International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys.13, No.1 (May-23)

http://dx.doi.org/10.12785/ijcds/1301119

An Efficient Stacked Deep Incremental Model for Online Streaming Video QoE Prediction

Radhia Elwerghemmi[1], Maher Heni[1], Riadh Ksantini[2] and Ridha Bouallegue[1]

[1]*Innov'COM Laboratory, Higher School of Communication (Sup'com), Ariana, Tunisia*
[2]*Department of Computer Science, College of IT, University of Bahrain, Kingdom of Bahrain*

Received 13 Feb. 2023, Revised 5 Apr. 2023, Accepted 23 Apr. 2023, Published 30 May. 2023

Abstract: The Quality of Experience (QoE) metric is used as a direct evaluation of customers' experiences in video streaming diffusion, which is very important for network management, especially for the optimization and the improvement of the network. Hence, it is important to continuously quantify the perceived QoE of streaming video clients to minimize the QoE degradation. Nevertheless, the continuous evaluation of the perceived quality is challenging since it is defined by complex dynamic interactions between the QoE influencing factors. Thus, in this work, a new Deep Incremental Support Vector Machine (ISVM) QoE assessment model is developed that integrates deep learning techniques and a multiclass ISVM. The deep learning layer is employed to extract deep features which have discriminative power and lead to performance improvement. ISVM algorithm aims to manage non-stationary and massive amounts of data in real-time scenarios. Experiments are carried out on a real-world public datasets. The results demonstrate that our approach outperforms state-of-the-art approaches for evaluating QoE.

Keywords: Quality of experience, Deep Learning, Online Learning, Incremental Support Vector Machine, Video Streaming service.

1. Introduction

In recent years, Video streaming is becoming a popular Internet usage scenario, accounting for more than 80% of the Internet traffic [1]. Controlling the video quality of experience (QoE), which is employed as a real evaluation of clients' experiences in mobile video diffusion, gives application providers such as YouTube and Netflix, a deep insight into the quality of their network services for video transmission. Specifically, real-time assessment of video QoE allows network providers to dynamically improve their network capacity provisioning and traffic routing techniques [2].

In literature, many researchers have studied the video streaming QoE [3] [4] [5]. Yet, measuring, modeling, and predicting video streaming QoE are still challenging tasks. Video streaming QoE is affected by multiple intertwined factors, also known as Influence Factors (IFs), and they can be divided into three groups: system IFs, context IFs, and human IFs [6]. Traditional IF modeling approaches concentrate on system parameters including the Peak Signal-to-Noise Ratio (PSNR) [7]. These approaches have increasingly been supplanted by methods that rely on context IFs [8] and human IFs [9] since they are unable to adequately evaluate human perceptual involvement.

To quantify the two categories of IFs mentioned above in conditions where human cognition is not fully understood, there are two major categories for the prediction of the QoE: subjective methods and objective methods. Subjective models were offered as ways for directly measuring customer QoE by requesting evaluation scores from clients in a completely controlled environment [10]. Objective models are based on comparing methodologies to predict the perceived quality and provide an interpretable score. Subjective tests are used to confirm objective results.

Despite their high performance, these models have several downsides and limits. For example, the subjective tests are time and money consuming, making real-time QoE measurement extremely challenging. Moreover, These techniques rely mainly on hand-crafted characteristics and data representations that are specific to a database. As a result, they are difficult to apply directly to different contexts, thus these methods frequently do not correspond to human perception.

To fill this gap, this paper proposes an incremental deep learning-based model, namely, Deep Incremental Support Vector Machine (Deep ISVM) model, employed for video streaming QoE prediction. Specifically, we have combined two models.

First, we have a DeepQoE model used for the feature preprocessing and the representation learning [11], which could provide generalized features with a unified representation that is unaffected by datasets of heterogeneous modalities,

E-mail address: radhia.werghemmi@supcom.tn, maher.heni@supcom.tn, rksantini@uob.edu.bh, ridha.bouallegue@supcom.tn

using deep learning techniques.
Second, we have an Incremental Support Vector Machine (ISVM) employed for the real-time QoE assessment. This model is constructed online using real-time customer feedback, thus there is no need for a dataset collected from subjective studies to train our model. Moreover, it is unnecessary to select particular users, as the services own users are taken into account. Since service circumstances change from one stream to another, the ISVM learns more and becomes more complete, and its accuracy increases. Furthermore, the proposed incremental model can adapt to changes in the customer preferences and the introduction of new environmental circumstances, for example, novel content and novel terminal devices.
The rest of the paper is organized as follows: Section II includes related works. Section III presents the Deep ISVM method in detail. Section IV describes the experimental results. Section V concludes and discusses the future direction of this work.

2. Related works

The main existing research works dealing with the prediction of video streaming QoE and based on machine/deep learning techniques, can be categorized into two types: batch learning based models and incremental learning based models. We briefly discuss these two categories in this section.

A. Batch learning based models

In literature, many researchers have employed ML methods for the perception of users' QoE. For instance, the authors of [12] present a survey of ML strategies employed in the automatic identification of relation between Quality of Service (QoS) metrics and QoE values, such as random forest model (RF), SVM model, naïve Bayes model, and K-nearest neighbors model. Moreover, the study of [13] proposes a video streaming QoE assessment model using various regression methods, including ridge and lasso regression, as well as ensemble approaches, including RF, gradient boosting (GB), and extra trees (ET). Also, authors in [14] introduced a framework for predicting video QoE using Optimized Learning Models based on Multi-Feature Fusion (MFF) (OLMs). The OLMs are an optimized neural network algorithms built to estimate user experience. Besides, a transfer learning-based ML model is presented in the work of [15] for the video QoE estimation, which stacks the predictions of a generic pre-trained model with a specific trained model, to enhance the global accuracy. Authors have used the XGBoost (XGB) and Neural Network (NN) methods at both the source and target domains.
Despite the potential for applying traditional ML techniques in the perception of user QoE, these models require manual intervention for expert feature engineering, which is costly in terms of time, and they are almost not reusable because wireless network conditions change rapidly.
To overcome these problems, recently, Deep Learning (DL) approaches have been widely used by researchers for building autonomous models for the estimation of the QoE. For example, authors in [16] present a video streaming QoE estimation method, employing the Long Short Term Memory (LSTM) approach. Also, the study of [17] builds a hybrid deep learning model for medical video QoE estimation, based on the combination of the Long Short Term Memory (LSTM) technique and the boosting ensemble learning method. Furthermore, authors in [18] developed a framework made up of a Convolutional Neural Network (CNN) and an LSTM networks. In [11], a DeepQoE framework for user video streaming assessment is presented. This framework is built on a mix of DL approaches including word embedding, 3D CNN and representation learning.
All the models stated above are batch learning models. These methods are learned using expensive subjective studies, which are composed of static, uniformly distributed, and labeled training examples. Yet, dynamic changes in the external environment of the real world require models that are capable of continuous learn and memorize. Additionally, batch learning techniques build new models from scratch rather than continuously incorporating new data into previously trained models. This not only wastes time but also results in out-of-date patterns.

B. Incremental learning based models

The incremental learning technique has multiple advantages in building an accurate QoE model, which can be summed up in the following key points:

- Efficiency in time complexity: processing one sample at a time is significantly more efficient and more practical than batch learning algorithms. These latters rely on iterative optimization techniques. Therefore, they perform the same computation over all the training set for many iterations.
- Efficiency in space memory: Online learning improves space memory efficiency by updating their hypothesis about the unknown rule based solely on the old hypothesis and recent examples. As a result, storing the whole collection of examples is avoided.
- Handling large amounts of data: by learning recent knowledge of the data and avoiding storage of all trained samples, this help online learning to handle large amounts of data, as opposed to batch learning algorithms which are forced to delete trained models to learn new ones.
- Provide real-time results: online learning provides real-time results, which aim to react instantaneously to optimize parameters as soon as possible. This important feature of incremental learning is becoming a key area of data mining research since various applications demand such processing.
- Solve complex problems: online learning algorithms minimize the worst-case mistake bound while batch algorithms generally minimize the loss of training samples.

- Solve Single Sample Size (SSS) problem: sample size tends to be the dominant limitation of batch learning. Although some amount of information is available in online learning, This latter can achieve comparable performance to the more complex optimal batch methods.

For video streaming QoE assessment, only few researchers have adopted incremental learning techniques. We will mention two of them.
In the study of [19], an online QoE prediction method based on Hoeffding Trees is presented. Four variants of this approach are employed, namely, Standard Hoeffding Trees (HT), HT with Naive Bayes (NB), HT with adaptive NB and Hoeffding Option Trees with NB and adaptive approach (HOTNBAdaptive). This method produces good performance in terms of accuracy and strong flexibility to concept drift in the database. Yet, the proposed model is based on a decision tree model, thus its performance decreases when handling large and complex datasets.
The authors in [20] proposed a stacked multiclass incremental support vector machine in the online prediction of QoE for video streaming services. The experimental results demonstrated that this method has good performance. Although this model has high performance, there remains the problem of human intervention in feature extraction and preprocessing.
Thus, in this paper, we develop a new deep incremental method for video streaming QoE prediction. The deep learning model is used to minimize as far as possible handcrafted features and dataset-specific representation, and the ISVM model, as an incremental model, is employed for handling streams of data. In fact, many other learning techniques have been reviewed and changed into incremental methods, which can learn over time. The ISVM-based methods are said to have a number of desirable properties, making them an interesting tool for dealing with incrementally acquired data. For example, when we employ ISVM-based methods, the training phase scales using a few numbers of support vectors rather than the complete data examples. Besides, authors in [21] prove that the ISVM model gives the best accuracy compared to the rest incremental models, such as the Online Random Forest (ORF), Learn++ and Incremental Learning Vector Quantization (ILVQ), etc...

3. Proposed Deep ISVM model for QoE assessment

In this section, we present our proposed Deep Incremental Support Vector Machine (Deep ISVM) model used for video streaming QoE assessment. For that reason, first, we describe the feature preprocessing and representation learning steps using a deep learning model. After that, we provide a detailed explication of the multi class incremental SVM model used for the online QoE estimation. Finally, the suggested deep incremental QoE prediction approach is described using a flowchart.

A. *DeepQoE model*

The Deep QoE framework is presented mainly for avoiding the over-reliance on particular dataset extracting features and for managing heterogeneous QoE IFs in terms of data types, modality, and representation. As is mentioned in figure 1, the proposed framework is composed of three parts.
In the first phase, convolutional neural network (CNN) models are used to extract features from various datasets. More specifically, the input data are divided based on its category as follows:

- Videos are converted into vectors using a Convolutional 3D (C3D) model [22], which is pre-trained with Sport-1M data.
- Textual data are interpreted using a Global Vectors (GloVe) model [23], pre-trained with Wikipedia database.
- Categorical data are handled using an embedding layer.
- Continuous values are processed using a dense layer.

The Deep Neural Network (DNN) model is used in the second phase to provide a representation for the DeepQoE framework as inputs that is independent of specific feature types or databases. Therefore, the outputs of the first phase are flattened to create a single 1-dimensional feature vector. After that, this vector will be connected to successive fully connected layers. Furthermore, the dropout approach [24] is used on fully connected layers to avoid overfitting.
In the last phase, the outputs of the DNN model are sent to a MLP network, which can be employed for classification or regression operations as follow:
For classification, the decision function is given by the equation below:

$$Y_{pred} = F_s(Wx + b). \quad (1)$$

With F_s denoting the softmax activation function, Y_{pred} is the predicted class, The weight matrix is denoted by W, and the bias is denoted by b. As a loss function, a cross-entropy function is employed.
For regression, the final QoE values are given by the equation below:

$$Y_{pred} = F_l(Wx + b). \quad (2)$$

Where Fl represents the linear activation function, $Ypred$ represents the predicted QoE value, W represents the weight matrix, and b represents the bias. A Mean Squared Error (MSE) loss function is used.
In our study, we have employed this model only for feature preprocessing and representation learning. The QoE prediction will be performed by the ISVM model.

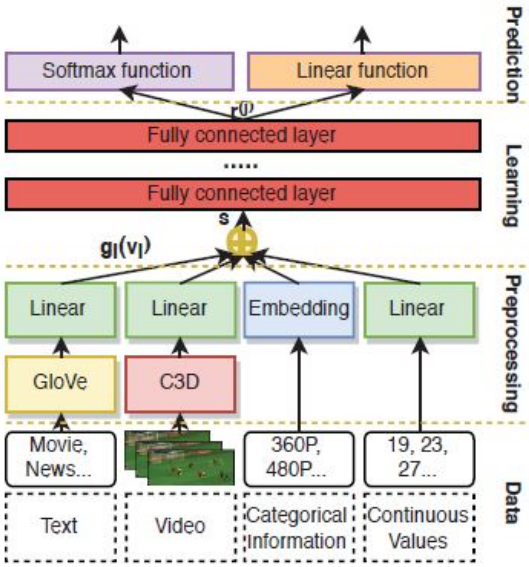


Figure 1. DeepQoE framework architecture [11]

B. Incremental Classification

In this part, we begin by presenting the classical batch SVM technique, since it is the basis of the employed incremental SVM model. The ISVM method is then thoroughly described using mathematical formulations and schematic depictions, which will be used for video streaming QoE perception.

1) Batch SVM model

Let $X = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$ be the training dataset of N examples, where $x_i = \{QoE\_IFs\} \in R^k (k \geq 1), i = 1, ..., N$ are the QoE IFs, $y_i = QoEscores \in R \in \{1, 2, 3, 4, 5\}$ and k denoted the input feature vectors size. Batch SVM is a binary supervised model that aims to build an optimal hyper-plane that best separates the tags by maximizing the margins from both tags [25]. To construct this hyper-plane, the quadratic problem shown below should be resolved.

$$\underset{0 < \alpha_i < C}{min} : W = \frac{1}{2} \sum_{i,j} \alpha_i Q_{ij} \alpha_j - \sum_i \alpha_i + b \sum_i y_i \alpha_i. \quad (3)$$

With α_i denoting the Lagrange multipliers, b denotes the offset, $Q_{ij} = y_i y_j K(x_i, x_j), K(x_i, x_j)$ is the kernel function and C is a parameter which controls the loss function.
After resolving this problem, the classification of any test point can be determined by the following equation:

$$f(x) = \sum_{i=1}^{N} y_i \alpha_i K(x_i, x) + b. \quad (4)$$

Yet, in a real-time context, the batch SVM shows several challenges as well as significant performance loss, where the whole dataset is not available at the beginning of the learning process, as data are gathered sequentially. Furthermore, this approach necessitates a considerable amount of storage space and increases training time, particularly for big datasets. Therefore, an incremental learning technique is crucial.

2) The Incremental SVM model

Our purpose is to convert the classical batch SVM model into an incremental one. Thus, when novel data is introduced, we can add it to an existing optimal solution that retains the previous knowledge, without the need of retraining the whole dataset [26]. More precisely, the Lagrange multipliers must be adjusted, while conserving the Karush-Kuhn-Tucker (KKT) conditions on all already collected examples.
1) KKT condition:
The saddle point of the problem given by Eq. 1 is provided by the KKT conditions as:

$$g_i = \frac{\partial W}{\partial \alpha_i} = \sum_j Q_{ij} \alpha_j + y_i b - 1. \quad (5)$$

$$\frac{\partial W}{\partial b} = \sum_j y_j \alpha_j = 0. \quad (6)$$

The KKT conditions divide the database (D) into three subsets:

- The subset S represents support vectors ($g_i = 0, 0 < \alpha_i < C$).
- The subset E denotes error vectors ($g_i < 0, \alpha_i = C$).
- The subset R denotes non-support vectors ($g_i > 0, \alpha_i = 0$).

2) Adiabatic increments:
To keep the KKT conditions in equilibrium, we can represent them differently in the following equations.

$$\Delta g_i = Q_{ic} \Delta \alpha_c + \sum_{j \in S} Q_{ij} \Delta \alpha_j + y_i \Delta b \qquad \forall i \in D \cup \{c\}. \quad (7)$$

$$0 = y_c \Delta \alpha_c + \sum_{j \in S} y_j \Delta \alpha_j. \quad (8)$$

With α_c denoting the coefficient that will be incremented. The following are the equations:

$$Q. \begin{bmatrix} \Delta b \\ \Delta \alpha_S \end{bmatrix} = - \begin{bmatrix} y_c \\ Q_{S,c} \end{bmatrix} \Delta \alpha_c. \quad (9)$$

$$With \quad Q = \begin{bmatrix} 0 & y_S^T \\ y_S & Q_S \end{bmatrix}.$$

Where $\Delta \alpha_S$ represents a vector holding the corresponding $\Delta \alpha_i : i \in S(\alpha)$, Q_S denotes a kernel matrix holding S_s and $Q_{S,c}$ represents a kernels vector between S_s and x_c.
As a result, in equilibrium

$$\Delta_b = \beta \Delta \alpha_c. \quad (10)$$

$$\Delta\alpha_j = \beta_j \Delta\alpha_c \qquad \forall j \in D. \tag{11}$$

β coefficients are computed as follows:

$$\begin{bmatrix} \beta \\ \beta_S \end{bmatrix} = -R. \begin{bmatrix} y_c \\ Q_{S,c} \end{bmatrix}. \tag{12}$$

In which $R = Q^{-1}$ and $\beta_j \equiv 0 \ \ \forall j \notin S$.

$$\Delta g_i = \gamma_i \Delta\alpha_c \ \ \forall i \in D \cup \{c\}. \tag{13}$$

Where
$\gamma_i = Q_{ic} + \sum_{j \in S} Q_{ij}\beta_j + y_i\beta, \gamma_i = 0 \ \ , \forall i \in S.$

3) The resulting updates:
For the ISVM approach, a novel input x_c should be inserted in one of the three subsets mentioned above according to the values of g_c and α_c. The Woodbury Formula states that if a variable xc is classified as in support vector subcategory, the R matrix expands to:

$$R \leftarrow \begin{bmatrix} R & 0 \\ 0 & 0 \end{bmatrix} + \frac{1}{\gamma_c} \begin{bmatrix} \beta \\ \beta_S \\ 1 \end{bmatrix} . \begin{bmatrix} \beta \, \beta_S \ 1 \end{bmatrix}. \tag{14}$$

Using the same formula as x_k leaves S, the R matrix is contracted recursively as follows:

$$R_{ij} \leftarrow R_{ij} - R_{kk}^{-1} R_{ik} R_{kj} \qquad \forall i, j \in S \, ; i, j \neq k. \tag{15}$$

Thanks to the two previous formulas, the complexity of the incremental SVM method is converted from $O(n^3)$ to $O(ns^2)$, in which n represents the number of training data points and ns is the number of support vectors.
To recapitulate, Algorithm 1 contains the ISVM model's pseudo-code.
We keep moving parameters consecutively until the KKT conditions are satisfied, and we reach equilibrium. The main purpose is to achieve the highest feasible increase α_c, while preserving the subsets' decomposition. During the updating operation, we must consider the migration of certain components from one subset to another. This is referred to as adiabatic increments [26].

Algorithm 1 Incremental SVM model algorithm: high-level summary.

```
Read example x_c, y_c
Calculate R, and employ it to find β and γ using Eqs. (8)-(11)
Set α_c and Δα_c = 0
Compute g_c using Eq. (3)
while g_c < 0 and α_c < C do
    if g_c = 0 then
        Add x_c to S and equilibrium is reached
        Set α_c = Δα_c
        Update (α_i)_{i=1...n}
        Update R according to (12)
    end if
    if g_c < 0 then
        Add x_c to E and equilibrium has been attained
        Set α_c = c
    end if
    Update the subsets S, E, and R
    Update R recursively according to Eqs. (12)-(13).
end while
```

C. Deep Incremental SVM model algorithm

Our proposed Deep ISVM model is the outcome of a combination of two powerful models as is shown in the following flowchart presented in Figure 2:
In the training process, the output of the DNN model will serve as input for the incremental SVM model. Thus, the ISVM will be trained with DeepQoE features instead of original features. As we classify data into five classes, following the protocol of ACR recommendation in ITU-T P.910 [27], we have extended the binary ISVM to a multiclass model using the One-against-all method. As a result, 5 binary one-versus-all ISVM classifiers are built. The output class is that of the classifier with the highest output value before thresholding.

4. EXPERIMENTAL RESULTS

For highlighting the prominence of the suggested Deep ISVM model for video streaming QoE prediction in real-time contexts, several experiments are carried out. The employed datasets are introduced first. Then, the experimental protocol is presented in detail. Finally, we conduct an analysis of the obtained results.

A. Dataset used

To evaluate the Deep ISVM model we have employed three publicly available datasets. Their specifics are detailed as follows:
Poqemon Database [1]: This dataset contains 300 samples covering 7 QoE IFs, which is constructed on the basis of a controlled laboratory testbed. The employed videos have different types and complexities. 62 testers participated in the test step, which are researchers and students from several fields ranging in age from 20 to 37 years and having little or no expertise with video evaluation experiments. The QoE scores collected in this dataset represent the single rating score provided by the user, which lies between [1, 5]. The lower the score value, the lower the video quality. The dataset contains five distinct categories of features, which are listed below:

- The video content feature: Text information
- The bandwidth feature : Categorical information
- The packet-loss feature: Categorical information
- The delay feature: Continuous values
- The jitter feature: Continuous values

[1] https://github.com/Lamyne/Poqemon-QoE-Dataset

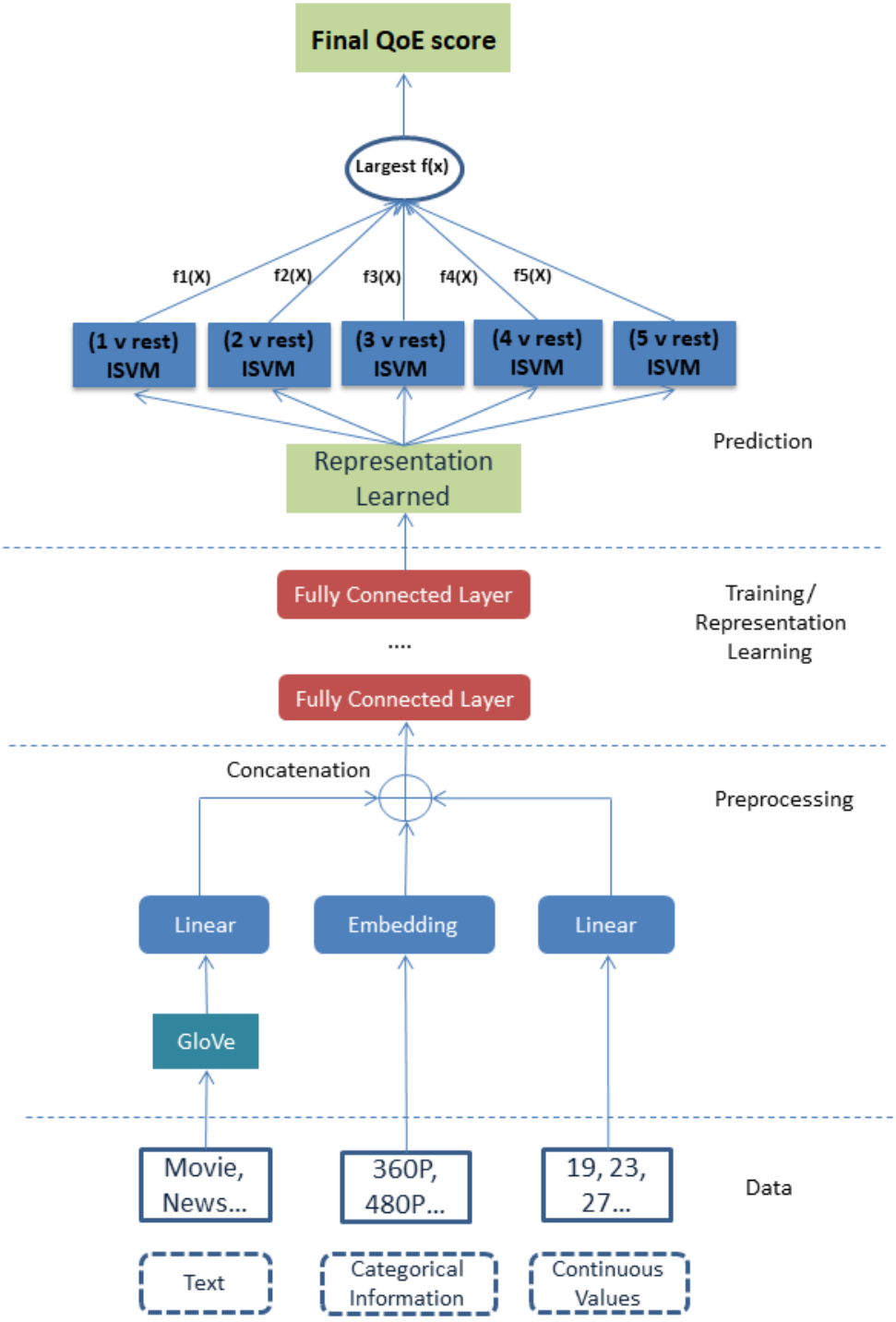


Figure 2. DeepISVM framework architecture

LIVE-Netflix Video Database [28]: Compared to the existing datasets, the mentioned dataset can be employed for real-time context. In particular, for video streaming scenarios, as it incorporates networking data including rebuffering and compression. The dataset comprises 14 distinct categories of videos (action, comedy, anime, etc.) with 1080P as resolution. The videos are treated to 8 distinct playout patterns, which include dynamically varying H.264 compression rates, re-buffering events, and a combination of both at 24, 25, and 30 fps. As a result, 112 videos are produced. 55 testers participated in the test step using mobile devices and they provide continuous and retrospective QoE scores. In this study, we have used only the 112 retrospective scores. The subjective QoE scores values lie between [-2.28, 1.53]. The dataset includes five features of different types as detailed below:

- The objective video quality assessment (VQA) feature: Continuous values
- The re-buffering duration feature: Continuous values
- The re-buffering times feature: Categorical information
- The memory feature: Continuous values
- The impairment duration feature: Continuous values

LFOVIA Database [29]: The dataset comprises 36 distinct video sequences of 120 seconds duration, derived from 18 reference videos. Videos are of FHD and UHD resolutions. The dataset comprises a different variety of content including nature, wildlife, outdoor, and marine. Training and testing processes are carried out on this dataset using the same manner as detailed in [16]. As a result, there are 36 train-test subsets, where 25 of the 36 videos are used to train our model for every test video. The QoE scores lie between 0 and 100. The lower the score value, the lower the video quality.
From the dataset features, three features of different types are employed, as detailed below:

- The Short Time Subjective Quality (STSQ) feature: Continuous values
- The Playback Indicator (PI) feature: Categorical information
- The Time elapsed since last Rebuffering (TR): feature: Continuous values

B. Experimental protocol

The performance of the proposed video streaming QoE estimation approach is evaluated in two phases:
First, to demonstrate the advantage of using deep features rather than original dataset features, we perform a comparison between several ML techniques when using original dataset features, and the same ML techniques when using representations derived from DeepQoE part.
Second, to highlight the benefit of the proposed deep ISVM method over batch single and ensemble learning approaches, we compare our model to various relevant approaches.
The same features databases and experimental settings are used to evaluate these classifiers. The radial basis kernel was employed for kernelization by SVM and ISVM based models. This kernel is defined as $k(x_i, x_j) = e^{-\|x_i - x_j\|^2/\sigma}$, with σ denoting a positive "width" parameter.
We used the 10-fold cross-validation technique to ensure that the outcomes are premeditated and unbiased [30]. More precisely, we randomly split the original database into 10 subsets. This way, a single subset is retained for testing, and the rest are employed as training data. Finally, the global accuracy is calculated by averaging the ten obtained accuracies.
Moreover, we used the Accuracy and the F1-Score as measures of a classifier's performance. The Accuracy is the percentage of correct results that a classifier has achieved out of the total number of observations in the dataset. The F1-Score is defined as a harmonic mean of the precision and recall scores, where the precision denotes the success for a situation deemed as positive and the recall denotes how successful positive situations have been assessed.

TABLE I. Mapping between Continuous rating scores and Discrete rating scores.

Continuous rating score	Discrete rating score
−2.28 <= & < −1.51	1
−1.51 <= & < −0.75	2
−0.75 <= & < 0	3
0 <= & < 0.76	4
0.76 <= & <= 1.53	5

C. Implementation details

In this section, we describe the evaluation procedure of each dataset. We then present evaluation results.

1) LIVE-Netflix Video Dataset

In this experiment, we have used the LIVE-Netflix database for the classification task, hence the data rating scores are converted from the provided form (lie between [-2.28, 1.53]) to 5 equal groups laying from 1 to 5 as mentioned in the Table 1.

As we have only a small number of examples, we average over 1000 trials to achieve consistent results. In each trial, 112 randomly selected rating scores are divided into 80% for the training process and 20% for the test process.
During the training process of our DeepISVM model, the VQA feature is transformed into a 20-dimensional vector after a linear layer. The re-buffering duration feature is converted to a 5-dimensional vector after a linear layer. The memory feature and the impairment feature are mapped

TABLE II. Mapping between Continuous rating scores and Discrete rating scores.

Continuous rating score	Discrete rating score
0 <= & < 20	1
20 <= & < 40	2
40 <= & < 60	3
60 <= & < 80	4
80 <= & <= 100	5

to a 10-dimensional vector. Finally, the re-buffering times feature is converted to a 5-dimensional vector after an embedding layer.
Concatenation of the obtained feature vectors results in one feature vector that will be used as input for the second phase, which is composed of 3 successive fully connected layers. Dropout is used to avoid overfitting, with a value of 0.5.

2) Poqemon Dataset

In the first step, the video content characteristic (specified by words) is converted to a 50-dimensional vector using a pre-trained GloVe method. The delay feature and the jitter feature are converted to a 10-dimensional vector after a linear layer. Finally, the bandwidth and the packet-loss features are converted to a 5-dimensional vector after an embedding layer.
One feature vector is created by concatenating the acquired feature vectors that will be used as input for the second phase, which is composed of three consecutive fully connected layers. Dropout is also used with this dataset, with a ratio of 0.5.

3) LFOVIA Database

We have employed the LFOVIA database for the classification task, hence the data rating scores are converted from the provided form (lie between [0, 100]) to 5 equal groups laying from 1 to 5 as mentioned in the Table 2.

During the training process of our DeepISVM model, the STSQ feature is transformed into a 20-dimensional vector after a linear layer. The playback indicator feature is converted to a 5-dimensional vector after an embedding layer. Finally, the time elapsed since last rebuffering feature is converted to a 20-dimensional vector after a linear layer. The obtained feature vectors are concatenated to form a single feature vector that will serve as input for the second phase, which is composed of 3 successive fully connected layers. Dropout is used to avoid overfitting, with a value of 0.5.

D. Results and discussion

This section presents the result of the experimental comparisons and evaluations that we conducted, in order to highlight the superiority of our deep incremental learning model over other well-known learning methods, using three datasets.

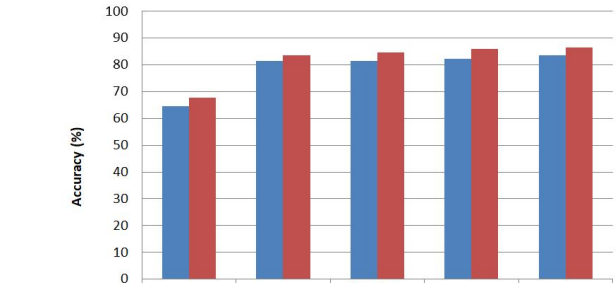


Figure 3. Performance comparison for the Poqemon dataset between employing the original features and employing deep features derived from the Deep QoE model.

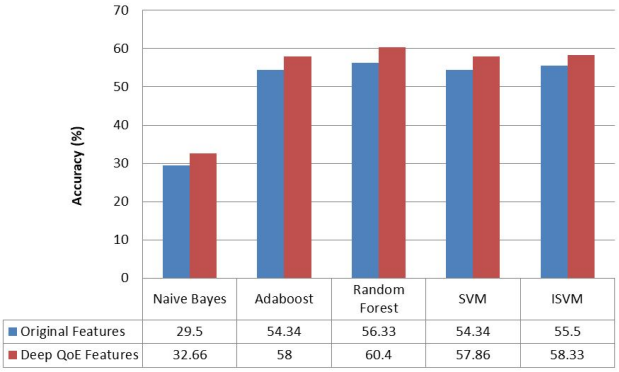


Figure 4. Performance comparison for the LIVE-Netflix Video dataset between employing the original features and employing deep features derived from the Deep QoE model.

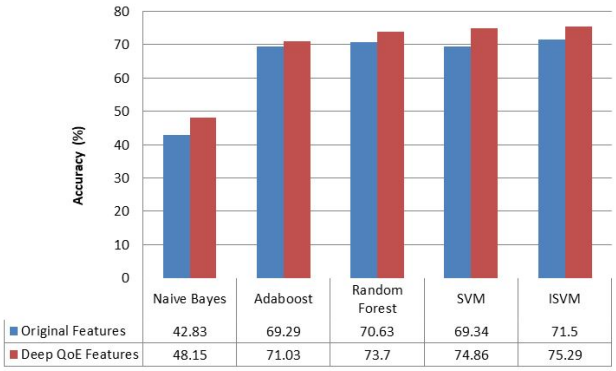
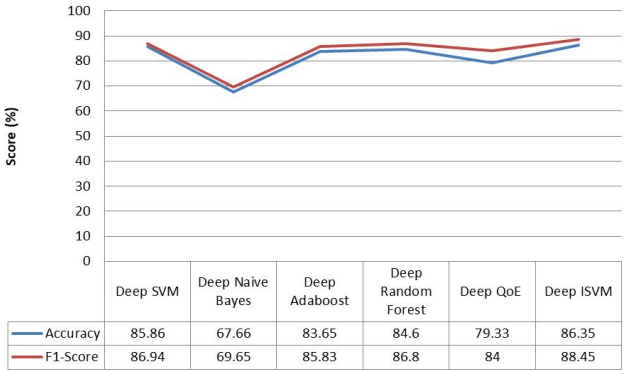


Figure 5. Performance comparison for the LFOVIA Video dataset between employing the original features and employing deep features derived from the Deep QoE model.

First, to evaluate the effectiveness of using the deep learning part for the feature extraction and representation in our DeepISVM model, we compare the performance of several models using original features and deep features, separately. As we can see from Figure 3, Figure 4, and Figure 5 the accuracy of all algorithms is improved by

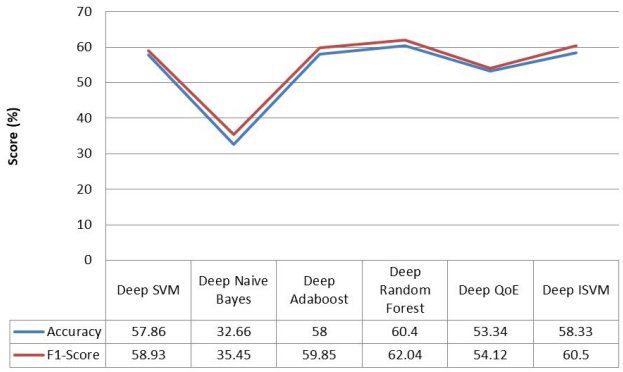
using deep features. That was expected given that the DeepQoE part takes advantage of the pre-trained models to overcome the dataset size constraint. This demonstrates the effectiveness of deep learning model's representation.

We evaluate the effectiveness of the suggested method by comparing the DeepISVM classifier with five other learning models. As we can see in Figures 6, 7, and 8, our proposed model produces superior outputs in terms of accuracy and F1-Score measures when employing the Poqemon dataset and LFOVIA dataset. Also, with the LIVE-Netflix Video Dataset, our model outperforms all models except the Deep Random Forest (DRF) model. This exception is because of the small dataset size (112 rating scores). Therefore, the DRF model can give better results by taking advantage of the ensemble learning process. But with the growth of datasets, the complexity of the models will also grow, which causes a decrease in the performance of models based on the decision tree model (Random Forest, AdaBoost). In contrast, thanks to the use of the ISVM model for the QoE prediction in our framework, the proposed model is not affected by the growth of databases, as data are trained incrementally. So, the margins are adjusted and improperly classified examples may be inserted into the support vector subset. As a result, each choice made for new information will be employed incrementally to update and enhance the ISVM classifier's preceding result. Furthermore, because the support vectors are assessed incrementally, multi-class ISVM provides a cleaner solution. As can be observed from Table 3 and Table 4, the execution time and the model size of our DeepISVM model is slightly high compared to the Deep QoE model. This was expected because we have added an incremental layer to the existing Deep QoE architecture for the estimation of the video streaming QoE, which results in a complexity overhead. On the other hand, the proposed model has higher training speed compared to the other deep batch classifiers. That is because the ISVM model is capable of receiving and integrating new examples without having to retrain the dataset from scratch. More specifically, unlike the complexity of the SVM, which is equal to $O(n^3)$, where n represents the number of samples used for training, the incremental SVM complexity is $O(ns^2)$, where ns denotes the number of support vectors and $ns \leq n$. This is made possible by utilizing the Woodbery formula to recalculate the gradient, β and γ, and to perform matrix vector multiplication and recursive updates of the matrix R (section 3.2), which has a dimension equal to the number of support vectors ns. As a result, the execution time required for updating R is quadratic in the number of support vectors. In contrast, when using batch models the whole training procedure should be repeated.



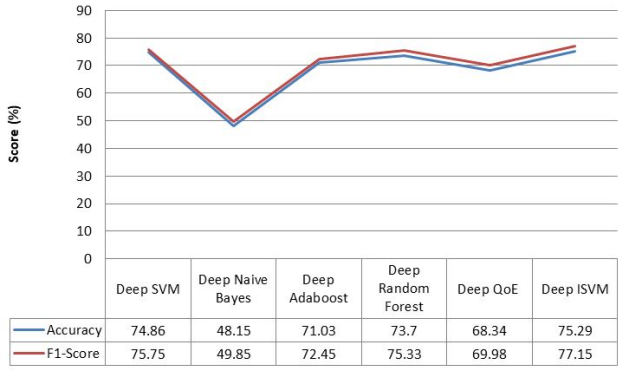
	Deep SVM	Deep Naive Bayes	Deep Adaboost	Deep Random Forest	Deep QoE	Deep ISVM
Accuracy	85.86	67.66	83.65	84.6	79.33	86.35
F1-Score	86.94	69.65	85.83	86.8	84	88.45

Figure 6. Performance comparison of different deep learning methods for Poqemon dataset.



	Deep SVM	Deep Naive Bayes	Deep Adaboost	Deep Random Forest	Deep QoE	Deep ISVM
Accuracy	57.86	32.66	58	60.4	53.34	58.33
F1-Score	58.93	35.45	59.85	62.04	54.12	60.5

Figure 7. Performance comparison of different deep learning methods for LIVE-Netflix Video dataset



	Deep SVM	Deep Naive Bayes	Deep Adaboost	Deep Random Forest	Deep QoE	Deep ISVM
Accuracy	74.86	48.15	71.03	73.7	68.34	75.29
F1-Score	75.75	49.85	72.45	75.33	69.98	77.15

Figure 8. Performance comparison of different deep learning methods for LFOVIA Video dataset

5. Conclusion

We have suggested a new deep incremental SVM method for the online prediction of the QoE of video streaming services. The presented model is built on the combination of a Deep QoE layer for the features preprocessing and representation, and an ISVM model for the online prediction of the QoE score. As a result, a powerful deep incremental model is constructed.

We performed rigorous comparative experimental evaluations on three datasets, where we compared our proposed method against several machine/deep learning models in terms of precision and complexity.

The evaluation outputs show the superiority of the DeepISVM model. In fact, the proposed approach inherits the

TABLE III. Average execution time in millisecond (ms) of Deep ISVM vs other deep learning models.

Machine learning model	LIVE-Netflix Video	Poqemon	LFOVIA
Deep ISVM	25.68	39.42	43.33
Deep QoE [11]	25.5	39.25	43.02
Deep Adaboost	201.89	403.56	620.23
Deep Random forest	43.29	64.33	79.89
Deep SVM	28.76	43.8	49.28
Deep Naïve Bayes	27.1	41.12	44.11

TABLE IV. The model size in kilobyte (kB) of Deep ISVM vs other machine learning models.

Machine learning model	LIVE-Netflix Video	Poqemon	LFOVIA
Deep ISVM	30.02	39.87	38.26
Deep QoE [11]	29.15	38.45	37.16
Deep Adaboost	64.04	79.33	72.37
Deep Random forest	39,76	50.83	47.77
Deep SVM	6094,67	322.46	225.8
Deep Naïve Bayes	58.53	60.01	59.39

benefits of employing a DeepQoE framework and an incremental learning process.
In future work, we will add the C3D pre-trained method to extract deep video characteristics which have discriminative power and lead to performance improvement.

6. ACKNOWLEDGMENT

We thank Michael Hutton of the öbex project for language editing support.

7. Conflicts of interest

Not applicable

References

[1] Cisco, "Cisco annual internet report (2018–2023)," Cisco Systems Inc., Tech. Rep. C11-741490-01, 2022.

[2] G. Ananthanarayanan, P. Bahl, P. Bodik, K. Chintalapudi, M. Philipose, L. Ravindranath, and S. Sinha, "Real-time video analytics: The killer app for edge computing," *Computer (Long Beach Calif.)*, vol. 50, no. 10, pp. 58–67, 2017.

[3] Q. Huynh-Thu and M. Ghanbari, "Temporal aspect of perceived quality in mobile video broadcasting," *IEEE Trans. On Broadcast.*, vol. 54, no. 3, pp. 641–651, Sep. 2008.

[4] T. Hoßfeld, M. Seufert, M. Hirth, T. Zinner, P. Tran-Gia, and R. Schatz, "Quantification of YouTube QoE via crowdsourcing," in *2011 IEEE International Symposium on Multimedia*. IEEE, Dec. 2011.

[5] S. Tasaka, "Bayesian hierarchical regression models for QoE estimation and prediction in audiovisual communications," *IEEE Trans. Multimedia*, vol. 19, no. 6, pp. 1195–1208, Jun. 2017.

[6] T. Zhao, Q. Liu, and C. W. Chen, "QoE in video transmission: A user experience-driven strategy," *IEEE Commun. Surv. Tutor.*, vol. 19, no. 1, pp. 285–302, 2017.

[7] X. Deng, L. Chen, F. Wang, Z. Fei, W. Bai, C. Chi, G. Han, and L. Wan, "A novel strategy to evaluate QoE for video service delivered over HTTP adaptive streaming," in *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*. IEEE, Sep. 2014.

[8] J. Li, O. Kaller, F. De Simone, J. Hakala, D. Juszka, and P. Le Callet, "Cross-lab study on preference of experience in 3DTV: Influence from display technology and test environment," in *2013 Fifth International Workshop on Quality of Multimedia Experience (QoMEX)*. IEEE, Jul. 2013.

[9] J. Song, F. Yang, Y. Zhou, S. Wan, and H. R. Wu, "QoE evaluation of multimedia services based on audiovisual quality and user interest," *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 444–457, Mar. 2016.

[10] O. B. Maia, H. C. Yehia, and L. de Errico, "A concise review of the quality of experience assessment for video streaming," *Computer Communications*, vol. 57, pp. 1–12, 2015.

[11] H. Zhang, L. Dong, G. Gao, H. Hu, Y. Wen, and K. Guan, "DeepQoE: A multimodal learning framework for video quality of experience (QoE) prediction," *IEEE Trans. Multimedia*, vol. 22, no. 12, pp. 3210–3223, Dec. 2020.

[12] "Survey on machine learning-based QoE-QoS correlation models," in *2014 International Conference on Computing, Management and Telecommunications (ComManTel)*. IEEE, Apr. 2014.

[13] C. G. Bampis and A. C. Bovik, "Learning to predict streaming video QoE: Distortions, rebuffering and memory," *arXiv [cs.MM]*, Mar. 2017.

[14] M. Ghosh and C. Singhal, "MO-QoE: Video QoE using multi-feature fusion based optimized learning models," *Signal Process. Image Commun.*, vol. 107, no. 116766, p. 116766, Sep. 2022.

[15] S. Ickin, M. Fiedler, and K. Vandikas, "Customized video QoE estimation with algorithm-agnostic transfer learning," Mar. 2020.

[16] 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.

[17] K. Lavanya, K. V. Devi, and M. Subramaniam, "Quality of experience (QOE) content aware hybrid lean predictive models for medical video transmission over internet of things (IOT) networks," *International Journal of Communication Systems*, Jun. 2021.

[18] M. Ghosh, D. C. Singhal, and R. Wayal, "DeSVQ: Deep learning based streaming video QoE estimation," in *23rd International Conference on Distributed Computing and Networking*, ser. ICDCN 2022. New York, NY, USA: Association for Computing Machinery, 2022, pp. 19–25.

[19] V. Menkovski, G. Exarchakos, and A. Liotta, "Online QoE prediction," in *2010 Second International Workshop on Quality of Multimedia Experience (QoMEX)*. IEEE, Jun. 2010.

[20] R. Elwerghemmi, M. Heni, R. Ksantini, and R. Bouallegue, "Online QoE prediction model based on stacked multiclass incremental support vector machine," in *2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO)*. IEEE, Apr. 2019.

[21] V. Losing, B. Hammer, and H. Wersing, "Incremental on-line learning: A review and comparison of state of the art algorithms," *Neurocomputing*, vol. 275, pp. 1261–1274, Jan. 2018.

[22] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," Dec. 2014.

[23] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2014.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[25] V. N. Vapnik, *Statistical learning theory*, ser. Adaptive and Cognitive Dynamic Systems: Signal Processing, Learning, Commun ications and Control. Nashville, TN: John Wiley & Sons, Sep. 1998.

[26] P. Laskov, C. Gehl, S. Krueger, and K.-R. Müller, "Incremental support vector learning: Analysis, implementation and applications," *Journal of Machine Learning Research (JMLR)*, vol. 7, 2006.

[27] "ITU-T recommendation, approved in 1999-09," *Subjective video quality assessment methods for multimedia applications*, vol. 910, 2008.

[28] 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.

[29] 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.

[30] C. Alippi and M. Roveri, "Virtual k-fold cross validation: An effective method for accuracy assessment," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*. IEEE, Jul. 2010.

Radhia Elwerghemmi received the M.Sc. degree in computer science and multimedia from the Higher Institute of Computing and Multimedia of Gabes, Gabes University, Tunisia, in 2015. She is currently pursuing the Ph.D. degree with the research laboratory Innov'COM, Sup'Com, where she has worked on QoE prediction for video streaming services.

Maher Heni received the national engineering diploma in telecommunication from the National Engineering School of Tunis (ENIT), the master diploma in Telecommunications from ENIT with collaboration of IEF institute in parissud university, and the Ph.D. degree in Telecommunications in 2013 from the Higher School of Communication (Sup'COM), Tunisia. His research interests are ad hoc networks, mobile communication systems, and Big Data.

Riadh Ksantini received the M.Sc. and Ph.D. degrees in Computer Science from the Université de Sherbrooke, Sherbrooke, QC, Canada, in 2003 and 2007, respectively. From 2001 to 2007, he was a graduate research associate with Bell Canada Laboratories and the research center MOIVRE (MOdElisationen Imagerie, Vision et REseaux de neurones). Presently, he is Associate Professor at the Department of Computer Science, College of IT, University of Bahrain, Adjunct Associate Professor at the School of Computer Science, within the Faculty of Science of the University of Windsor, Windsor, Ontario, Canada, and Adjunct Professor at the Department of Computer Science, Université du Québec à Montréal (UQAM). Prior to that, he was Postdoctoral Research Associate in the Ecole de Technologie Superieure, University du Quebec, Montreal, Canada. He has also served as Visiting Fellow Research Scientist at the Canadian Space agency, and Postdoctoral Research Associate in the School of Computer Science, within the Faculty of Science of the University of Windsor. In 2008, he was awarded a fellowship (of excellence) for postdoctoral research from the granting agency ¨Fonds quebecois de la recherche sur la nature et les technologies ¨ (FQRNT). His PhD was evaluated and ranked third Ph.D. in Quebec for 2007 by the committee of Information Technology and Communications. His research interests include Artificial Intelligence, Machine/Deep Learning, Pattern Recognition and Computer Vision. His research work on Artificial Intelligence and Data Science has always involved collaboration between academia and industry. More than 70 Articles stemming from his research work have been published in several prestigious journals and conferences.

Ridha Bouallegue (Member, IEEE) received the M.S., Ph.D., and H.D.R. degrees in telecommunications from the National Engineering School of Tunis (ENIT), Tunisia, in 1990, 1994, and 2003, respectively. He is currently a Professor with ENIT and the Director of the research laboratory Innov'COM/Sup'Com. His current research interests include mobile and satellite communications, access techniques, intelligent signal processing, code division multiple access (CDMA), multi-in multi-out (MIMO), OFDM, and UWB systems..