}}\overleftarrow{\mathbf{h}_{t-1}} + \overleftarrow{\mathbf{b}_o}), \tag{19}$$
$$\overleftarrow{\mathbf{c}_t} = \overleftarrow{\mathbf{f}_t} \odot \overleftarrow{\mathbf{c}_{t-1}} + \overleftarrow{\mathbf{i}_t} \odot \overleftarrow{\tilde{\mathbf{c}}_t}, \tag{20}$$
$$\overleftarrow{\mathbf{h}_t} = \overleftarrow{\mathbf{o}_t} \odot tanh(\overleftarrow{\mathbf{c}_t}), \tag{21}$$

where the parameters of $\overrightarrow{\mathbf{W}} \in \mathbb{R}^{d\times M}$, $\overrightarrow{\mathbf{V}} \in \mathbb{R}^{d\times d}$, $\overrightarrow{\mathbf{b}} \in \mathbb{R}^d$, $\overleftarrow{\mathbf{W}} \in \mathbb{R}^{d\times M}$, $\overleftarrow{\mathbf{V}} \in \mathbb{R}^{d\times d}$, and $\overleftarrow{\mathbf{b}} \in \mathbb{R}^d$ are learned in the training process.

Then, the outputs of the bidirectional LSTM network $\overrightarrow{\mathbf{h}_t}$ and $\overleftarrow{\mathbf{h}_t}$ are aggregated using element-wise addition as follows.

$$\mathbf{h}_t^* = \overrightarrow{\mathbf{h}_t} \oplus \overleftarrow{\mathbf{h}_t}. \tag{22}$$

Let $\mathbf{H}^* = [\mathbf{h}_1^*, \mathbf{h}_2^*, \ldots, \mathbf{h}_T^*] \in \mathbb{R}^{d\times T}$ be a matrix consisting of the output vectors $\mathbf{h}_t^*$. The weights of the hidden states are calculated by the following equation.

$$\alpha = softmax(\mathbf{w}_{ah}^T tanh(\mathbf{H}^*)), \tag{23}$$

where $\mathbf{w}_{ah} \in \mathbb{R}^d$ is a parameter to be learned.

Finally, the output of the *adLSTM* network is given by

$$\mathbf{y} = \mathbf{H}^* \alpha^T. \tag{24}$$

III. Input Features

In this section, we will propose two input choices, called complex option and simple option, of the feature set. In each option, the features could be divided into three groups, namely stalling duration, padding, and quality variations. The features of stalling duration and padding are the same for both the options. The aim of using the stalling duration feature is to represent the impacts of stalling events occurring in a streaming session. For the padding feature, it is to ensure that all sessions have the same length in the training process of the LSTM networks. Besides, to reflect the impact of quality variations, two different feature sets are considered in the two options. Particularly, the complex option (denoted *I1*) includes two features of segment quality and content characteristics, which may be very costly to obtain. For the simple option (denoted *I2*), the features are based on bitstream-level parameters, which can be easily extracted from a bitstream. In the following, the features will be described in detail.

A. Complex Input Option (I1)

As mentioned, each segment is attributed by a number of features. In the first option, there are four features, namely segment quality, content characteristics, stalling duration, and padding.

1) Segment Quality: The segment quality feature can be represented by various quality metrics. In this study, we consider eight metrics, namely *S-MOS* [12], Bitrate (*BR*), *PSNR*, *PSNR-HVS* [30], *PSNR-HVS-M* [31], *SSIM* [32], *MS-SSIM* [33], and *VIF* [34]. The description of these metrics is presented in Table I. Among these metrics, only the six metrics of *S-MOS*, *PSNR-HVS*, *PSNR-HVS-M*, *SSIM*, *MS-SSIM*, and *VIF* do take into account human visual system properties. It should be noted that some quality metrics (e.g., *PSNR* and its variants) are easy to measure, whereas the *S-MOS* metric is very time-consuming to obtain.

TABLE I
METRICS USED TO REPRESENT THE SEGMENT QUALITY FEATURE

Notation	Description
S-MOS	Mean opinion score of a segment [12]
BR	Bitrate of a segment
PSNR	Peak Signal-to-Noise Ratio of a segment
PSNR-HVS	PSNR-variant modified to take into account the Human Visual System (HVS) properties [30]
PSNR-HVS-M	PSNR-HVS-variant modified to take into account between-coefficient masking of Discrete Cosine Transform (DCT) basis functions [31]
SSIM	Structural Similarity of a segment [32]
MS-SSIM	Multi-Scale Structural Similarity of a segment [33]
VIF	Visual Information Fidelity of a segment [34]

2) Content Characteristics: It is well known that video quality may be affected by content characteristics [35]. Similar to [35], two dimensions of content characteristics, namely spatial complexity and temporal complexity, are taken into account in the proposed approach.

To represent the spatial complexity of each segment, we use the Spatial Variance (*SV*) metric [35], [36]. This metric is calculated from the MPEG-7 edge histogram algorithm. Specifically, each frame in a segment is firstly divided into 4×4 sub-blocks, and then a histogram of five edge types (vertical, horizontal, 45°, 135°, and non-direction) is calculated for each sub-block [36]. Let S_{lm} denote the average edge histogram value for all sub-blocks in the l^{th} frame with edge type $m \in \{0, 1, 2, 3, 4\}$. The SV value of a segment is derived by

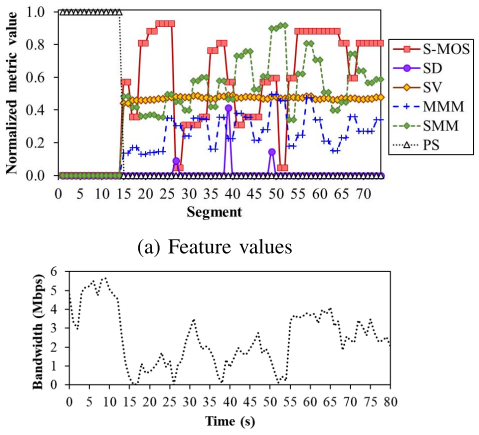
$$SV = \frac{1}{N_l \times N_m} \sum_{l=0}^{N_l-1} \sum_{m=0}^{N_m-1} S_{lm}, \tag{25}$$

where N_l and N_m are respectively the total number of frames in the segment and the total number of edge types.

To represent the temporal complexity of each segment, two metrics calculated from motion vectors are used. The first metric, denoted *MMM*, is the mean of motion vector magnitudes in a segment. The second metric, denoted *SMM*, is the standard deviation of motion vector magnitudes. Note that, since the three metrics of *SV*, *MMM*, and *SMM* are independent, they are all fed into the LSTM network.

3) Stalling Duration: The stalling duration feature of a segment is represented by the amount of time (denoted *SD*) that the user has to wait since the end of the previous segment until the start of the current segment. If the current segment arrives at the client before the playback of the previous segment finishes (called the playback deadline), *SD* is set to 0. Otherwise, a stalling event occurs and *SD* is a positive value. Note that the number of stalling events is the number of segments with *SD* larger than 0.

4) Padding: In practice, streaming sessions usually have different lengths (i.e., the number of segments). Hence, we employ zero-padding method in the training process to ensure that the sessions have the same length. In particular, some segments, called *padded segments*, are appended to the beginning of every session so that its length is the same as the length of the longest session. To differentiate the padded and actual segments, we define a boolean variable *PS* as follows.



(a) Feature values

(b) Bandwidth trace

Fig. 6. An example of a streaming session.

$$PS(t) = \begin{cases} 1, & \text{if segment } t \text{ is a padded segment} \\ 0, & \text{otherwise} \end{cases} \tag{26}$$

Note that, for all padded segments, their other features such as segment quality, content characteristics, and stalling duration take a value of 0. To investigate the impact of padding on the performance of the proposed approach, an evaluation of four different padding cases will be presented in Subsection VI-B.

Fig. 6 illustrates the normalized feature values of a streaming session and the corresponding bandwidth trace. We can see that the first fourteen segments are padded segments (i.e., $PS = 1$). The remaining ones are actual segments (i.e., $PS = 0$). Also, because of bandwidth fluctuations (as shown in Fig. 6b), the segment quality (i.e., the *S-MOS* values in Fig. 6a) varies during the streaming session. In general, segment quality is improved as the bandwidth increases, and vice versa. In this session, there are in total three stalling events occurring at the 27^{th}, 39^{th}, and 49^{th} segments where $SD > 0$. As the *SV* values are stable, this session does not have significant changes in the spatial complexity. Meanwhile, the temporal complexity (i.e., the *MMM* and *SMM* values) varies drastically.

B. Simple Input Option (I2)

In practice, the features of segment quality and content characteristics are generally very costly since it is time-consuming to obtain these features and resource-consuming to store. Therefore, we additionally propose another option, denoted *I2*, which is simpler and can be directly employed in practice. Particularly, the features in this option include:

- Bitstream-level parameters
- Stalling duration
- Padding

Here, the features of stalling duration and padding are the same as the complex option. The bitstream-level parameters include Quantization Parameter (*QP*), bitrate, resolution, and frame-rate. *QP* is calculated as the average of quantization parameter values over all macro-blocks of all frames in a segment. The other parameters are simply the bitrate, resolution,

TABLE II

SETTINGS OF HAND-CRAFTED SESSIONS FOR THE NEWLY CREATED DATASET

(a) Settings of versions

Version	**v1**	**v2**	**v3**	**v4**	**v5**
QP	26				
Resolution	1280×720	854×480	640×360	426×240	256×144

(b) Versions of segments

Version		**Segments**					
		1–10	***11–20***	***21–30***	***31–40***	***41–50***	***51–60***
Session	***#1***	v1	v1	v1	v1	v1	v1
	#2	v2	v2	v2	v2	v2	v2
	#3	v3	v3	v3	v3	v3	v3
	#4	v4	v4	v4	v4	v4	v4
	#5	v5	v5	v5	v5	v5	v5
	#6	v1	v2	v1	v2	v1	v2
	#7	v1	v3	v1	v3	v1	v3
	#8	v1	v4	v1	v4	v1	v4
	#9	v1	v5	v1	v5	v1	v5
	#10	v2	v3	v2	v3	v2	v3
	#11	v2	v4	v2	v4	v2	v4
	#12	v2	v5	v2	v5	v2	v5
	#13	v3	v4	v3	v4	v3	v4
	#14	v3	v5	v3	v5	v3	v5
	#15	v4	v5	v4	v5	v4	v5
	#16–#42	v1	v1	v1	v1	v1	v1

(c) Stalling durations

Session	**Durations of Stalling Events (T_k)**					
	T_1	T_2	T_3	T_4	T_5	T_6
#1–#15	—					
#16	0.25					
#17	0.5					
#18	1					
#19	2					
#20	3					
#21	4					
#22	0.25	0.5				
#23	0.25	2				
#24	0.25	4				
#25	0.5	1				
#26	0.5	3				
#27	1	2				
#28	1	4				
#29	2	3				
#30	3	4				
#31	0.25	0.5	1			
#32	0.25	1	3			
#33	0.5	1	2			
#34	0.5	2	4			
#35	1	2	3			
#36	2	3	4			
#37	0.25	0.5	1	2		
#38	0.5	1	2	3		
#39	1	2	3	4		
#40	0.25	0.5	1	2	3	
#41	0.5	1	2	3	4	
#42	0.25	0.5	1	2	3	4

and frame-rate of a segment. Obviously, these parameters are very basic and can be easily extracted from a video bitstream. Note that, among these parameters, bitrate is actually a choice of quality metrics in the complex option.

In Section V, an evaluation of the performance is conducted for both the input options, where the focus is on the choice of the best segment quality metric and the benefits of different features.

IV. DATASETS AND EXPERIMENT SETTINGS

In this section, we first describe the generation of datasets. Then we present some experiment settings such as the training parameters of the proposed approach, cases of input features, and performance evaluation metrics used in this study.

A. Dataset

To address the problem of lack of training data, the dataset used in this study is combined from three datasets. The first and second datasets are from our previous work of [11] and [37]. The remaining dataset is newly created by conducting a subjective test using the same procedure as in [11], [37].

TABLE III

SETTINGS OF VERSIONS USED TO GENERATE REAL STREAMING SESSIONS FOR THE NEWLY CREATED DATASET

Version	**QP**	**Resolution**	**Average bitrate**	
			Video #1	*Video #2*
1	26	256×144	223	377
2	24	256×144	268	477
3	26	426×240	484	903
4	24	426×240	577	1146
5	26	640×360	841	1589
6	24	640×360	1003	2018
7	26	854×480	1213	2308
8	24	854×480	1453	2926
9	26	1280×720	1948	3692
10	24	1280×720	2378	4683

In the third dataset, there are totally 144 sessions, which are generated from two 1-minute long videos (denoted Video #1 and Video #2). Note that these videos are different from those used in the first and second datasets. To generate the sessions, the individual videos are first divided into 1-second long segments and encoded using H.264/AVC (libx264) with a frame-rate of 24fps. Then, for each video, 72 sessions consisting of 42 hand-crafted sessions and 30 real streaming sessions are generated.

Table II shows the settings of the hand-crafted sessions. In particular, they consist of 5 sessions having no quality variation and no stalling event (i.e., #1−#5), 10 sessions having periodic quality variations with the period of 10 segments and no stalling event (i.e., #6−#15), and 27 sessions containing from 1 to 6 stalling events with the durations of 0.25s, 0.5s, 1s, 2s, 3s, and 4s and no quality variation (i.e., #16−#42). Note that, to create these sessions, each segment is encoded into 5 versions with QPs and resolutions shown in Table IIa. In addition, stalling events are regularly introduced into sessions #16−#42.

With respect to the real streaming sessions, each segment is encoded into 10 versions corresponding to different QP values and resolutions as shown in Table III. To decide the versions of segments, we use two adaptation methods of [38], [39], which are run in a streaming testbed using bandwidth traces from a mobile network. The real streaming sessions consist of both quality variations and stalling events.

Similar to prior studies [11], [37], the test conditions are designed following Recommendation ITU-T P.913 [40]. In order to minimize subjects' fatigue, the subjective test is divided into four parts that are conducted in different days. The duration of each part is approximately 50 minutes. Every 20 minutes there is a break of 10 minutes. Each subject takes part in at most two test parts. Before doing actual subjective tests, subjects are trained to get accustomed to the rating procedure and the range of video quality scores. The sessions are randomly displayed on a 14-inch screen and a black background. At the end of each session, each subject gives a rating score with the score range from 1 (worst) to 5 (best).

Totally, 53 subjects between the ages 18 and 41 take part in the subjective test. The total time of the subjective test is

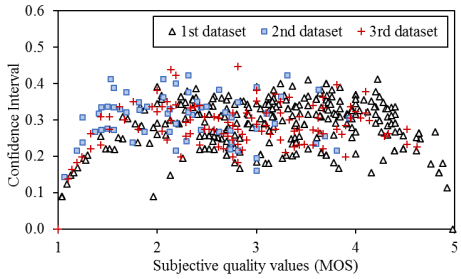


Fig. 7. 95% confidence intervals of three used datasets.

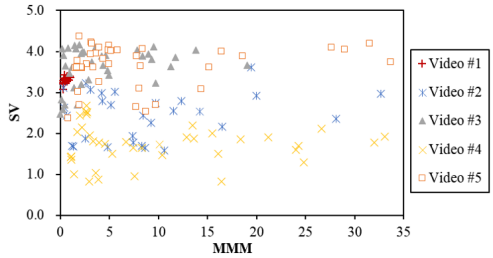


Fig. 8. Content characteristics per segment of videos used in the datasets.

approximately 78 hours. A screening analysis of the test results is performed following Recommendation ITU-T P.913 [40], and two subjects are rejected. After eliminating the scores of the rejected subjects, each session is rated by 21 valid subjects. The subjective overall quality value of each session is calculated as the average score of the valid subjects.

Fig. 7 shows the 95% confidence intervals of the subjective overall quality values for the three datasets. We can see that the confidence intervals of the three datasets are in the same range from 0 to 0.45. In addition, the confidence intervals are generally smaller at the two ends of the score range. This is because subjects are more confident in rating 1) sessions of very high (or very low) quality scores and 2) sessions with small quality variations.

The combined dataset consists of totally 515 sessions with 183 hand-crafted sessions and 332 real streaming sessions generated from 5 different videos (i.e., Video #1, Video #2, Video #3, Video #4, and Video #5). The lengths of the sessions are from 60 to 76 seconds. Fig. 8 shows the content characteristics of spatial complexity (*SV*) and temporal complexity (*MMM*) per segment of the used videos. It can be seen that the content characteristics of the used videos diverge considerably. In particular, Video #1 has medium spatial complexity and very low temporal complexity, and both characteristics are almost constant. Meanwhile, the spatial and the temporal complexity of the other videos is dramatically variable. Specifically, the variation of the spatial complexity (i.e., *SV* values) is generally in a high range for Videos #3 and #5, a medium range for Video #2, and a low range for Video #4. For the temporal complexity, the *MMM* values of all these four videos range greatly from low to high levels.

B. Training Parameters and Cases of Input Features

To evaluate the prediction performance of the proposed approach, the dataset is randomly divided into a training set of 412 sessions and a test set of the 103 remaining sessions. The division is repeated 100 times, resulting in 100 pairs of training and test sets. The results presented in the following sections are the average values over the 100 pairs of training and test sets.

In the training process, the loss function is calculated as the root mean square error between the predicted quality values and the corresponding subjective quality values. This function is minimized using batch gradient descent method based on Adam optimization algorithm [41]. The parameters of the Adam algorithm are set as follows: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-08$. The learning rate is set to 0.01. We test different numbers of epochs e from 500 to 7000 with a step size of 500. Regarding the setting of LSTM network, the number of hidden units d in the LSTM unit is set to three values of $d \in \{1, 3, 5\}$.

To investigate the roles of the features in the proposed approach, we consider four cases of input features in the first option *I1*. In the first case (denoted *Full*), each segment is represented by all the four features described in Subsection III-A. For the three remaining cases, only three of the four features are used. In particular, the content characteristic feature is excluded from the inputs in the second case (denoted *w/oCC*). In the third case (denoted *w/oSD*), the stalling duration feature is not considered. For the last case (denoted *w/oSQ*), the segment quality feature is not used as the input of the proposed approach.

In previous studies, switching frequency, i.e., the number of segment quality switches, is investigated and employed to predict the overall quality of streaming sessions [21]. In our investigation, an additional option of the feature set, denoted *exI1*, which is an extended option of *I1*, is investigated. Beside the four features of the option *I1*, the option *exI1* additionally takes into account the switching frequency feature. The switching frequency feature of a segment is represented by the cumulative number of segment quality switches since the beginning of the session until the current segment.

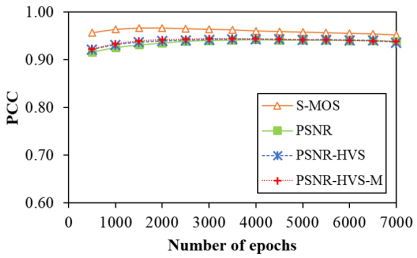
C. Evaluation Metrics

Regarding performance evaluation metrics, we use Pearson Correlation Coefficient (PCC), Root Mean Square Error (RMSE), and the Spearman rank-order correlation coefficient (SROCC), which are averaged over the 100 test sets. The PCC, RMSE, and SROCC are respectively used to measure the linear relationship, difference, and rank correlation between the overall quality values predicted from an approach and the subjective overall quality values obtained by subjective tests. Note that a higher PCC, a lower RMSE, and a higher SROCC mean better prediction performance. These behaviors are indicated by notations PCC (↑), RMSE (↓), and SROCC (↑) in Tables IV,V,VI,VII.

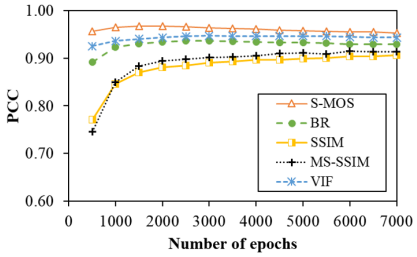
V. Analysis of the Input Feature Options

A. Performance of the Complex Option I1

In this subsection, we will investigate the impacts of the features in the complex option *I1*, especially the benefits of content characteristics and the best quality metric for segments. For this purpose, we evaluate the four cases of input



(a) *S-MOS*, *PSNR*, and *PSNR*-variants



(b) *S-MOS*, *BR*, *SSIM*, *MS-SSIM*, and *VIF*

Fig. 9. PCC values averaged over the 100 test sets for the *Full* case using the different quality metrics of the option *I1* and the *adLSTM* network.

features as presented in subsection IV-B. Here, the *adLSTM* network is used, the number of epochs e is from 500 to 7000, and the number of hidden units d is set to 5.

Fig. 9 shows the PCC values averaged over the 100 test sets for the *Full* case using different quality metrics. It can be seen that *S-MOS* consistently results in the highest PCC values. This reveals that *S-MOS* is the best metric to represent the segment quality feature.

Interestingly, although *PSNR* is a simple metric, its PCC values are just slightly lower than those of *S-MOS*. In particular, the highest PCC value is 0.942 for *PSNR* (when the number of epochs is 4000) and 0.966 for *S-MOS* (when the number of epochs e is 1500). Besides, the PCC values corresponding to *PSNR*-variants and *VIF* are also a little lower compared to *S-MOS*. Hence, when content characteristics are included, the metrics of *PSNR*, *PSNR*-variants, and *VIF* can also be used instead of *S-MOS* to represent the segment quality feature.

Fig. 11 compares the best performance of the *Full* and *w/oCC* cases for the test sets when using different segment quality metrics. We can see that the *Full* case achieves significantly higher performance compared to the *w/oCC* case for all the metrics except for *S-MOS*. The largest performance difference is found for *PSNR* and *VIF*. Meanwhile, the *Full* and *w/oCC* cases have similar performance when using *S-MOS*. This result implies that, when using *S-MOS* to represent the segment quality feature, the additional use of the content characteristic feature does not bring significant improvements. Meanwhile, for the metrics of *BR*, *PSNR*, *PSNR*-variants, *SSIM*, *MS-SSIM*, and *VIF*, it is beneficial to include the content characteristic feature.

Among the three metrics of *PSNR*, *PSNR-HVS*, and *PSNR-HVS-M*, the gain of the *Full* case compared to the *w/oCC* case is largest for *PSNR* and smallest for *PSNR-HVS-M*. This indicates that the *PSNR*-variants, which additionally take into account the human visual system properties, are less dependent on content characteristics than the original *PSNR*. A similar conclusion can also be found for *SSIM* and its variant *MS-SSIM*.

TABLE IV

BEST PERFORMANCE OF THE PROPOSED APPROACH FOR THE DIFFERENCE INPUT CASES OF *Full*, *w/oSD*, *w/oSQ*, AND *exI1* USING *S-MOS* AND THE *adLSTM* NETWORK FOR THE TEST SETS

Input option	PCC (↑)	RMSE (↓)	SROCC (↑)
Full	0.966	0.248	0.965
w/oSD	0.838	0.518	0.820
w/oSQ	0.638	0.734	0.584
exI1	0.963	0.257	0.963

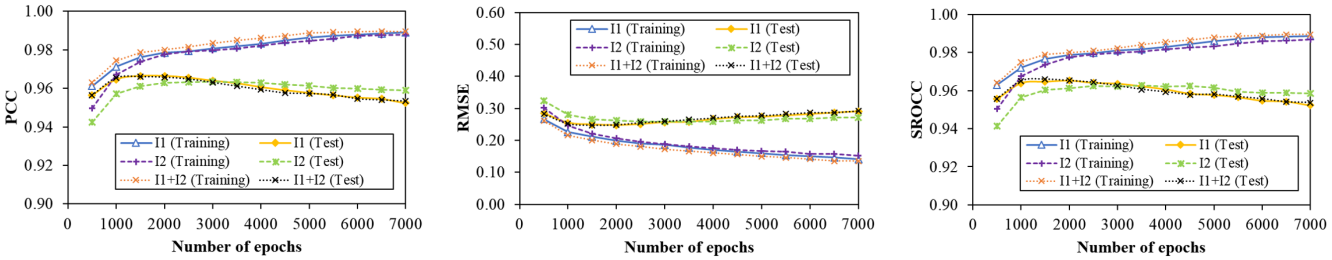
Table IV shows the best performance of the proposed approach for the different input cases of *Full*, *w/oSD*, *w/oSQ*, and *exI1*. It can be seen that the performance of the proposed approach is significantly reduced when either the stalling duration feature or the segment quality feature is excluded from the inputs. In particular, when the stalling duration feature is not considered (i.e., the *w/oSD* case), the PCC and SROCC values drop respectively from 0.966 to 0.838 and from 0.965 to 0.820 while the RMSE value increases from 0.248 to 0.518. This indicates that stalling events have significant impacts on the overall quality of a session. Compared to the *w/oSD* case, the *w/oSQ* case results in a much lower PCC value (i.e., 0.638 vs. 0.838), a much higher RMSE value (i.e., 0.734 vs. 0.518), and a significantly lower SROCC value (i.e., 0.584 vs. 0.820). This means that the segment quality feature plays a more important role than the stalling duration feature in the proposed approach.

In addition, we investigate whether or not the addition of the switching frequency feature is able to improve the performance of the proposed approach. From Table IV, it can be seen that in fact the switching frequency feature cannot bring significant improvements. Even, this results in a slightly decrease of the performance. In particular, the PCC, RMSE, and SROCC values are respectively 0.966, 0.248, and 0.965 for the option *Full*, and 0.963, 0.257, and 0.963 for the option *exI1*. This can be explained by the fact that the switching frequency feature cannot differentiate impacts of segment quality switches with different switching degrees [37]. This is inline with [42] where it is found that the impact of the switching frequency is negligible on the overall quality.

It should be noted that, because *S-MOS* provides the best performance, it will be used for the complex option *I1* in the rest of this article.

B. Comparison of Input Options

In this subsection, we present a performance comparison of the simple option (*I2*), the complex option (*I1*), and the combination option of *I1* and *I2* (denoted *I1*+*I2*). The obtained result of each option using the *adLSTM* network is shown in Fig. 10. Note that the *Full* case and *S-MOS* quality metric are used in the complex and combination options. From Fig. 10, it can be seen that the performance of the complex option and the combination option is the same for the test



(a) PCC (b) RMSE (c) SROCC

Fig. 10. Prediction performance of the proposed approach using the simple, the complex, and the combination options.

TABLE V

BEST PERFORMANCE OF THE PROPOSED APPROACH USING THE *adLSTM*, *baLSTM* NETWORKS, AND SVR MODEL

(a) *adLSTM* and *baLSTM* networks

Option		*adLSTM* network						*baLSTM* network						Gain of *adLSTM* network		
		Training set			*Test set*			*Training set*			*Test set*			*Test set*		
		PCC (↑)	*RMSE* (↓)	*SROCC* (↑)	*PCC* (↑)	*RMSE* (↓)	*SROCC* (↑)	*PCC* (↑)	*RMSE* (↓)	*SROCC* (↑)	*PCC* (↑)	*RMSE* (↓)	*SROCC* (↑)	ΔPCC	$\Delta RMSE$	Δ *SROCC*
I1	*Full*	0.976	0.211	0.977	0.966	0.250	0.965	0.978	0.205	0.978	0.967	0.248	0.965	-0.001	-0.002	0.000
	w/oCC	0.973	0.218	0.974	0.966	0.249	0.964	0.974	0.216	0.974	0.967	0.247	0.965	-0.001	-0.002	-0.001
	w/oSD	0.891	0.424	0.876	0.838	0.519	0.819	0.871	0.461	0.858	0.827	0.534	0.810	0.011	0.015	0.009
	w/oSQ	0.722	0.646	0.685	0.638	0.734	0.584	0.685	0.678	0.637	0.609	0.751	0.551	0.029	0.017	0.033
I2		0.981	0.182	0.981	0.963	0.259	0.963	0.979	0.200	0.978	0.964	0.257	0.963	-0.001	-0.002	0.000
I1+I2		0.979	0.201	0.979	0.966	0.248	0.966	0.979	0.202	0.979	0.967	0.250	0.965	-0.001	0.002	0.001

(b) SVR model

Option		*SVR model*						Gain of *adLSTM* network			Gain of *baLSTM* network		
		Training set			*Test set*			*Test set*			*Test set*		
		PCC (↑)	*RMSE* (↓)	*SROCC* (↑)	*PCC* (↑)	*RMSE* (↓)	*SROCC* (↑)	ΔPCC	$\Delta RMSE$	Δ *SROCC*	ΔPCC	$\Delta RMSE$	Δ *SROCC*
I1	*Full*	0.816	0.547	0.813	0.805	0.564	0.813	0.161	0.314	0.152	0.173	0.359	0.165
	w/oCC	0.825	0.535	0.833	0.818	0.549	0.833	0.148	0.300	0.131	0.156	0.333	0.141
	w/oSD	0.812	0.554	0.794	0.799	0.573	0.794	0.039	0.054	0.025	0.072	0.112	0.064
	w/oSQ	0.448	0.839	0.464	0.437	0.851	0.464	0.201	0.117	0.120	0.248	0.173	0.173
I2		0.806	0.573	0.799	0.792	0.592	0.799	0.171	0.333	0.164	0.187	0.392	0.179
I1+I2		0.819	0.548	0.813	0.806	0.566	0.813	0.160	0.318	0.153	0.173	0.364	0.166

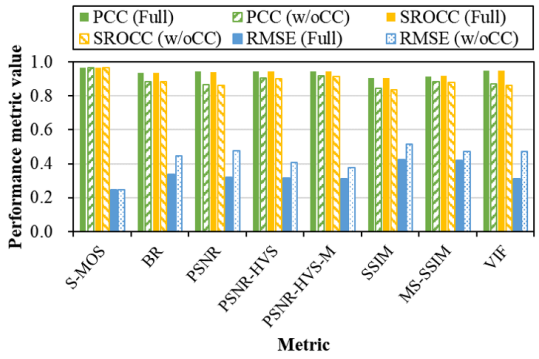


Fig. 11. Best performance of the proposed approach using the *adLSTM* network for the *Full* and *w/oCC* cases of the option *I1* over the test sets.

sets. Therefore, the additional use of the features in the simple option does not bring considerable improvements. In other words, the combination of the two options *I1* and *I2* is redundant.

In addition, it is interesting to see that the performance of the simple option *I2* is very high. In particular, the best performance, with PCC=0.963, RMSE=0.259, and SROCC=0.963, is reached when the number of epochs is 3500. When the number of epochs increases beyond 3500, the training performance is also (slightly) increased while the test performance is marginally decreased.

In the high range of the number of epochs (higher than 1500), the performance of the simple and complex options is very close. Specifically, when the number of epochs is 1500 (i.e., around the peak performance), the gain of the complex option is 0.005 for PCC, 0.017 for RMSE, and 0.004 for SROCC. As the simple option has much less complexity while achieving a high performance, it can be said that the simple option *I2* is both efficient and effective. In addition, this result implies that bitstream-level parameters can replace segment quality and content characteristics.

For in-depth understanding of the input options, the comparison is also conducted using a simple regression model of SVR instead of the LSTM network. The result is shown in Table V. It can be seen that similar conclusions can also be made. In particular, the performance of the complex option *I1* (*Full* case) and the combination option *I1+I2* is comparable and both are just slightly higher than that of the simple option *I2*. In addition, the crucial roles of the segment quality and the stalling duration features are again confirmed since the performance of the *w/oSQ* and the *w/oSD* cases is considerably lower compared to the *Full* case. More discussions on the input options will be made in Section VIII.

VI. Analysis of LSTM networks

In this section, we will evaluate the performance of the two LSTM network types, using the different options of input features. Also, different issues that are inherent in an LSTM network, namely padding manners, the numbers of epochs, and the number of hidden units, will be investigated.

A. Evaluation of baLSTM and adLSTM Networks

In this subsection, a comparison of the two LSTM network types and a baseline regression model (SVR) is presented with the complex input option (using *S-MOS* as the segment quality metric and four input cases *Full*, *w/oCC*, *w/oSD*, and *w/oSQ*), the simple input option, and the combination option. The number of hidden units d is set to 5. Similar to Section V, the performance reported in this part is also the best performance when the number of epochs e is from 500 to 7000. The reason is that the optimal number of epochs may be different for different options of the feature set as well as network types. Table V shows the obtained results for each LSTM network type and the SVR model. It can be seen that, with the same input option, both the LSTM networks achieve significantly higher performance than the SVR model. The maximum gain in terms of PCC, RMSE, and SROCC is respectively 0.201, 0.333, and 0.164 for the *adLSTM* network, and 0.248, 0.392, and 0.179 for the *baLSTM* network. This result implies that the LSTM networks are much more effective than the SVR model in pooling segment features to predict the overall quality.

In comparison to the *baLSTM* network, the performance of the *adLSTM* network is generally equal or higher. In particular, the difference between them is small for the complex option with the *Full* and the *w/oCC* cases, the simple option, and the combination option. However, it is significant for the complex option with the *w/oSD* and the *w/oSQ* cases. The gain of the *adLSTM* network is up to 0.029 for PCC, 0.017 for RMSE, and 0.033 for SROCC in comparison to the *baLSTM* network. However, since the *adLSTM* network additionally includes the backward layer, its number of parameters is approximately 2 times higher than that of the *baLSTM* network. In particular, the number of parameters which are learned in the training process is $d(4M + 4d + 5) + 1$ for the *baLSTM* network and $d(8M + 8d + 9) + 1$ for the *adLSTM* network.

Therefore, to obtain the highest performance regardless of input option, the *adLSTM* network should be used. Meanwhile, for a simple computation, the *baLSTM* network can also be employed for the *Full* case of the complex option or for the simple option. More discussions on the LSTM network types will be made in Section VIII. In the following, the investigation will be based on the *adLSTM* network unless otherwise stated.

B. Impact of Padding

In this work, we investigate the performance of the proposed approach with four different padding cases. The first case, denoted *prePadd*, adds padded segments to the beginning of every session as presented in Subsection III-A.4. In the second case, denoted *postPadd*, the added segments are appended to the end of every session. For both of the cases, the length of sessions after padding is the same as the length of the longest session. Regarding the two remaining cases, denoted *addPadd10* and *addPadd30*, the padded segments are inserted to the beginning of every session until the length of that session is equal to the length of the longest session plus K segments. K is set to 10 segments for the *addPadd10* case and 30 segments for the *addPadd30* case.

TABLE VI

Best Performance of the Proposed Approach Using the Option *I1* With Different Padding Cases for the Test Sets

Padding cases	PCC (↑)	RMSE (↓)	SROCC (↑)
prePadd	0.966	0.248	0.965
postPadd	0.966	0.250	0.965
addPadd10	0.965	0.252	0.964
addPadd30	0.966	0.250	0.964

The best performance corresponding the different padding cases for the test sets is shown in Table VI. Here, the number of hidden units is set to 5. Interestingly, the performance differences between the cases are trivial. In particular, the PCC, RMSE, and SROCC values are respectively 0.966, 0.248, and 0.965 for the *prePadd* case, 0.966, 0.250, and 0.965 for the *postPadd* case, 0.965, 0.252, and 0.964 for the *addPadd10* case, and 0.966, 0.250, and 0.964 for the *addPadd30* case. This result suggests that there is no significant impact of padded segments as well as their positions on the performance of the proposed approach.

C. Impacts of Numbers of Epochs and Hidden Units

In this part, we will investigate the impacts of the number of epochs and the number of hidden units on the performance of the proposed approach. Fig. 12 shows the PCC, RMSE, and SROCC values at different numbers of epochs and different numbers of hidden units using the option *I1* (*Full* case) and the option *I2*. It can be seen that, in general, the PCC and SROCC values increase quickly and the RMSE values reduce rapidly when the number of epochs e first increases. When the number of epochs increases further, the PCC, SROCC, and RMSE values become stable. Also, we can see that the higher the number of hidden units is, the faster the stable state is reached. In particular, the performance is almost unchanged for $e \geq 2500$ ($d = 1$), $e \geq 1500$ ($d = 3$), and $e \geq 1500$ ($d = 5$) for the *Full* case of the option *I1*. With the option *I2*, the stable state is obtained when $e \geq 3000$ ($d = 1$), $e \geq 2500$ ($d = 3$), and $e \geq 1500$ ($d = 5$).

Note that, when $e \geq 3500$ for the option *I1* and $e \geq 5000$ for the option *I2*, the test performance gradually diminishes. This is due to over-learning of the network when en-longing the training process [43]. This result suggests that, when training the proposed approach, the setting of $d = 5$ and $e \in [1500, 3500]$ provides good and stable performance for both the options.

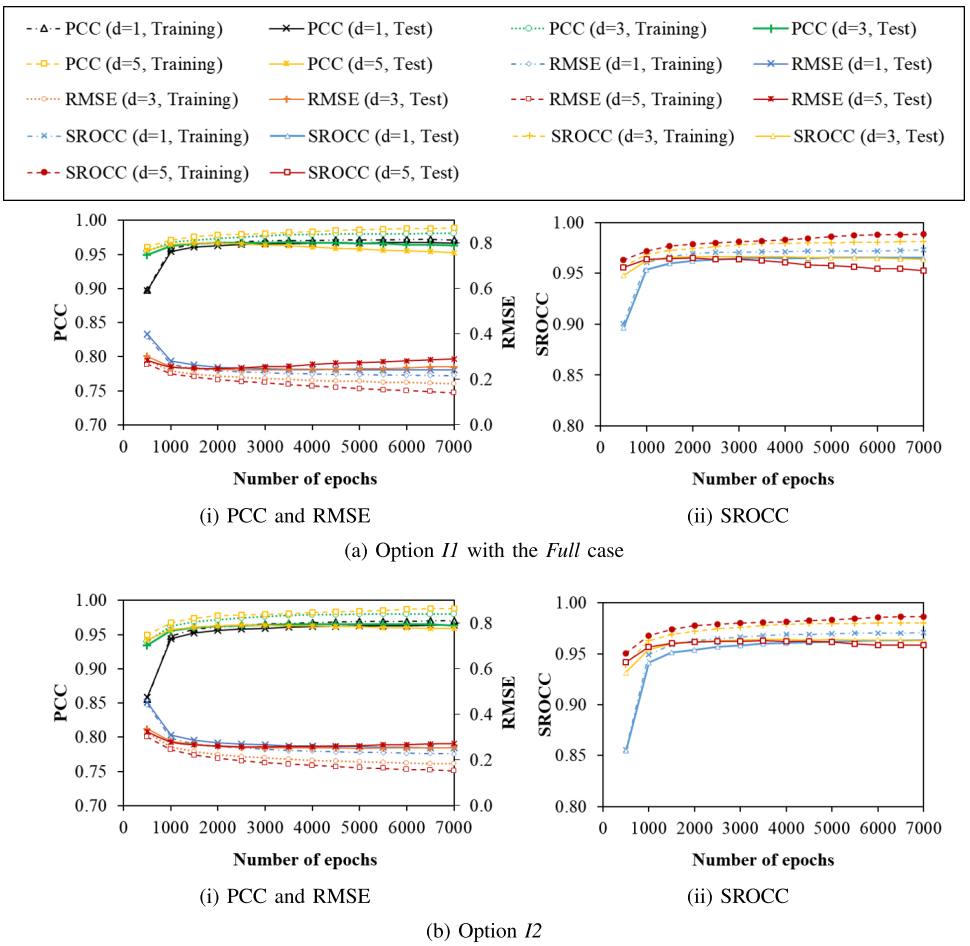


Fig. 12. Prediction performance of the proposed approach at different numbers of epochs and numbers of hidden units using the option *I1* (with the *Full* case) and the option *I2*.

VII. Comparison with Existing Approaches

In this part, we compare the performance of the proposed approach to seven existing approaches, namely *Guo's* [12], *Liu's* [14], *Tran's* [11], *Singh's* [15], *ATLAS* [16], *P.1203.3* [21], and *Eswara's* [7]. Among the reference approaches, the first three are analytical model-based approaches. The rest are advanced machine learning approaches. Note that *Guo's* approach only takes into account the impact of quality variations while the remaining approaches consider both quality variations and stalling events.

Similar to [18], [44], we implemented the four approaches of *Guo's*, *Liu's*, *Tran's* and *Singh's* based on the corresponding publications. The reason is that the implementations of these approaches are not publicly available. For the *ATLAS* and *Eswara's* approaches, we used the implementation publicized by the original authors [7][1] [16].[2] Note that, to obtain the overall quality values, *Eswara's* approach uses a mean pooling strategy of instantaneous quality values [7]. For the *P.1203.3* approach, we use an implementation of the standard that is free to use for research purposes [22], [45].[3]

[1]https://github.com/lfovia/lstm_qoe
[2]http://live.ece.utexas.edu/research/quality/VideoATLAS_release.zip
[3]https://github.com/itu-p1203/itu-p1203/

Note that the parameters in the approaches of *Guo's*, *Liu's*, *Tran's*, *P.1203.3*, and *Eswara's* are set to the values stated in the corresponding publications without re-training. Therefore, before evaluating the performance of these approaches, a compensation for differences in subjective test conditions is conducted for each approach using a first-order linear regression following Recommendations ITU-T P.1401 [46] and ITU-T P.1203 [21]. The adjustment coefficients (i.e., slopes and intercepts) are reported in Table VII.

For the proposed approach, the basic and advanced LSTM networks are used with both the simple option and the complex option (using the *Full* case and *S-MOS* quality metric). The number of epochs e and the number of hidden units d are set to 1500 and 5, respectively. Note that, before evaluating the performance of the two advanced machine learning approaches of *Singh's* and *ATLAS*, their parameters are (re-)trained in the same way as the proposed approach.

In order to compare the performance of the considered approaches, we use three different datasets. The first is our dataset, where the PCC, RMSE, and SROCC values are averaged over the 100 test sets mentioned in Subsection IV-B. The second, third, and fourth columns of Table VII show the obtained results of the approaches using our dataset.

The second consists of a training set, called *TR04*, and a test set, called *VL04*, which are publicized from the

TABLE VII

PREDICTION PERFORMANCE OF THE PROPOSED APPROACH AND SEVEN EXISTING APPROACHES FOR THE TEST SETS OF THE DATASETS. THE BOLD NUMBERS INDICATE THE HIGHEST PERFORMANCE FOR EACH DATASET. THE UNDERLINED NUMBERS SHOW THE HIGHEST PERFORMANCE AMONG THE REFERENCE APPROACHES.)

(a) Our dataset and P.1203 dataset

Approach	Our dataset					P.1203 dataset				
	PCC (↑)	***RMSE*** (↓)	***SROCC*** (↑)	***Slope***	***Intercept***	***PCC*** (↑)	***RMSE*** (↓)	***SROCC*** (↑)	***Slope***	***Intercept***
Guo's	0.438	0.847	0.430	0.868	1.362	0.606	0.709	0.656	1.106	1.297
Liu's	0.575	0.938	0.632	-0.027	3.000	N/A	N/A	N/A	N/A	N/A
Tran's	0.905	0.401	0.925	1.107	-0.588	0.843	0.479	0.855	0.970	0.247
Singh's	0.723	0.653	0.715	—	—	0.216	0.871	0.150	0.303	2.071
ATLAS	0.879	0.453	0.896	—	—	N/A	N/A	N/A	N/A	N/A
P.1203.3	0.913	0.382	0.909	1.133	-1.302	0.884	0.416	0.868	1.041	0.075
Eswara's	0.652	1.061	0.674	0.192	3.272	N/A	N/A	N/A	N/A	N/A
Proposed (*I1* + *baLSTM*)	0.963	0.265	0.962	—	—	0.906	0.378	0.907	0.957	0.480
Proposed (*I2* + *baLSTM*)	0.950	0.305	0.948	—	—	0.870	0.435	0.878	0.781	0.864
Proposed (*I1* + *adLSTM*)	**0.966**	**0.250**	**0.965**	—	—	**0.915**	**0.359**	**0.914**	0.905	0.654
Proposed (*I2* + *adLSTM*)	0.961	0.267	0.960	—	—	0.900	0.387	0.900	0.794	0.795

(b) WaterlooSQoE-II

Approach	WaterlooSQoE-II-Case#1					WaterlooSQoE-II-Case#2				
	PCC (↑)	***RMSE*** (↓)	***SROCC*** (↑)	***Slope***	***Intercept***	***PCC*** (↑)	***RMSE*** (↓)	***SROCC*** (↑)	***Slope***	***Intercept***
Guo's	0.707	0.506	0.721	2.129	0.220	0.707	0.503	0.723	2.129	0.220
Liu's	0.631	0.555	0.634	0.048	0.394	0.616	0.561	0.617	0.048	0.394
Tran's	0.850	0.376	**0.864**	0.790	0.695	0.848	0.376	0.862	0.790	0.695
Singh's	0.430	0.650	0.370	0.386	2.179	0.423	0.931	0.373	—	—
ATLAS	0.720	0.500	0.750	0.760	1.190	0.735	0.501	0.768	—	—
P.1203.3	0.857	0.368	0.853	0.882	-0.396	0.852	0.371	0.848	0.882	-0.396
Eswara's	0.442	1.393	0.456	0.069	4.149	0.407	1.390	0.416	0.063	4.150
Proposed (*I1* + *baLSTM*)	0.759	0.452	0.752	4.304	-11.258	**0.901**	**0.333**	**0.897**	—	—
Proposed (*I2* + *baLSTM*)	0.222	0.690	0.229	0.261	2.191	0.814	0.446	0.807	—	—
Proposed (*I1* + *adLSTM*)	**0.859**	**0.366**	0.861	3.376	-8.658	0.893	0.342	0.888	—	—
Proposed (*I2* + *adLSTM*)	0.300	0.668	0.310	0.812	0.197	0.811	0.460	0.806	—	—

ITU-T P.1203 standardization (P.NATS) [22].[4] Each set is composed of 60 sessions generated from three 1-minute long videos. Regarding *Singh's* approach and the proposed approach, we use all the 515 sessions in our dataset and the 60 sessions in the *TR04* set for (re)-training their parameters to avoid dependencies in data divisions. In addition, to account for the impact of weight initialization, the training process is repeated 100 times. Accordingly, the performance is averaged over the test set *VL04*. The results are shown in the seventh, eighth, and ninth columns of Table VII. Note that the three approaches of *Liu's*, *ATLAS*, and *Eswara's* are not evaluated over this dataset because of the lack of input data. Also, because the content characteristic feature is not publicly available, the option *I1* with the *w/oCC* case is evaluated, instead of the *Full* case.

The final dataset, called *WaterlooSQoE-II*, contains 588 sessions with 4-second long segments generated from twelve 8-second long source videos with diverse quality variation patterns of QP, resolution, and frame-rate [19]. To evaluate the performances of the approaches, we consider two cases of using this dataset. In the first case (denoted *WaterlooSQoE-II-Case#1*), the training set is our dataset, and the test set is the whole *WaterlooSQoE-II* dataset. For the second case (denoted *WaterlooSQoE-II-Case#2*), beside our dataset, the training set additionally includes all the 147 sessions generated from three of the twelve source videos in the *WaterlooSQoE-II* dataset. The 441 remaining sessions from the other videos constitute the corresponding test set. The selection of the three training videos is repeated 100 times to produce 100 different pairs of the training and test sets. Similar to the evaluation using the two above datasets, the average PCC, RMSE, and SROCC values are calculated over the 100 different test times and reported in Table VII. Note that, since our dataset, which is also included in the training set, was built using 1-second long segments, each 4-second long segment in the *WaterlooSQoE-II* dataset is considered as four 1-second long segments in our experiment.

From Table VII, it can be seen that, for three of the four datasets/cases (i.e., except *WaterlooSQoE-II-Case#1*), the proposed approach using the *adLSTM* network with the option *I1* always achieves the highest performance compared to the reference approaches (i.e., PCC$\geq$0.893, RMSE$\leq$0.359, SROCC$\geq$0.888). For the *WaterlooSQoE-II-Case#1* case, it also obtains the highest PCC value and the lowest RMSE value (i.e., PCC=0.859 and RMSE=0.366). Besides, its SROCC value is just slightly lower than the highest one (i.e., 0.861 vs. 0.864), which is obtained by *Tran's* approach. Hence, this result indicates that our approach outperforms all the seven reference approaches.

Compared to the *adLSTM* network, the performance of the *baLSTM* network is generally lower with the same input option, especially for the P.1203 dataset and *WaterlooSQoE-II-Case#1* case. In particular, the maximum difference is 0.100 for PCC, 0.086 for RMSE, and 0.109 for SROCC. In comparison to the existing approaches, the performance of the *baLSTM* network is significantly higher when using the option *I1* for most of the datasets/cases (i.e., except *WaterlooSQoE-II-Case#1*). For the option *I2*, its performance is also considerably higher for our dataset, but slightly lower than that of the *P.1203.3* approach for the P.1203 dataset.

[4]https://github.com/itu-p1203/open-dataset

For the *WaterlooSQoE-II-Case#1* case, the use of the option *I2* results in very low performances for both the LSTM networks (i.e., PCC$\leq$0.300, RMSE$\geq$0.668, SROCC$\leq$0.310). The reason is that quality variations in frame-rate, which are not included in the training set (i.e., our dataset), are presented in the test set (i.e., the *WaterlooSQoE-II* dataset). However, a significant performance improvement is found in the *WaterlooSQoE-II-Case#2* case when streaming sessions of varying frame-rate are added in the training set (i.e., PCC$\geq$0.811, RMSE $\leq$0.460, SROCC$\geq$0.806). Since the *baLSTM* network with the option *I1* always achieves a quite high performance for most of the datasets/cases (i.e., PCC$\geq$0.759, RMSE$\leq$0.452, and SROCC$\geq$0.752), the *baLSTM* network can also be used in the proposed approach.

Although both our and *Eswara's* approaches use LSTM networks, the proposed approach with the option *I1* and any LSTM network has significantly higher performance. This may be because that *Eswara's* approach is actually proposed to predict the instantaneous quality, but not the overall quality. In addition, the mean pooling strategy may be not effective to aggregate instantaneous quality values into an overall quality value.

In addition, we can see that, for *Tran's* approach and the *P.1203.3* approach, their performances are quite good. Specifically, the PCC values are equal to or higher than 0.843, the RMSE values are equal to or smaller than 0.479, and the SROCC values are not lower than 0.848. This result suggests that these approaches also achieve good prediction performances.

It can also be seen that, in general, the analytical model-based approaches have lower prediction performance than the advanced machine learning approaches. Interestingly, *Tran's* approach has higher performance than *Singh's* approach and the *ATLAS* approach. This result implies that, in order to quantify the impacts of quality variations and stalling events, the histograms of quality switches and stalling durations used in *Tran's* approach are more effective than the statistics used in *Singh's* approach and the *ATLAS* approach such as the average of segment quality values, the number of stalling events, the average and the maximum of stalling durations.

VIII. Discussions

From the above experimental results, it is obvious that the use of feature inputs taken on a segment basis and an LSTM network is very effective in predicting the overall quality of HTTP adaptive streaming sessions. Among the considered features of the option *I1*, the role of the segment quality feature is the most important, followed by the stalling duration feature. The role of the content characteristic feature depends on the metric used to represent the segment quality feature.

To represent the segment quality feature, segment-MOS (*S-MOS*) is found to be the best metric. In addition, the simple *PSNR* metric can also be used with a good performance (in the *Full* case of the complex input option).

The employment of content characteristics is a new and interesting point in our study. As mentioned, when using the *S-MOS* metric to represent the segment quality feature, the additional use of the content characteristic feature does not bring significant improvements in prediction performance. Meanwhile, for the metrics of *BR*, *PSNR*, *PSNR*-variants, *SSIM*, *MS-SSIM*, and *VIF*, it is beneficial to include the content characteristic feature. So, in some sense, the *S-MOS* metric already includes the impact of content characteristics.

Another related issue is the good performance of the simple input option although it does not explicitly contain content characteristics. This can be explained as follows. Given certain values of QP, resolution, and frame-rate, a video with more complex content characteristics generally results in a higher bitrate. This means the bitstream-level parameters used in this option can implicitly represent the complexity (or content characteristics) of a video. So, the simple option *I2*, which simply consists of bitstream-level parameters, stalling, and padding, is both efficient and effective. In the future work, instead of using manually-selected features, we will focus on deep learning approaches to automatically obtain effective input features for the LSTM networks.

Regarding LSTM network types, the use of the advanced network achieves a higher performance in comparison to the basic network. For the advanced network, the setting of the number of hidden units $d = 5$ and the number of epochs $e \in [1500, 3500]$ provides good and stable performance. However, the performance of the two networks is in fact quite close. In addition, the number of parameters in the basic network is only about half compared to the advanced network. Especially, the key advantage of the basic network is the capability to process using current and past segments only. This allows us to continuously predict the quality of a on-going session without having to wait until the end of the session.

Also, LSTM networks can help take into account all variations in the temporal dimension, so the addition of the switching frequency as a feature does not bring improvements in the performance. This could be a key reason that the proposed approach outperforms the existing approaches.

In this study, the used datasets include about 1-minute long sessions with 1-second long segments and 8-second long sessions with 4-second long segments, which are encoded using H.264/AVC. In practice, service providers can use different segment durations (e.g., 2 seconds) and other video codecs (e.g., H.265 and VP9). However, obviously, such a long segment could be divided into multiple short segments of a 1-second duration. By this way, the proposed approach can be applied by inputting the features of these short segments. In future work, we plan to evaluate our approach using various segment durations and session lengths as well. In addition, an evaluation using other video codecs will be an interesting direction to verify and improve the proposed approach.

Regarding the computation complexity, we measured the average time to obtain an overall quality value given a session duration. In total, we used 1223 sessions with the lengths from 8s to 76s, which are all the sessions in the three datasets employed in our above performance evaluations. This measurement was conducted on a computer with Intel Core i3-3240 processor at 3.40GHz and with 8GB RAM. For all the cases of LSTM networks and input options, the time complexities of the proposed approach for the session durations

of 8s, 60s, and 76s are respectively less than 1.5ms, 4.5ms, and 5.5ms. In general, the computation complexity of the proposed approach increases linearly with the session duration. In our future work, we will try to reduce the time complexity of this LSTM-based approach for real-time quality monitoring.

IX. Conclusion

In this study, we have proposed a new machine learning approach for predicting the overall quality of HTTP adaptive streaming sessions using Long Short Term Memory (LSTM) network. The proposed approach takes into account some features such as segment quality, content characteristics, stalling duration, and padding. Through experimental results, we found that the proposed approach achieves very high performance in the overall quality prediction of HTTP Adaptive Streaming sessions. Based on an extensive evaluation of input options and network settings, some interesting findings were gained, such as the role of the input features, the best metric to represent the segment quality feature, the effective and efficient input option, and the optimal setting of the LSTM network. In future work, we plan to evaluate the proposed approach using various segment durations, session lengths, and video codecs. For this purpose, it is necessary to construct a new dataset. In addition, we intend to employ the proposed approach in performance evaluations of adaptation strategies for HTTP adaptive streaming.

References

[1] T. C. Thang, H. T. Le, A. T. Pham, and Y. M. Ro, "An Evaluation of Bitrate Adaption Methods for. [Online]. Available: http Live Streaming," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 4, pp. 693–705, Apr. 2014.

[2] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia, "A survey on quality of experience of. [Online]. Available: http adaptive streaming," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 469–492, 2015.

[3] *Methodology for the Subjective Assessment of the Quality of Television Pictures*, document Recommendation ITU-R BT.500-13Int. Telecommun. Union, vol. 13, 2012, pp. 1–48.

[4] *Methods for Objective and Subjective Assessment of Quality: Continous Evaluation of Time Varying Speech Quality*, document Recommendation ITU-T P.880 Int. Telecommun. Union, 2004.

[5] B. Weiss, D. Guse, S. Möller, A. Raake, A. Borowiak, and U. Reiter, "Temporal development of quality of experience," in *Quality of Experience: Advanced Concepts, Applications and Methods*. Cham, Switzerland: Springer, 2014, pp. 133–147.

[6] C. G. Bampis, Z. Li, I. Katsavounidis, and A. C. Bovik, "Recurrent and dynamic models for predicting streaming video quality of experience," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3316–3331, Jul. 2018.

[7] N. Eswara *et al.*, "Streaming video QoE modeling and prediction: A long short-term memory approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 3, pp. 661–673, Mar. 2020.

[8] N. Eswara *et al.*, "A continuous QoE evaluation framework for video streaming over HTTP," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 11, pp. 3236–3250, Nov. 2018.

[9] D. Ghadiyaram, J. Pan, and A. C. Bovik, "A subjective and objective study of stalling events in mobile streaming videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 183–197, Jan. 2019.

[10] T. Alpert and J. Evain, "Subjective quality evaluation: The SSCQE and DSCQE methodologies," *EBU Tech. Rev.*, pp. 12–20, Feb. 1997.

[11] H. T. T. Tran, N. P. Ngoc, A. T. Pham, and T. C. Thang, "A multi-factor QoE model for adaptive streaming over mobile networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2016, pp. 1–6.

[12] Z. Guo, Y. Wang, and X. Zhu, "Assessing the visual effect of non-periodic temporal variation of quantization stepsize in compressed video," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 3121–3125.

[13] T. Hossfeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch, and C. Lorentzen, "Initial delay vs. Interruptions: Between the devil and the deep blue sea," in *Proc. 4th Int. Workshop Qual. Multimedia Exper.*, Jul. 2012, pp. 1–6.

[14] Y. Liu, S. Dey, F. Ulupinar, M. Luby, and Y. Mao, "Deriving and validating user experience model for DASH video streaming," *IEEE Trans. Broadcast.*, vol. 61, no. 4, pp. 651–665, Dec. 2015.

[15] K. D. Singh, Y. Hadjadj-Aoul, and G. Rubino, "Quality of experience estimation for adaptive HTTP/TCP video streaming using H.264/AVC," in *Proc. IEEE Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2012, pp. 127–131.

[16] C. G. Bampis and A. C. Bovik, "Feature-based prediction of streaming video QoE: Distortions, stalling and memory," *Signal Process., Image Commun.*, vol. 68, pp. 218–228, Oct. 2018.

[17] C. Chen, L. Kwon Choi, G. de Veciana, C. Caramanis, R. W. Heath, and A. C. Bovik, "Modeling the time—Varying subjective quality of HTTP video streams with rate adaptations," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2206–2221, May 2014.

[18] Z. Duanmu, A. Rehman, and Z. Wang, "A Quality-of-Experience database for adaptive video streaming," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 474–487, Jun. 2018.

[19] Z. Duanmu, K. Ma, and Z. Wang, "Quality-of-experience for adaptive streaming videos: An expectation confirmation theory motivated approach," *IEEE Trans. Image Process.*, vol. 27, no. 12, pp. 6135–6146, Dec. 2018.

[20] D. Ghadiyaram, A. C. Bovik, H. Yeganeh, R. Kordasiewicz, and M. Gallant, "Study of the effects of stalling events on the quality of experience of mobile streaming videos," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Dec. 2014, pp. 989–993.

[21] *Parametric Bitstream-Based Quality Assessment of Progressive Download and Adaptive Audiovisual Streaming Services Over Reliable Transport-Quality Integration Module*, document Recommendation ITU-T P.1203.3, Int. Telecommun. Union, 2017.

[22] W. Robitza *et al.*, "HTTP adaptive streaming QoE estimation with ITU-T rec. P. 1203: Open databases and software," in *Proc. 9th ACM Multimedia Syst. Conf.*, Jun. 2018, p. 1203.

[23] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognit. Lett.*, vol. 42, pp. 11–24, Jun. 2014.

[24] G. White, A. Palade, and S. Clarke, "Forecasting QoS attributes using LSTM networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.

[25] H. T. T. Tran, D. V. Nguyen, D. D. Nguyen, N. P. Ngoc, and T. C. Thang, "An lstm-based approach for overall quality prediction in. [Online]. Available: http adaptive streaming," in *IEEE Comput. Commun. Conf. Workshops*, Paris, Apr. 2019, pp. 702–707.

[26] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," 2014, *arXiv:1409.2329*. [Online]. Available: http://arxiv.org/abs/1409.2329

[27] S. Tavakoli, S. Egger, M. Seufert, R. Schatz, K. Brunnstrom, and N. Garcia, "Perceptual quality of HTTP adaptive streaming strategies: Cross-experimental analysis of multi-laboratory and crowdsourced subjective studies," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 8, pp. 2141–2153, Aug. 2016.

[28] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[29] P. Zhou *et al.*, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics (Short Papers)*, vol. 2, 2016, pp. 207–212.

[30] K. Egiazarian, J. Astola, N. Ponomarenko, V. Lukin, F. Battisti, and M. Carli, "New full-reference quality metrics based on HVS," in *Proc. Int. Workshop Video Process. Quality Metrics*, Scottsdale, AZ, USA, vol. 4, Jan. 2006, pp. 1–4.

[31] N. Ponomarenko, F. Silvestri, K. Egiazarian, M. Carli, J. Astola, and V. Lukin, "On between-coefficient contrast masking of DCT basis functions," in *Proc. Int. Workshop Video Process. Qual. Metrics*, Scottsdale, AZ, USA, vol. 4, Jan. 2007, pp. 1–4.

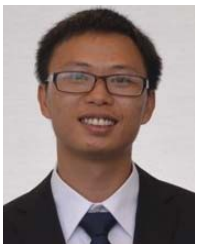
[32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[33] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2003, pp. 1398–1402.

[34] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 430–444, Feb. 2006.

[35] H. Sohn, H. Yoo, W. De Neve, C. S. Kim, and Y. M. Ro, "Full-reference video quality metric for fully scalable and mobile SVC content," *IEEE Trans. Broadcast.*, vol. 56, no. 3, pp. 269–280, Sep. 2010.
[36] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–703, Jul. 2003.
[37] H. T. Tran, N. P. Ngoc, Y. J. Jung, A. T. Pham, and T. C. Thang, "A histogram-based quality model for HTTP adaptive streaming," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. 100, no. 2, pp. 555–564, 2017.
[38] T. C. Thang, H. T. Le, H. X. Nguyen, A. T. Pham, J. W. Kang, and Y. M. Ro, "Adaptive video streaming over HTTP with dynamic resource estimation," *J. Commun. Netw.*, vol. 15, no. 6, pp. 635–644, 2013.
[39] P. Juluri, V. Tamarapalli, and D. Medhi, "SARA: Segment aware rate adaptation algorithm for dynamic adaptive streaming over HTTP," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, Jun. 2015, pp. 1765–1770.
[40] *Methods for the Subjective Assessment of Video Quality, Audio Quality and Audiovisual Quality of Internet Video and Distribution Quality Television in Any Environment*, document Recommendation ITU-T P.913, Int. Telecommun. Union, 2014.
[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[42] T. Hossfeld, M. Seufert, C. Sieber, and T. Zinner, "Assessing effect sizes of influence factors towards a QoE model for HTTP adaptive streaming," in *Proc. 6th Int. Workshop Qual. Multimedia Exper. (QoMEX)*, Sep. 2014, pp. 111–116.
[43] A. Shukla, R. Tiwari, and R. Kala, *Towards Hybrid and Adaptive Computing: A Perspective*, vol. 307. Berlin, Germany: Springer, 2010.
[44] Z. Duanmu, K. Ma, and Z. Wang, "Quality-of-experience of adaptive video streaming: Exploring the space of adaptations," in *Proc. ACM Multimedia Conf. MM*, 2017, pp. 1752–1760.
[45] A. Raake, M.-N. Garcia, W. Robitza, P. List, S. Goring, and B. Feiten, "A bitstream-based, scalable video-quality model for HTTP adaptive streaming: ITU-T P.1203.1," in *Proc. 9th Int. Conf. Qual. Multimedia Exper. (QoMEX)*, May 2017, p. 1203.
[46] *Methods, Metrics and Procedures for Statistical Evaluation, Qualification and Comparison of Objective Quality Prediction Models*, document Recommendation ITU-T P.1401, Int. Telecommun. Union, 2012.

Huyen T. T. Tran (Member, IEEE) received the B.E. degree from the Hanoi University of Science and Technology, Vietnam, in 2014, and the M.Sc. and Ph.D. degrees from The University of Aizu, Japan, in 2017 and 2020, respectively. She is currently a Post-Doctoral Researcher with the Department of Computer and Information Systems, The University of Aizu. Her research interests include quality of experience (QoE), multimedia networking, and content adaptation. She was a recipient of Japanese Government Scholarship (MonbuKagaku-sho) for graduate study from 2015 to 2020. In 2017, she received the IEEE Signal Processing Society (SPS) student travel grant. She is also serves as a Reviewer for IEEE ACCESS, IEEE TRANSACTIONS ON BROADCASTING, and IEEE TRANSACTIONS ON IMAGE PROCESSING.



Duc V. Nguyen (Member, IEEE) received the B.E., M.E., and Ph.D. degrees in computer science and engineering from the University of Aizu, Japan, in 2014, 2016, and 2019, respectively. Since 2019, he has been working with KDDI Research Inc. His research interests include video streaming, virtual reality, and networking.

Nam Pham Ngoc (Member, IEEE) received the B.E. degree in electronics and telecom from the Hanoi University of Science and Technology, Vietnam, in 1997, and the M.Sc. and Ph.D. degrees from KU Leuven, Belgium, in 1999 and 2004, respectively. Since 2004, he has been working with the Hanoi University of Science and Technology, Vietnam. His research interests include multimedia applications and embedded system design.

Truong Cong Thang (Senior Member, IEEE) received the B.E. degree from the Hanoi University of Science and Technology, Vietnam, in 1997, and the Ph.D. degree from KAIST, South Korea, in 2006. From 1997 to 2000, he worked as a Network Engineer with Vietnam Post and Telecom (VNPT). From 2007 to 2011, he was a member of Research Staff with the Electronics and Telecommunications Research Institute (ETRI), South Korea. He was also an active member of Korean and Japanese delegations to standard meetings of ISO/IEC and ITU-T from 2002 to 2014. Since 2011, he has been an Associate Professor with The University of Aizu, Japan. His research interests include multimedia networking, image/video processing, content adaptation, IPTV, and MPEG/ITU standards.