Convolutional Neural Networks for Continuous QoE Prediction in Video Streaming Services

THO NGUYEN DUC[1], **CHANH TRAN MINH**[1], **TAN PHAN XUAN**[2],
AND EIJI KAMIOKA[1], **(Member, IEEE)**

[1]Graduate School of Engineering and Science, Shibaura Institute of Technology, Tokyo 135-8548, Japan
[2]Department of Information and Communications Engineering, Shibaura Institute of Technology, Tokyo 135-8548, Japan

Corresponding authors: Tho Nguyen Duc (mg18502@shibaura-it.ac.jp) and Tan Phan Xuan (tanpx@shibaura-it.ac.jp)

ABSTRACT In video streaming services, predicting the continuous user's quality of experience (QoE) plays a crucial role in delivering high quality streaming contents to the user. However, the complexity caused by the temporal dependencies in QoE data and the non-linear relationships among QoE influence factors has introduced challenges to continuous QoE prediction. To deal with that, existing studies have utilized the Long Short-Term Memory model (LSTM) to effectively capture such complex dependencies, resulting in excellent QoE prediction accuracy. However, the high computational complexity of LSTM, caused by the sequential processing characteristic in its architecture, raises a serious question about its performance on devices with limited computational power. Meanwhile, Temporal Convolutional Network (TCN), a variation of convolutional neural networks, has recently been proposed for sequence modeling tasks (e.g., speech enhancement), providing a superior prediction performance over baseline methods including LSTM in terms of prediction accuracy and computational complexity. Being inspired of that, in this paper, an improved TCN-based model, namely CNN-QoE, is proposed for continuously predicting the QoE, which poses characteristics of sequential data. The proposed model leverages the advantages of TCN to overcome the computational complexity drawbacks of LSTM-based QoE models, while at the same time introducing the improvements to its architecture to improve QoE prediction accuracy. Based on a comprehensive evaluation, we demonstrate that the proposed CNN-QoE model can provide a high QoE prediction performance on both personal computers and mobile devices, outperforming the existing approaches.

INDEX TERMS Convolutional neural networks, temporal convolutional network, quality of experience, video streaming.

I. INTRODUCTION

For years, video streaming services have increasingly become the most dominant services on the Internet, creating an extremely huge profit for streaming service providers. Within such a highly competitive streaming service market, service providers such as YouTube, Netflix, or Amazon must provide a sufficient video quality to satisfy the viewer's expectation, resulting in a high quality of experience (QoE). However, video streaming services are frequently influenced by dynamic network conditions that can lead to distorted events, subsequently causing QoE deterioration. Therefore, developing QoE models that quickly and accurately predict the user's QoE in real-time can significantly benefit QoE-aware

The associate editor coordinating the review of this manuscript and approving it for publication was Anton Kos.

applications. By relying on a QoE model, for instance, a stream-switching controller designed at a client-side [1], [2] with an adaptive bitrate selection algorithm can adaptively predict and request an optimal video quality level. However, the continuous evaluation of QoE is challenging since it needs to capture the complex temporal dependencies in sequential QoE data and the non-linear relationships among QoE influencing factors (e.g., video quality, bitrate switching, and rebuffering) [3]–[6]. To deal with this challenge, a QoE prediction model which leverages Long Short-Term Memory (LSTM) [7] was introduced. The LSTM-based QoE prediction model achieved the state-of-the-art accuracy since it is capable of capturing temporal dependencies in sequential QoE data. However, the chain structure in the LSTM architecture requires a high computational cost for practically predicting the user's QoE due to the use of sequential

processing over time. It means that the subsequent processing steps must wait for the output from the previous ones. This leads to an open question about the performance of the model on power-limited computers like mobile devices that may not have enough computational power to implement such QoE-aware algorithms.

Recently, Temporal Convolutional Network (TCN) [8], a variation of Convolutional Neural Network (CNN), has emerged as a promisingly alternative solution for the sequence modeling tasks. TCN adopts dilated causal convolutions [9]–[11] to provide a powerful way of extracting the temporal dependencies in the sequential data. Different from LSTM, the computations in TCN can be performed in parallel, providing computational and modeling advantages.

In practical deployments, TCN convincingly outperforms canonical recurrent architectures including LSTMs and Gated Recurrent Units (GRUs) across a broad range of sequence modeling tasks [8]. Enlightened by the great ability of TCN, in this paper, we propose an improved TCN-based model, namely QoE-CNN, for continuous QoE prediction on different viewing devices (i.e., personal computers and mobile devices).

The goal of this study is to enhance the QoE prediction accuracy while minimizing the computational complexity to support a diversity of platforms and devices. The contributions of this paper are as follows:

- First, *CNN-QoE* model, an improved model of TCN for continuous QoE prediction in real-time, is proposed.
- Second, an optimal model architecture hyperparameters set for the proposed model is introduced to achieve the best QoE prediction performance.
- Third, a comprehensive evaluation of the CNN-QoE is performed across multiple QoE databases in comparison with different baseline methods. The results show that the CNN-QoE achieves superior performance in terms of accuracy and computational complexity on both personal computers and mobile devices.

The remainder of the paper is organized as follows: Section II describes the limitations of existing works for QoE modeling in video streaming. Section III discusses the TCN architecture in detail. The proposed model is presented in Section IV. Section V and VI provide evaluation results of the proposed model and their discussion, respectively. Finally, the paper is concluded in Section VII.

II. RELATED WORK

QoE modeling for video streaming services has received enormous attentions due to its critical importance in QoE-aware applications. A number of different continuous QoE prediction models have been proposed [4], [5], [12]–[18]. The authors in [4] modeled the time-varying subjective quality (TVSQ) using a Hammerstein-Wiener model. The work in [15] proposed a model based on the augmented Nonlinear Autoregressive Network with Exogenous Inputs (NARX) for continuous QoE prediction. It should be noted that these

models did not consider rebuffering events which usually happen in video streaming [19], [20]. On the other hand, the study in [14] took into account rebuffering events, perceptual video quality and memory-related features for QoE prediction. However, the QoE prediction accuracy varied across different playout patterns. The reason is that the model suffered from the difficulty in capturing the complex dependencies among QoE influence factors, leading to unreliable and unstable QoE prediction performances.

In order to address the above challenges, the authors in [7] proposed a QoE prediction model, namely, LSTM-QoE, which was based on Long Short-Term Memory networks (LSTM). The authors argued that the continuous QoE is dynamic and time-varying in response to QoE influencing events such as rebuffering [21] and bitrate adaptation [4]. To capture such dynamics, LSTM was employed and the effectiveness in modeling the complex temporal dependencies in sequential QoE data was shown. The model was evaluated on different QoE databases and outperformed the existing models in terms of QoE prediction accuracy. However, the computational complexity of the model was not fully inspected. Since the recurrent structure in LSTM can only process the task sequentially, the model failed to effectively utilize the parallel computing power of modern computers, leading to a high computational cost. Though compression techniques such as pruning and quantization [22], [23] can be utilized to reduce the complexity of the model, the original problem in the LSTM architecture still persists. Therefore, the efficiency of the model on different viewing devices with limited computing power (i.e., mobile devices) remains an open question.

Recently, a CNN architecture for sequence modeling, Temporal Convolutional Network (TCN) [8], was proposed. The dilated causal convolutions [9]–[11] in TCN enable it to efficiently capture the complex dependencies in a sequential data. The convolution operations can also be performed in parallel which effectively addresses the computational cost problem of LSTM. Besides, TCN has been successfully employed to tackle the complex challenges in sequence modeling tasks such as speech enhancement [8]. Therefore, in this paper, we present CNN-QoE, a continuous QoE prediction model based on TCN architecture, for improving the QoE prediction accuracy and optimizing the computational complexity.

III. TEMPORAL CONVOLUTIONAL NETWORK
In this section, TCN architecture is briefly discussed to summarize its advantages and disadvantages in sequence modeling tasks. Thereby, the conclusions of this section will be the crucial foundation for the subsequent improvements proposed in CNN-QoE, which are stated in section IV.

1) 1D CONVOLUTIONS
CNN was traditionally designed to operate on two dimensions (2D) data such as images. An input image is passed through a series of 2D convolution layers. Each 2D convolution applies

and slides a number of 2D filters through the image. To adapt CNN for time-series data, TCN utilizes 1D convolution where the filters exhibit only one dimension (time) instead of two dimensions (width and height). Concretely, a time-series input is convolved with a filter size of $1 \times k$.

Furthermore, 1D convolutions are well-suited for real-time tasks due to their low computational requirements. 1D convolutions require simple array operations rather than matrix operations, hence, the computational complexity is significantly reduced in comparison with 2D convolutions. In addition, the convolution operations allow fully parallel processing, resulting in a significant improvement of computational speed.

2) CAUSAL CONVOLUTIONS

A *causal convolution* is a convolution layer to ensure there is no information "leakage" from future into past. In other words, given an time-series input $x_0, \ldots, x_T$, the predicted output $\widehat{y}_t$ at a time instant t depends only on the inputs at time t and earlier $\mathbf{x}_t, \mathbf{x}_{t-1}, \ldots, \mathbf{x}_{t-r+1}$. For instance, as illustrated in Fig. 1, the predicted $\widehat{y}_8$ is computed by a combination of the inputs $x_1, \ldots, x_8$. It can be observed that, in order to achieve a long effective history size or a large receptive field size, an extremely deep network or very large filters are needed, which significantly increases the model computational complexity. Thus, TCN architecture utilizes dilated causal convolutions rather than causal convolutions. The advantages and disadvantages of dilated causal convolutions are discussed below.

3) DILATED CAUSAL CONVOLUTIONS

TCN adopts a dilated causal convolution comprising of the causal and the dilated convolutions. The causal convolution has already been described in the previous subsection. Meanwhile, dilated convolution [9]–[11] is a convolution where the convolution filter is applied to a larger area than its length by skipping input values with several steps. Therefore, the dilated causal convolution can effectively allow the network to operate on a larger scale than the one with a normal convolution while ensuring that there is no leakage of information from the future to the past. The dilated causal convolution is defined as:

$$D(t) = \sum_{i=0}^{k-1} f(i) \cdot \mathbf{x}_{t-d \cdot i} \quad (1)$$

where, d is the dilation factor, f is a filter size of $1 \times k$. d exponentially increases with the depth of the network (i.e., $d = 2^l$ at layer l of the network). For instance, given the network with L layers of dilated causal convolutions $l = 1, \ldots, L$, the dilation factors exponentially increase by a factor of 2 for every layer:

$$d \in [2^0, 2^1, \ldots, 2^{L-1}] \quad (2)$$

Fig. 2 depicts a network with three dilated causal convolutions for dilations 1, 2, and 4. Using the dilated causal
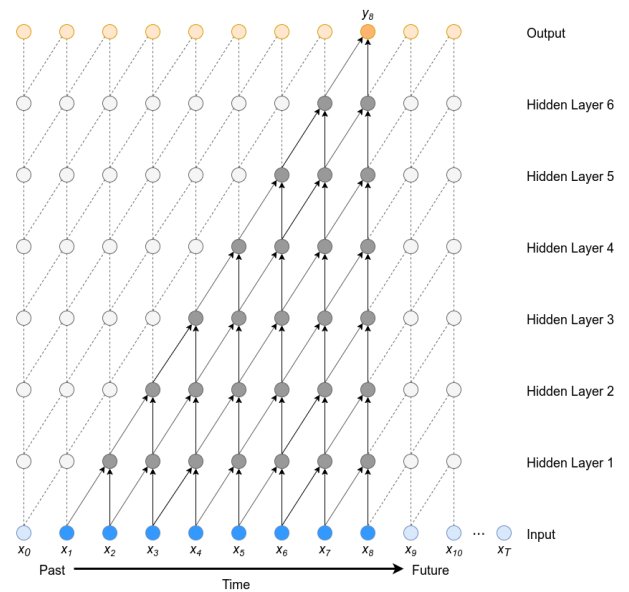

FIGURE 1. An illustration of a stack of causal convolution layers with the convolution filter size of 1 × 2.

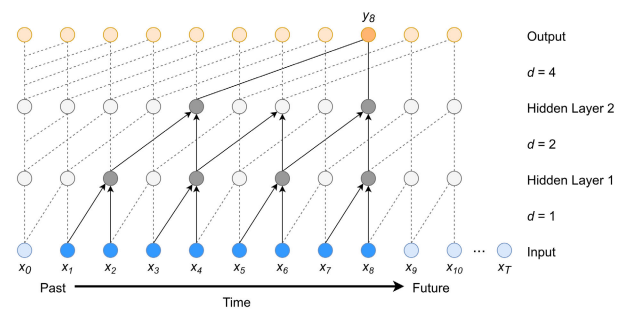**FIGURE 2.** An illustration of a stack of dilated causal convolution layers with the convolution filter size of 1 × 2.

convolutions, the model is able to efficiently learn the connections between far-away time-steps in the time series data. Moreover, as opposed to causal convolutions in Fig. 1, the dilated causal convolutions require fewer layers even though the receptive field size is the same. A stack of dilated causal convolutions enables the network to have a very large receptive field with just a few layers, while preserving the computational efficiency. Therefore, dilated causal convolutions reduce the total number of learnable parameters, resulting in more efficient training and light-weight model.

However, the dilated causal convolutions have problem with local feature extraction. As shown in Fig. 2, it can be seen that the filter applied to the time-series input is not overlapped due to the skipping steps of the dilation factor. As long as the dilation factor increases, the feature is extracted from only far-apart time-steps, but not from adjacent time-steps. Therefore, the local connection among adjacent time-steps is not fully extracted at higher layers.

4) RESIDUAL BLOCK

The depth of the model is important for learning robust representations, but also comes with a challenge of vanishing
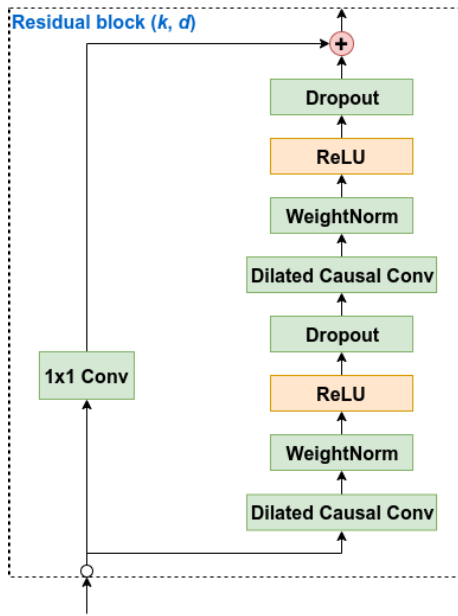
FIGURE 3. The residual block in TCN architecture.

FIGURE 4. The proposed CNN-QoE architecture.

gradients. The residual block has been found to be an effective way to address this issue and build very deep networks [24]. A residual block contains a series of transformation functions F, whose outputs are added to the input $\mathbf{x}$ of the block:

$$o = Activation(x + F(x)) \qquad (3)$$

The residual block is used between each layer in TCN to speed up convergence and enable the training of much deeper models. The residual block for TCN is shown in Fig. 3. It consists of dilated causal convolution, ReLU activation function [25], weight normalization [26], and spatial dropout [27] for regularization. Having two layers of dilated causal convolution in the TCN's residual block is suitable for complex challenges such as speech enhancement [8]. Compared with speech signal data, sequential QoE data is much simpler. That is to say, the two layers of dilated causal convolution are redundant and are not optimal for the QoE prediction problem.

In TCN architecture, equations (1) and (2) suggest that the TCN model heavily depends on the network depth L and the filter size k.

IV. PROPOSED QoE PREDICTION MODEL
In this section, the proposed model CNN-QoE is introduced to leverage the advantages and handles the problems of the TCN architecture [8] in QoE prediction tasks for video streaming services. The main objective of our study is to enhance the QoE prediction accuracy and minimize the computational complexity.

Let $\mathbf{x}_t$ be a vector of input features at a time instant t within a streaming session of T seconds.

Let y_t and $\widehat{y}_t$ be the subjective and the predicted QoE at a time instant t, respectively. In order to predict the subjective
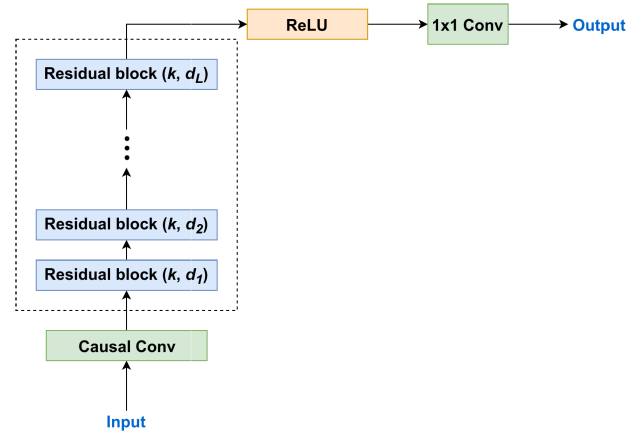
QoE continuously at any given time instant t, the following nonlinear function must be considered:

$$\widehat{y}_t = g(\mathbf{x}_t, \mathbf{x}_{t-1}, \ldots, \mathbf{x}_{t-r+1}), \quad 0 \le t \le T \qquad (4)$$

where r is the number of lags in the input. To learn the nonlinear function $g(\cdot)$, the CNN-QoE model is presented.

In the following subsections, the proposed architecture employed for the CNN-QoE model is discussed in detail. The model architecture hyperparameters are then analyzed to find the optimal values which can improve the QoE prediction accuracy, while minimizing the computational complexity.

A. PROPOSED MODEL ARCHITECTURE
Fig. 4 illustrates the overview of the CNN-QoE's architecture. The CNN-QoE leverages the advantages of 1D convolutions, dilated causal convolutions and residual block in TCN architecture. To adapt the TCN to QoE prediction tasks, a number of improvements are made as follows:

- An initial causal convolution layer is added to the input and then connects to residual block which includes a dilated causal convolution layer.
- The residual block is simplified by leveraging the advantages of Scaled Exponential Linear Units (SeLU) activation function [28].

These distinguishing characteristics are discussed as below.

1) CAUSAL CONVOLUTION TO EXTRACT LOCAL FEATURES
The architecture of the proposed model comprises of one causal convolution layer and a stack of dilated causal convolutions, while the TCN consists of only a number of dilated causal convolutions. A causal convolution layer is added between the input time-series and the first residual block as shown in Fig. 4. This causal convolution layer can extract the local features of the adjacent time-steps in the sequential QoE data. Afterward, the following dilated causal convolution layers are leveraged to extract the global features between far-apart time steps. These layers help the model to learn the most informative features in the time series input, resulting in higher accuracy.
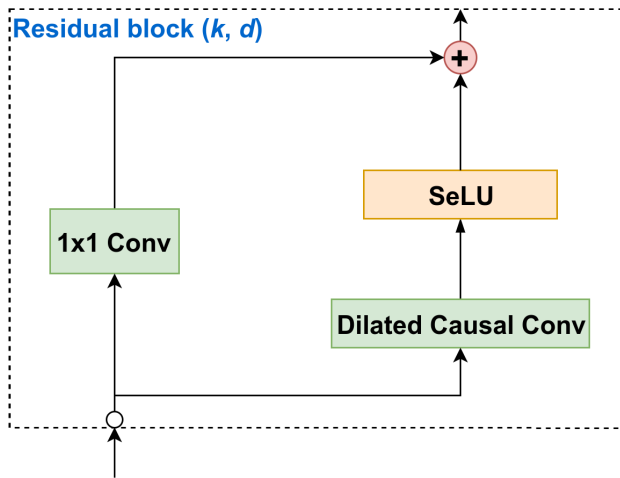
FIGURE 5. The proposed residual block used in the proposed architecture.

2) SeLU ACTIVATION FUNCTION

Activation function plays an important role in allowing the model to learn non-linear representations of the input features. When training a deep learning model, the vanishing and exploding gradient are the most challenging problems that prevent the network from learning the optimal function $g(\cdot)$. The TCN model gets rid of these problems by integrating ReLU activation function [25], weight normalization [26] and dropout [27] layer as shown in Fig. 3. In the proposed CNN-QoE, those layers are replaced with the SeLU to leverage its advantages and simplify the residual block as shown in Fig. 5. SeLU is a self-normalizing activation function. It converges to zero mean and unit variance when propagated through multiple layers during network training, thereby making it unaffected by vanishing and exploding gradient problems. Moreover, SeLU also solves the "dying ReLU" problem where the ReLU function always outputs the same value of 0 for any input, so the gradient descent is not able to alter the learnable parameters. At the same time, SeLU also reduces the training time and learns robust features more efficiently than other networks with normalization techniques, such as weight normalization [28]. SeLU activation function described as follow [28]:

$$SeLU(x) = \lambda \begin{pmatrix} x, & if \ x > 0 \\ \alpha exp(x) - \alpha, & if \ x \leq 0 \end{pmatrix} \quad (5)$$

where $\alpha = 1.67733$ and $\lambda = 1.0507$. These are the same values as the ones proposed in [28].

B. ARCHITECTURE HYPERPARAMETERS SELECTION

When training the model, an adequate set of architecture hyperparameters must be selected to achieve the best performance. The proposed model consists of L residual block layers, each layer contains a dilated causal convolution with a filter size of $1 \times k$, as shown in Fig. 4 and 5. Each dilated convolution layer has a dilation factor d doubled at each layer up, as shown in (2). The proposed model depends on the

TABLE 1. Hyperparameters for the best performance model.

Architecture Hyperparameters	Description	Derived Value
r	Receptive field size	8
k	Filter size	2
L	Number of dilated causal convolution layers	3
n	Number of filters on each convolution layer	32

network depth L and the filter size k. These hyperparameters control the trade-off between QoE prediction accuracy and computational complexity of the model. To effectively optimize the hyperparameters, it is important to set a boundary for the space of possible hyperparameter values.

The user's QoE is mostly affected by the recent experiences, also known as the recency effect [6], [29], [30]. The recency effect gradually decreases within 15 to 20 seconds [6], [30] after distorted events (e.g., bitrate fluctuations, rebuffering events). Therefore, the effective history or the receptive field size r of the model cannot be larger than 20 time-steps

$$r \leq 20 \quad (6)$$

Moreover, the receptive field depends on the number of dilated causal convolution layers L and the filter size k. For example, with $l \in [1, L]$, the receptive field r can be determined by (7) [8], [31]

$$r = 2^L, \quad if \ k = 2 \quad (7)$$

or (8) [32]

$$r = 2^{L+1} - 1, \quad if \ k = 3 \quad (8)$$

Fig. 2 shows an example of a three-layer ($L = 3$) dilated convolutional network. In this figure, given the filter size of 1×2 ($k = 2$), the receptive field is computed by $r = 2^3 = 8$. From (6), (7), and (8), the range of L values can easily be defined $L \in [2, 3, 4]$.

In a 1D convolution, the number of filters n is also important to effectively extract the information from the inputs. To minimize the computation complexity of the model, the range of n is set to $n \in \{16, 32, 64\}$. We conduct a simple grid-search of the model architecture hyperparameters with $k \in [2, 3]$, $L \in [2, 3, 4]$, and $n \in \{16, 32, 64\}$. Table 1 shows the values of r, k, L, and n that achieves the best performance.

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the CNN-QoE in terms of QoE prediction accuracy and computational complexity. The evaluation is performed by comparing the proposed model with numerous baseline models across multiple databases on both personal computers and mobile devices. In the following subsections, firstly, the employed four input features for QoE prediction are described, followed by a brief explanation of baseline models. Then, the evaluation results on accuracy and computational complexity are presented.

TABLE 2. An overview of the three public QoE databases used in the proposed model evaluation.

Database	Device Type	Rebuffering Events	Bitrate Fluctuations	Duration	QoE Range
LFOVIA Video QoE Database	TV	yes	yes	120 seconds	[0, 100]
LIVE Mobile Stall Video Database II	Mobile	yes	no	29-134 secs	[0, 100]
LIVE Netflix Video QoE Database	Mobile	yes	yes	at least 1 minute	[-2.28, 1.53]

Finally, the overall performance of the proposed model is discussed to illustrate its capability for real-time QoE prediction.

A. INPUT FEATURES FOR QoE PREDICTION

Video streaming users are sensitively affected by the video quality, known as *short time subjective quality* (STSQ) [4]. STSQ is defined as the visual quality of video being rendered to the user and can be predicted using any of the robust video quality assessment (VQA) metrics, such as Spatio-Temporal Reduced Reference Entropic Differences (STRRED) [33], Multi-Scale Structural Similarity (MS-SSIM) [34], Peak Signal to Noise Ratio (PSNR) [35], etc. Recent experiments have demonstrated that STRRED is a robust and high-performing VQA model when being tested on a very wide spectrum of video quality datasets, on multiple resolution and device types [7], [16], [36]. Therefore, in this paper, STRRED is utilized to measure the STSQ.

Rebuffering greatly impacts the user's QoE [37]. Therefore, rebuffering information such as rebuffering length, rebuffering position and the number of rebuffering events must be investigated. As a result, two rebuffering-related inputs are employed in this paper. Firstly, *playback indicator* (PI) [7], [14], [16] is defined as a binary continuous-time variable, specifying the current playback status, i.e., 1 for rebuffering and 0 for normal playback. Secondly, as the user's annoyance increases whenever a rebuffering event occurs [37], the *number of rebuffering events* (NR) happened from the start to the current time instant of the session is considered.

Besides, the user's QoE is also affected by memory factors. For example, more recent experiences have larger impacts on the user's perceived video quality, known as the recency effect [6], [29], [30]. To capture the relation between the recency effect and the user's QoE, *time elapsed since the last video impairment* (i.e., bitrate switch or rebuffering occurrence) [7], [14], [16], denoted as TR, is utilized.

B. BASELINE MODELS

To evaluate the QoE prediction accuracy of the proposed model on personal computers, the comparison with the state-of-the-art QoE models comprising of LSTM-QoE [7], NLSS-QoE [16], SVR-QoE [30], and NARX [14] will be performed. It is worth noting that we also make a comparison with the original TCN model, or TCN-QoE for short, in the QoE prediction task. The TCN-QoE model uses the same network hyperparameters and input features with ones described in Section IV-B and V-A.

To evaluate the QoE prediction accuracy and computational complexity of the proposed model on mobile devices,

we focus on the comparison with deep learning-based QoE prediction models since they achieve exceptionally higher accuracy. Particularly, LSTM-QoE [7] and TCN-QoE are utilized in the comparison. It is important to note that the LSTM-QoE [7] model hyperparameters are employed as reported in its respective works in order to ensure a fair comparison.

C. ACCURACY

1) DATABASES

There are three public QoE databases used for the evaluation of QoE prediction accuracy, including LFOVIA Video QoE Database [30], LIVE Netflix Video QoE Database [6], and LIVE Mobile Stall Video Database II [37]. The descriptions of these databases are summarized in Table 2.

To evaluate the QoE prediction accuracy on personal computers, the evaluation procedures performed on each database are described as follows:

- LFOVIA Video QoE Database [30] consists of 36 distorted video sequences of 120 seconds duration. The training and testing procedures are performed on this database in the same way as the one described in [7]. The databases are divided into different train-test sets. In each train-test sets, there is only one video in the testing set, whereas the training set includes the videos that do not have the same content and playout pattern as the test video. Thus, there are 36 train-test sets, and 25 of 36 videos are chosen for training the model for each test video.
- LIVE Netflix Video QoE Database [6]: The same evaluation procedure as described for LFOVIA Video QoE Database is employed. There are 112 train-test sets corresponding to each of the videos in this database. In each train-test set, the training set consists of 91 videos out of a total of 112 videos in the database (excludes 14 with the same playout pattern and 7 with the same content).
- LIVE Mobile Stall Video Database II [37]: The evaluation procedure is slightly different from the one applied to the above databases. Firstly, 174 test sets corresponding to each of 174 videos in the database are created. For each test set, since the distortion patterns are randomly distributed across the videos, randomly 80% videos from the remaining 173 videos are then chosen for training the model and perform evaluation over the test video.

To evaluate the QoE prediction accuracy on mobile devices, for simplicity, only the LFOVIA Video QoE Database [30] is utilized to train and test the proposed model. In this experiment, the set of 36 distorted videos in the
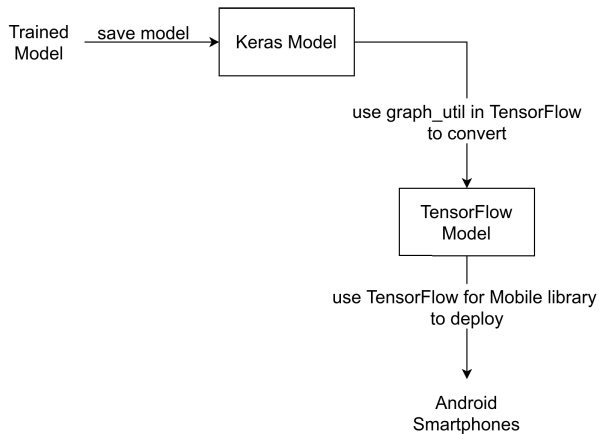
FIGURE 6. A diagram illustrating the steps to convert the CNN-QoE into a mobile device model for Android smartphones. The trained model is saved as a Keras model. The TensorFlow model is then extracted from the Keras model using the graph utility sub-module (graph_util) in TensorFlow [40]. The sub-module converts model variables to constants for inference only and removes all training related operations (e.g. gradients for backward propagation) to optimize the model on mobile devices. Finally, the TensorFlow model is deployed on Android smartphones using "TensorFlow for Mobile" library [40], [41].

database are divided into training and testing sets with a training:test ratio of 80:20. Thus, there are 28 videos in the training set and 8 videos in the testing set.

2) EVALUATION SETTINGS

To evaluate the accuracy of the proposed model, the model hyperparameter sets and input features are used as described in Section IV-B and V-A, respectively. The QoE prediction performance of the proposed model is first compared with baseline models described in Section V-B on personal computers. Then, on mobile devices, the comparison with other deep learning-based QoE prediction models (TCN-QoE and LSTM-QoE [7]) is focused on. To provide a fair comparison, the TCN-QoE and LSTM-QoE [7] models are also trained and tested on LFOVIA Video QoE Database with the training:test ratio of 80:20. The implementations of CNN-QoE, TCN-QoE, and LSTM-QoE [7] are based on Keras library [38] with the TensorFlow [39] backend.

In order to access the accuracy of the deep learning-based QoE prediction models on mobile devices (i.e., CNN-QoE, TCN-QoE, and LSTM-QoE [7]), Android smartphones are utilized for evaluation since the Android is the most popular mobile operating system[1] all over the world. To do so, after training phase, the trained models must be converted to the models that can be executed on the Android environment (as shown in Fig. 6).

3) EVALUATION CRITERIA

In this paper, three evaluation metrics, namely, Pearson Correlation Coefficient (PCC), Spearman Rank Order Correlation Coefficient (SROCC) and Root Mean Squared

---

[1] https://gs.statcounter.com/os-market-share/mobile/worldwide

TABLE 3. QoE prediction performance of the CNN-QoE over the LFOVIA Video QoE Database.

	PCC	SROCC	RMSE (%)
CNN-QoE	**0.820**	**0.759**	**4.81**
TCN-QoE	0.670	0.732	5.47
LSTM-QoE [7]	0.800	0.730	9.56
NLSS-QoE [16]	0.767	0.685	7.59
SVR-QoE [30]	0.686	0.648	10.44

Error (RMSE) are considered for QoE prediction accuracy assessment. The SROCC measures the monotonic relationship, while PCC measures the degree of linearity between the subjective and the predicted QoE For PCC and SROCC, a higher value illustrates a better result, while for the RMSE, the lower value is better.

4) RESULTS

Figs. 7, 8 and 9 illustrate the QoE prediction performance over the three databases using the proposed CNN-QoE model on personal computers. In general, the proposed model produces superior and consistent QoE prediction performance in different situations with and without rebuffering events. Patterns #1-#3 in Fig. 7, 8, and 9 show that the proposed model can effectively capture the effect of rebuffering events on the user's subjective QoE. Especially, even the rebuffering event repeatedly occurs as illustrated in pattern #3 in Fig. 7 and patterns #2, #3 in Fig. 8, the QoE predictions still correlate well with the subjective QoE. Meanwhile, pattern #0 in Fig. 7 and pattern #1 in Fig. 9 show some fluctuations in the predicted QoE. However, the amplitudes of these fluctuations are small and the varying trends in the subjective QoE are still adequately captured by the proposed model. Additionally, a linear trend is subsequently introduced in the predicted QoE after each rebuffering event as shown in patterns #1 - #3, Fig. 7. It means that the model is not overfitting and can be trained on the LFOVIA Video QoE Database with larger epochs to further increase its nonlinearity and QoE prediction accuracy.

The QoE prediction performance results over each database in comparison with existing models are shown in the Tables 3, 4 and 5. It is important to note that the Hammerstein-Wiener model in [18] was employed as reported in their work in order to ensure a fair comparison. From these tables, it is revealed that the CNN-QoE outperforms the existing QoE models within all the criteria, especially in terms of RMSE. Moreover, the accuracy produced by CNN-QoE is consistent across the databases, thus marking it as an efficient comprehensive model. The results illustrate that the CNN-QoE architecture is capable of capturing the complex inter-dependencies and non-linear relationships among QoE influence factors. Interestingly, there is a significant improvement in QoE prediction accuracy when comparing with TCN-QoE. It means that the enhancements in the proposed architecture have made the model more suitable for QoE prediction.

On mobile devices (i.e., Android smartphones), the QoE prediction accuracy of the proposed CNN-QoE is assessed
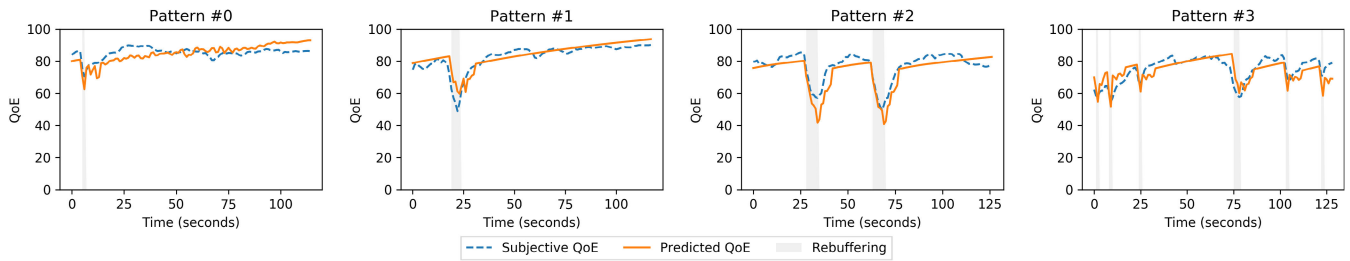
FIGURE 7. QoE prediction performance of the CNN-QoE over the LFOVIA Video QoE Database.
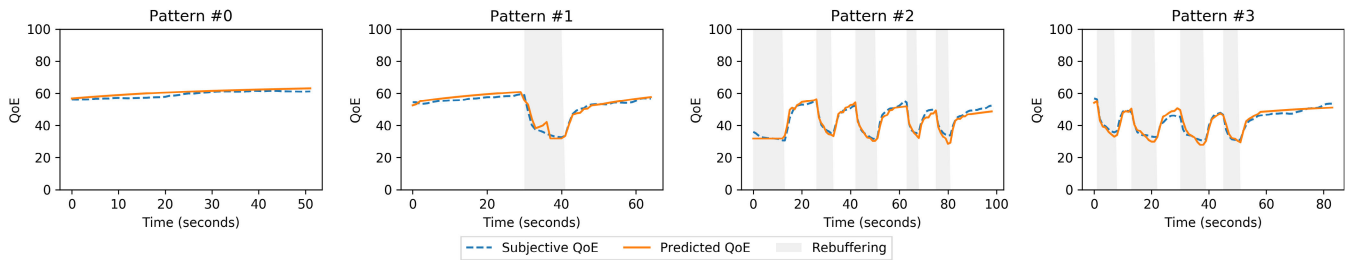

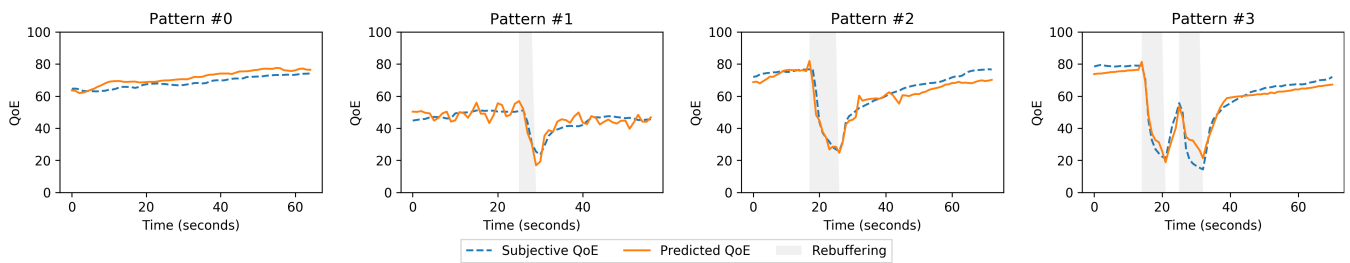**FIGURE 8.** QoE prediction performance of the CNN-QoE over the LIVE Mobile Stall II Video Database.

FIGURE 9. QoE prediction performance of the CNN-QoE over the LIVE Netflix Video QoE Database.

TABLE 4. QoE prediction performance of the CNN-QoE over the LIVE Mobile Stall Video Database II. Boldface indicates the best result.

	PCC	SROCC	RMSE (%)
CNN-QoE	**0.892**	**0.885**	**5.36**
TCN-QoE	0.667	0.603	9.71
LSTM-QoE [7]	0.878	0.862	7.08
NLSS-QoE [16]	0.680	0.590	9.52

TABLE 5. QoE prediction performance of the CNN-QoE over the LIVE Netflix Video QoE Database. Boldface indicates the best result.

	PCC	SROCC	RMSE (%)
CNN-QoE	**0.848**	**0.733**	**6.97**
TCN-QoE	0.753	0.720	7.62
LSTM-QoE [7]	0.802	0.714	7.78
NLSS-QoE [16]	0.655	0.483	16.09
NARX [14]	0.621	0.557	8.52
[18]	0.611	0.515	18.96

in comparison with TCN-QoE and LSTM-QoE [7]. The results are shown in Table 6. Accordingly, when performing on different platforms (personal computers and mobile devices), the QoE prediction accuracy of both CNN-QoE and TCN-QoE remains unchange. However, the LSTM-QoE, when performing on mobile devices,

TABLE 6. A comparison of the CNN-QoE's QoE prediction performance on personal computer and mobile device over LFOVIA Video QoE Database with 80/20 split.

	PCC		SROCC		RMSE (%)	
	PC	Mobile	PC	Mobile	PC	Mobile
CNN-QoE	**0.873**	**0.873**	**0.878**	**0.878**	**5.27**	**5.27**
TCN-QoE	0.731	0.731	0.792	0.792	6.71	6.71
LSTM-QoE [7]	0.863	0.259	0.822	0.650	6.82	22.57

suffers from a significant loss in the QoE prediction accuracy. This is because the TensorFlow fail to convert the recurrent connection in the LSTM architecture to mobile devices.

D. COMPUTATIONAL COMPLEXITY

In this subsection, the computational complexity of the proposed model on personal computers and mobile devices is investigated. The purpose is to show the effectiveness of the CNN-QoE on both high and low computational devices in comparison with baseline methods including TCN-QoE and LSTM-QoE [7]. These models are trained and tested on the LFOVIA Video QoE Database with a training:test ratio of 80:20.

TABLE 7. Computational complexity of the CNN-QoE on the personal computer.

	Inference Time (ms)	Model Size (kB)	FLOPs	Number of Parameters
CNN-QoE	**0.673**	82.08	175,498	9,605
TCN-QoE	0.856	145.86	253,076	13,766
LSTM-QoE [7]	1.996	**41.18**	**30,953**	**6,364**

TABLE 8. Computational complexity of the CNN-QoE on Android smartphones.

	Inference Time (ms)	Model Size (kB)
CNN-QoE	**0.581**	**56.222**
TCN-QoE	0.640	88.67
LSTM-QoE [7]	1.550	77.04

1) EVALUATION SETTINGS

For running these deep learning-based QoE prediction models, a personal computer running 18.04 Ubuntu LTS with an Intel i7-8750H @ 2.20GHz and 16GB RAM system is used. On the Android side, Sony Xperia XA2, which runs Android 9.0 and possesses a Qualcomm Snapdragon 630 64-bit ARM-based octa-core system on a chip, is used. Its CPU clock speed varies between 1.8-2.2 GHz depending on the core being used. The internal memory of this smartphone is 3GB LPDDR4 RAM. It should be noted that the GPU computation power is not utilized both on the personal computer and on Android smartphones.

2) EVALUATION CRITERIA

To conduct the evaluation on personal computers, the following four evaluation metrics are considered:

- Inference time: the time taken to predict the user QoE $\widehat{y}_t$ at any given time instant t.
- Model size: the storage size of the trained model on the hard drive.
- FLOPs: the number of operations performed.
- Number of Parameters: number of learnable parameters in the model.

On the Android smartphones, after the conversion from Keras to TensorFlow model as described in Section V-C2, the model is for inference only since all the learnable parameters were converted to constants. Therefore, the complexity of the CNN-QoE was compared with the others on Android smartphones using only two metrics: 1) Inference time and 2) Model size.

3) RESULTS

Table 7 and 8 show the computational complexity results of the proposed CNN-QoE compared to the TCN-QoE and LSTM-QoE. In general, the CNN-QoE requires a higher number of parameters and FLOPs in comparison with LSTM-QoE to achieve higher accuracy. Although the FLOPs of the proposed model are larger, the inference time is 3 times faster than the LSTM-QoE model both on the personal computer and the Android smartphone. This indicates that the proposed model can efficiently leverage the power of parallel computation to boost up the computing speed. It can be seen from Table 7 that the architecture complexity of TCN-QoE is extremely higher than our proposed CNN-QoE model in terms of number of parameters and FLOPs. However, the accuracy of TCN-QoE is not quite comparable with the CNN-QoE as shown in Table 6. It proves that the proposed improvement adapted on the original TCN architecture

allow CNN-QoE to effectively capture the complex temporal dependencies in a sequential QoE data.

E. OVERALL PERFORMANCE

Accurate and efficient QoE prediction models provide important benefits to the deployment and operation of video streaming services on different viewing devices. As shown in subsection V-C and V-D, the proposed model CNN-QoE can achieve not only the state-of-the-art QoE prediction accuracy but also the reduction on computational complexity. Therefore, the CNN-QoE can be an excellent choice for future QoE prediction systems or QoE-driven video streaming mobile applications.

VI. DISCUSSION

According to the above-mentioned evaluation results, it can be seen that the proposed model completely outperforms TCN-QoE where the original TCN architecture is adopted in the QoE prediction task. Thereby, it generally demonstrates the efficiency of the proposed improvements upon the original TCN architecture in QoE prediction for video streaming services. In the following subsections, the effects of the improvements including the interactions between causal convolutions and dilated causal convolutions are discussed in detail.

A. EFFECTS OF COMPRISING CAUSAL CONVOLUTIONS AND DILATED CAUSAL CONVOLUTIONS

Different from TCN [8] architecture, the proposed architecture has an initial causal convolution instead of a dilated causal convolution, as shown in Fig. 4. Unlike dilated causal convolution, a causal convolution with denser filters is more effective in extracting the local dependencies among adjacent time-steps. However, a stack of causal convolutions dramatically increases the model complexity. Therefore, we combine causal convolutions with dilated causal convolutions to achieve desirable prediction accuracy, while eliminating the complexity possibly caused by only utilizing causal convolutions in the architecture. As a result, the proposed model can effectively capture the temporal dependencies among adjacent and far-apart time-steps in the sequential QoE data, providing a better QoE prediction accuracy, especially in terms of RMSE.

Moreover, it can be seen from Tables 7 and 8 that the FLOPs of the proposed model are larger than those of LSTM-QoE. The reason is that the convolution layers require more operations for performing convolution between a number of filters and the input time series. However, the proposed model

runs faster than the baseline models on both personal computers and Android smartphones. This indicates that the convolution operations are fully parallelized, leading to real-time QoE prediction advantages.

B. EFFECTS OF SIMPLIFYING THE RESIDUAL BLOCK AND USING SeLU

To simplify the residual block, we adopt only one dilated causal convolution in the residual block instead of two as in the original TCN architecture (as illustrated in Fig. 3 and Fig. 5). The reason behind this is the fact that the sequential QoE data is much simpler than the preferred data of TCN [8] (i.e., speech signal data). Therefore, two dilated causal convolution layers can make the model easily suffers from overfitting and reduces the QoE prediction accuracy. Reducing the number of dilated causal convolutions in the residual block helps the proposed model to be easily trained and reduce overfitting. Furthermore, SeLU [28] activation function also enables the model to learn faster and converge better to the optimal values, subsequently improving the QoE prediction accuracy.

In terms of computational complexity, observing from Tables 7 and 8, it is obvious that these improvements in the residual block tremendously reduced the number of parameters compared to the one in the original TCN architecture TCN-QoE. Thereby, the CNN-QoE can produce smaller model size and FLOPs, faster training and inference times.

In summary, the improvements in the proposed architecture help provide a more stable, accurate and light-weight QoE prediction model.

VII. CONCLUSION

In this paper, the CNN-QoE model is proposed for continuous QoE prediction. The proposed model introduces multiple improvements to the original TCN model to leverage its strengths and eliminate its drawbacks in the QoE prediction task for video streaming services. The comprehensive evaluation across different QoE databases on both personal computers and mobile devices demonstrates that CNN-QoE produces superior performance in terms of QoE prediction accuracy and computational complexity. Accordingly, CNN-QoE provides a highly competitive prediction performance. These results validate the robustness of the proposed model in real-time QoE prediction on different platforms and devices.

REFERENCES

[1] C. M. Tran, T. N. Duc, P. X. Tan, and E. Kamioka, "QABR: A QoE-based approach to adaptive bitrate selection in video streaming services," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, pp. 138–144, Sep. 2019.

[2] J. Liu, X. Tao, and J. Lu, "QoE-oriented rate adaptation for dash with enhanced deep Q-learning," *IEEE Access*, vol. 7, pp. 8454–8469, Dec. 2018.

[3] 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.

[4] 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. [Online]. Available: http://ieeexplore.ieee.org/document/6775292/

[5] D. Ghadiyaram, J. Pan, and A. C. Bovik, "Learning a continuous-time streaming video QoE model," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2257–2271, May 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8247250/

[6] C. G. Bampis, Z. Li, A. K. Moorthy, I. Katsavounidis, A. Aaron, and A. C. Bovik, "Study of temporal effects on subjective video quality of experience," *IEEE Trans. Image Process.*, vol. 26, no. 11, pp. 5217–5231, Nov. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7987076/

[7] N. Eswara, S. Ashique, A. Panchbhai, S. Chakraborty, H. P. Sethuram, K. Kuchi, A. Kumar, and S. S. Channappayya, "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] S. Bai, J. Zico Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*. [Online]. Available: http://arxiv.org/abs/1803.01271

[9] P. Dutilleux, "An implementation of the 'algorithme à trous' to compute the wavelet transform," in *Wavelets*. Berlin, Germany: Springer, 1990, pp. 298–304.

[10] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*. [Online]. Available: http://arxiv.org/abs/1511.07122

[11] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[12] M. N. Garcia, W. Robitza, and A. Raake, "On the accuracy of short-term quality models for long-term quality prediction," in *Proc. 7th Int. Workshop Qual. Multimedia Exper. (QoMEX)*, May 2015, pp. 1–6.

[13] Y. Shen, Y. Liu, Q. Liu, and D. Yang, "A method of QoE evaluation for adaptive streaming based on bitrate distribution," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, Jun. 2014, pp. 551–556.

[14] C. G. Bampis, Z. Li, and A. C. Bovik, "Continuous prediction of streaming video QoE using dynamic networks," *IEEE Signal Process. Lett.*, vol. 24, no. 7, pp. 1083–1087, Jul. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7931662/

[15] C. G. Bampis and A. C. Bovik, "An augmented autoregressive approach to HTTP video stream quality prediction," 2017, *arXiv:1707.02709*. [Online]. Available: http://arxiv.org/abs/1707.02709

[16] N. Eswara, H. P. Sethuram, S. Chakraborty, K. Kuchi, A. Kumar, and S. S. Channappayya, "Modeling continuous video QoE evolution: A state space approach," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2018, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8486557/

[17] Y. Liu, S. Ci, H. Tang, Y. Ye, and J. Liu, "Qoe-oriented 3D video transcoding for mobile online streaming," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 8, no. 3s, Oct. 2012, Art. no. 42, doi: 10.1145/2348816.2348821.

[18] W. Shi, Y. Sun, and J. Pan, "Continuous prediction for quality of experience in wireless video streaming," *IEEE Access*, vol. 7, pp. 70343–70354, 2019.

[19] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia, "A survey on quality of experience of http adaptive streaming," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 469–492, 1st Quart., 2015. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6913491

[20] N. Barman and M. G. Martini, "QoE modeling for HTTP adaptive video streaming—A survey and open challenges," *IEEE Access*, vol. 7, pp. 30831–30859, 2019.

[21] R. K. P. Mok, E. W. W. Chan, and R. K. C. Chang, "Measuring the quality of experience of HTTP video streaming," in *Proc. 12th IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM) Workshops*, May 2011, pp. 485–492.

[22] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1135–1143.

[23] F. Tung and G. Mori, "CLIP-Q: Deep network compression learning by in-parallel pruning-quantization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7873–7882.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[25] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.

[26] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 901–909.

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[28] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 971–980.

[29] D. S. Hands and S. E. Avons, "Recency and duration neglect in subjective assessment of television picture quality," *Appl. Cognit. Psychol.*, vol. 15, no. 6, pp. 639–657, 2001.

[30] N. Eswara, K. Manasa, A. Kommineni, S. Chakraborty, H. P. Sethuram, K. Kuchi, A. Kumar, and S. S. Channappayya, "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. [Online]. Available: https://ieeexplore.ieee.org/document/8013810/

[31] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," 2016, *arXiv:1609.03499*. [Online]. Available: http://arxiv.org/abs/1609.03499

[32] Y. A. Farha and J. Gall, "MS-TCN: Multi-stage temporal convolutional network for action segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3575–3584.

[33] R. Soundararajan and A. C. Bovik, "Video quality assessment by reduced reference spatio-temporal entropic differencing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 4, pp. 684–694, Apr. 2013.

[34] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, vol. 2, 2003, pp. 1398–1402.

[35] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack, "Study of subjective and objective quality assessment of video," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1427–1441, Jun. 2010.

[36] 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, doi: 10.1016/j.image.2018.05.017.

[37] 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. [Online]. Available: http://ieeexplore.ieee.org/document/8093636/

[38] F. Chollet. (2015). *Keras*. [Online]. Available: https://keras.io

[39] M. Abadi. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: http://tensorflow.org/

[40] A. Sehgal and N. Kehtarnavaz, "Guidelines and benchmarks for deployment of deep learning models on smartphones as real-time apps," *Mach. Learn. Knowl. Extraction*, vol. 1, no. 1, pp. 450–465, Feb. 2019.

[41] A. Ignatov, R. Timofte, W. Chou, K. Wang, M. Wu, T. Hartley, and L. Van Gool, "Ai benchmark: Running deep neural networks on Android smartphones," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018.

CHANH TRAN MINH received the B.E. degree in computer engineering from the Ho Chi Minh City University of Technology-Vietnam National University, Vietnam, in 2018. He is currently pursuing the master's degree with the Shibaura Institute of Technology, Japan. His research interests include the quality of experience (QoE), HTTP adaptive video streaming, fog/edge computing, and the Internet of Things. He has been a recipient of the Innovative Asia Scholarship given by the Japan International Cooperation Agency (JICA), since 2018.

TAN PHAN XUAN received the B.E. degree in electrical and electronics engineering from Military Technical Academy, in 2009, the M.S. degree in computer and communication engineering from the Hanoi University of Science and Technology (HUST), in 2013, and the Ph.D. degree in functional control systems from the Shibaura Institute of Technology (SIT), in 2018. From 2011 to 2014, he was a Network Administrator and the Head of the Department of E-Learning, Network Information Center (BKNIC), HUST. He joined the SIT as a doctoral course student, in 2015. Since October 2018, he has been an Assistant Professor of SIT Research Laboratories, SIT. His research interests are computer networks, bioinformatics, the quality of experience, and image and video processing.

THO NGUYEN DUC received the B.E. degree in computer science from the Ho Chi Minh City University of Technology-Vietnam National University, Vietnam. He is currently pursuing the master's degree with the Shibaura Institute of Technology, Japan. His current research interests include the quality of experience (QoE) and image processing. He has been a recipient of the Innovative Asia Scholarship given by the Japan International Cooperation Agency (JICA), since 2018.

EIJI KAMIOKA (Member, IEEE) received the B.S., M.S., and D.S. degrees in physics from Aoyama Gakuin University. He is a Professor with the Shibaura Institute of Technology (SIT). Before joining the SIT, he was working with the SHARP Communication Laboratory, Institute of Space and Astronautical Science (ISAS), as a JPSP Research Fellow, and the National Institute of Informatics (NII) as an Assistant Professor. His current research interests encompass mobile multimedia communications and ubiquitous computing.

• • •