QoE Multi-Stage Machine Learning for Dynamic Video Streaming

Michele De Filippo De Grazia, Daniel Zucchetto, *Student Member, IEEE*, Alberto Testolin, Andrea Zanella, *Senior Member, IEEE*, Marco Zorzi, and Michele Zorzi, *Fellow, IEEE*

***Abstract*—The rapid growth of video traffic in cellular networks is a crucial issue to be addressed by mobile operators. An emerging and promising trend in this regard is the development of solutions that aim at maximizing the quality of experience (QoE) of the end users. However, predicting the QoE perceived by the users in different conditions remains a major challenge. In this paper, we propose a machine learning approach to support QoE-based video admission control and resource management algorithms. More specifically, we develop a multi-stage learning system that combines the unsupervised learning of video features from the *size* of H.264-encoded video frames with a supervised classifier trained to automatically extract the quality-rate characteristics of unknown video sequences. This QoE characterization is then used to manage simultaneous video transmissions through a shared channel in order to guarantee a minimum quality level delivered to the final users. Simulation results show that the proposed video admission control and resource management algorithms, which are based on learning-based QoE classification of video sequences, outperform standard content-agnostic strategies.**

***Index Terms*—Resource management, feature extraction, Boltzmann machines, traffic control (communication), video signal processing.**

I. Introduction

NOWADAYS, the most appealing but also the most bitrate demanding services are those providing high-quality videos to users playing real-time streaming or progressive download applications. The deployment of heterogenous high-speed access points, such as LTE femto-cells and WiFi hotspots, dramatically increases the number of users accessing the network, which has an impact on the performance of both uplink and downlink channels. To cope with this issue, mobile operators need to increase the network capacity to effectively support high-quality and bitrate demanding services with the available network resources, while keeping mobile infrastructure costs at a reasonable level.

A good trade-off between perceived Quality-of-Experience (QoE) to be offered to the mobile users and smart use of network resources is achieved by dynamically adapting the coding rate of the requested videos to the available transmission resources. As observed in [2], reducing the encoding rate of a video is indeed much less critical in terms of QoE degradation than increasing the packet loss probability or the delivery delay. However, the perceived QoE at a certain encoding rate depends on the video content itself, e.g., the dynamics of the scene, the mobility of the source and frame-by-frame motion, which are not easy to predict. Knowing these characteristics would make it possible to adjust the video rates according to the available transmission resources, so as to maximize the QoE of the users.

In this paper we present a cognitive approach for video delivery in communication-constrained scenarios. The basic idea is to combine unsupervised and supervised machine learning techniques to infer the Quality-Rate (Q-R) characteristics[1] of the video sequences from high level information, readily available at the network layer.

We consider a scenario where a number of mobile users request video content from some remote servers, using a shared channel. We assume that videos are provided by the servers in the form of short chunks of a few seconds each (called *video segments*), which are then delivered to the mobile users through HTTP sessions, similarly to Dynamic Adaptive Streaming over HTTP (DASH) [3], [4]. Therefore, there is no need to maintain long streaming sessions between server and mobile users, dramatically simplifying mobility management. Each video streaming session starts with an HTTP request sent by the mobile user to the video server for the list of the titles and formats of the available videos [5]. Each DASH file is indeed associated to a Media Presentation Description (MPD) that provides information characterizing the video file and the available locations of the segments, and may contain multiple representations for the same media, that is, multiple versions with different resolutions and bitrates. A DASH client is then able to dynamically select the desired representation of each chunk of the video and to get it via HTTP.

Manuscript received April 3, 2017; revised October 3, 2017; accepted December 10, 2017. Date of publication December 18, 2017; date of current version March 7, 2018. A preliminary version of this work was presented at IEEE Med-Hoc-Net 2014 [1]. The associate editor coordinating the review of this paper and approving it for publication was A. B. MacKenzie. *(Corresponding author: Andrea Zanella.)*

M. De Filippo De Grazia, A. Testolin, and M. Zorzi are with the Department of General Psychology, University of Padova, 35131 Padova, Italy (e-mail: michele.defilippodegrazia@unipd.it; alberto.testolin@unipd.it; marco.zorzi@unipd.it).

D. Zucchetto, A. Zanella, and M. Zorzi are with the Department of Information Engineering, University of Padova, 35131 Padova, Italy (e-mail: zucchett@dei.unipd.it; zanella@dei.unipd.it; zorzi@dei.unipd.it).

Digital Object Identifier 10.1109/TCCN.2017.2784449

[1]The Q-R characteristic is often expressed in the literature in terms of rate-distortion curve, which conveys the same information, though presented in a different form.

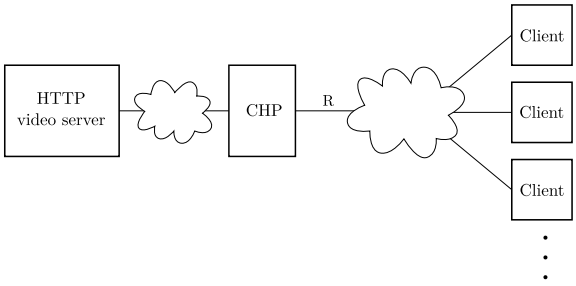


Fig. 1. Reference scenario: the Cognitive HTTP Proxy (CHP) implements the RM and VAC mechanisms to manage the rates of the active video flows across the bottleneck link of capacity R [bit/s].

While the DASH framework is well established, the quality-adaptation policy is still open to investigation. Typically, the policies adopted by legacy DASH clients are based on local measurements, such as the number of buffered video segments at the client side, or the estimated average downlink throughput. Instead, the actual Q-R characteristics of the streamed videos, or the number and type of contending flows, are not commonly considered.

In this work, we adopt a more systematic approach and propose a solution where the rate of each competing video flow is determined in a centralized manner by a *Cognitive HTTP proxy* (CHP), as represented in Fig. 1. The CHP can be instantiated in the access router of a private network, to control the video traffic towards the hosts of the network. Furthermore, leveraging the upcoming Network Function Virtualization (NFV) paradigm, instances of the CHP can be activated where multiple video flows merge into the same shared link, in order to provide minimum performance guarantees to multimedia flows and/or blocking excess video traffic.

The proxy intercepts all HTTP requests, performs traffic classification, and applies *Video Admission Control* (VAC) and *Resource Management* (RM) algorithms to improve the QoE of the clients. In particular, the CHP will be able to intercept, interpret, and modify the DASH packets exchanged by video clients and servers, thus performing a dynamic adaptation of the video quality according to a certain utility function, which depends on the Q-R characteristic of *each single video*, which is automatically estimated by using a multi-stage machine learning approach. The Q-R characteristic is able to summarize, in a single function, the map between Quality of Service (QoS) and QoE parameters. Such map is necessary since the VAC and RM mechanisms aim to maximize the QoE while satisfying some QoS constraints of the network (such as maximum channel capacity or minimum end-to-end delay).

Crucially, the proposed method does not require to process the original content of the video frames, but only uses information readily available at the network layer after the encoding process, namely the *size of the video frames*, with some other parameters that can be easily retrieved from the MPD file associated to the video, such as the structure of the Group-of-Pictures (GOP) used during the encoding, the resolution of the video, and the frame rate. The rationale is that the Q-R function of a video is closely related to the dynamics of its content that, in turn, impacts the spatial and temporal redundancy of the video frames and, consequently, the size of the frames generated by the encoder [1], [6]. Highly dynamic videos, containing complex spatial and temporal structure, will likely result in larger frame sizes, while more static videos will be likely encoded in frames of smaller and more homogeneous size.

To test the proposed method, we built a training dataset containing the frame sizes for a number of HD and CIF video clips, encoded at different compression levels. The dataset was then used to perform the unsupervised training of a Restricted Boltzmann Machine (RBM) [7]. The RBM captures the latent features of the input data, thus providing a high-level representation of the video segments at different compression levels, which can be exploited by supervised learning algorithms to estimate the Q-R characteristics of unknown videos. In our study, we consider the average Structural SIMilarity (SSIM) index [8] of the frames in a GOP as a measure of the perceived quality of a video segment. We remark that SSIM is not the only objective metric for QoE assessment of video sequences, nor is necessarily the best in all cases. The Q-R characteristics of a video can indeed be expressed with other metrics, either full reference (i.e., where the evaluation system has access to the original media) like the NTIA-Video Quality Metric General Model [9] and the MOVIE index [10], or no-reference, e.g., Video BLIINDS [11].

We observe that the SSIM focuses on the spatial dimension only, i.e., the quality of the image captured by the frames, while neglecting the time dimension that can be crucial to correctly assess the degradation of the visual experience due to gaps in the video streaming (freezing and rebuffering events) or sharp variations of the visual quality of the video frames. As it will be better discussed in Section VIII, however, when the link bitrate is known (as assumed in this study), suitable VAC and RM algorithms can choose the bitrates of the video segments to fit into the available channel capacity, thus avoiding that the client runs out of frames to play out. In this scenario, where the temporal aspect of the QoE metric is less critical, SSIM represents a reasonable low-complexity choice. In addition, we remark that the proposed framework can be applied to other QoE metrics with a qualitative similar Q-R characteristics (i.e., such that the quality increases with the frame size and the data rate).

To summarize, based on a machine learning scheme, we estimate the Q-R characteristics (in terms of SSIM vs normalized bitrate) of unknown videos from the distribution of the coded frame sizes. This characterization is then fed into QoE-aware VAC and RM algorithms. By means of simulations, we show that combining unsupervised feature extraction and linear classification provides better results than a more basic approach that tries to extract the SSIM characteristics directly from the raw data. Furthermore, we show that QoE-based VAC and RM algorithms make a better use of the available transmission resources than content-agnostic schemes and provide a valuable tool for quasi-realtime adaptive video streaming applications.

The remainder of the paper is organized as follows. In Section II we review the related work. Our video analysis is presented in Section III. The machine learning

approach is described in Section IV and validated in Section V. In Section VI we describe the QoE-based and QoE-agnostic resource management algorithms, and compare their performance by simulation in Section VII. Section VIII discusses possible improvements to the proposed approach and extends the performance analysis to some more challenging scenarios. Finally, Section IX concludes the paper.

II. Related Work

In this section we first overview the state of the art on DASH adaptation logics and then consider the literature on the objective quality metrics for video sequences, which represent the two main building elements of our approach.

A. Adaptation Logics for DASH Video Streaming

As briefly mentioned in the introduction, in the DASH framework, the video clips are split in short time segments, which are encoded at different compression levels and stored at the video server as independently addressable and reproducible multimedia objects. This makes it possible to download any of the available versions of each video segment, thus enabling the dynamic adaptation of the video rate (and, in turn, quality) to the channel conditions in order to guarantee good video quality, uninterrupted play out, and smooth quality variations.

For example, the scheme proposed in [12] privileges the stability of the video bitrate over instantaneous video quality, thus adopting a conservative approach when increasing the bitrate that also yields a lower probability of freezing events. Probe and Adapt (PANDA) [13] makes use of active channel probing to estimate the path throughput and adapt the video rate accordingly. To prevent fluctuations due to cross-traffic variations, however, the scheme adopts a conservative rate-increasing strategy when the channel capacity grows and hysteresis margins to avoid frequent rate switches. A similar but simpler heuristic was presented by Petrangeli *et al.* in 2014 [14]. The mechanisms proposed in [15] uses only buffer state information to adapt the video bitrate, resorting to channel capacity estimation only during transient periods. As shown in [16], however, such simple schemes may not be able to guarantee high video quality, even when the channel capacity is constant. More complex approaches make use of predictive or Markov Decision Process techniques to model the variations of the channel capacity and find the optimal adaptation strategy [17]–[19]. The main limit of these approaches is the computational load: the model is too complex to be solved at runtime. To overcome this issue, some recent works apply reinforcement learning techniques to automatically learn the best adaptation strategy from the past experience [20]. However, this approach is limited by the training time of the machine-learning algorithms, which grows very quickly with the size of the state space [21]. Alternatively, the state space can be roughly quantized to speed up the learning process to the detriment of the achieved performance [22], [23].

The work in [24] defines the bitrate adaptation strategy as an optimization problem, applied to a Markovian channel model. The function that links the video bitrate to the video quality, however, is not bound to any perceived quality metric. Chen *et al.* [25] developed a network-side mechanism to adapt video bitrate based on information from the clients that, however, are required to be all compliant with this protocol, which limits its practicality. A heuristic cross-layer algorithm for wireless networks is presented in [26], where both end-to-end bandwidth estimation and measurements from the WiFi link are used to determine the frame quality to download from the server.

For a more comprehensive overview of existing adaptation techniques for DASH we refer the reader to [27].

The main focus of this work is not to provide another adaptation algorithms for DASH. Rather, our purpose is to propose a new methodology to infer the Q-R characteristics of each single video sequence and to show how this information can be successfully exploited in a DASH framework. For this reason, we consider a rather simple scenario, where the transmission resources reserved to video contents are constant and can be arbitrarily assigned to the different flows. Therefore, rate-adaptation is only required to reallocate channel resources when new video flows are accepted into the system or active ones are terminated. The Q-R estimation technique we propose can be combined with more sophisticated DASH algorithms and employed in more challenging scenario, whose investigation however is left to future work.

B. Objective Quality Metrics

Prior work on video detection over communication networks mainly focuses on extracting objective networking and quality metrics. Xu and Li [28] classify videos based on selected common spatial-temporal audio and visual features described by the MPEG-7 compliant content descriptors. Due to the complexity of the method, the authors make use of principal component analysis (PCA) to reduce the set of features under study. Nevertheless, this work is strictly dependent on the MPEG-7 multimedia format.

The work in [29] marks the packets using a pre-congestion notification mechanism in order to detect congestion in the network. A linear programming method is then used to assign a quality level to each video flow, in order to maximize a revenue function. The considered quality levels, however, are only described by video resolution and bitrate, not by a metric that properly evaluates the perceived quality.

Qadir and Kist [30] exploit a measurement-based admission control mechanism for video flows in order to maximize the number of admitted video requests in a network. Again, the considered metric is the video bitrate, while the perceived video quality is not considered. Also, this technique requires to know the state of the entire network in order to solve the admission control problem, which may be infeasible in large networks.

Further related work focuses on quality prediction models to capture the behavior of video scenes. Feitor *et al.* [31] propose an objective model to predict the quality of 3D videos in the presence of frame losses, which is based on the header information of the video packets at different ISO/OSI layers. This model is able to roughly capture the SSIM of some video

clips based on the size of the lost frames and via deep packet inspection, which is usually avoided by operators in cellular deployments due to complexity and users' privacy concerns. Also, a model to extract the channel induced distortion in a no-reference fashion is described in [32]. The described algorithm exploits the received prediction residuals, coding modes, and the received and concealed motion vectors to compute an approximation of the SSIM index, therefore still requiring deep packet inspection. In any case, Seeling *et al.* [33] claim that the frame loss probability, which is mainly a network metric, provides only limited insight into the video quality perceived by the user. The paper [34] describes a model to map network QoS factors to a QoE value, whose complexity however makes it unsuitable for online applications. Other studies use learning techniques to predict video QoE from traffic data, including factors as the frequency of bitrate variations and the freezing events. However, the accuracy of the predicted QoE values is rather coarse [35], [36].

In our work, we analyze and group video test sequences based on the relation between video compression rate and SSIM. It is widely recognized that the SSIM index provides a more accurate QoE indication than more traditional metrics, like PSNR and mean square error (MSE), which have proven to be inconsistent with perceptual experience. Although the SSIM characterization of a video sequence is computationally expensive, many studies have shown that the extraction of perceived quality metrics, like SSIM, from the features of the encoded video is feasible. In [37] an artificial neural network is used to extract the SSIM of a video sequence using information on quantization parameters, frame structure, and motion vectors. Lin *et al.* [38] approximate the SSIM using, instead, an extension of the Support Vector Machine (SVM), namely the ϵ-Support Vector Regression. In this case, the considered features are derived from the frame structure, the quantization parameters, and the motion vectors. A much larger feature space is considered in [39], where 20 features describing the frame structure, motion vectors, and texture information are fed into a model, which is estimated using multipass polynomial regression. A simpler linear regression is employed in [40], which, however, is able to estimate both SSIM and Video Quality Metric (VQM) [9] for noisy channels using features related to motion vectors, the mean residual energy of the frames and error concealment information. In a related way, [41] describes the use of a multi linear regression technique to extract different video quality metrics (including PSNR, SSIM, and VSSIM) from the video bitrate and frame rate, and from information on motion vectors and on frame and group-of-picture structures. All of these methods, however, require the extraction of a large number of features from the video stream, thus requiring deep packet inspection and considerable computational cost.

Machine learning algorithms represent the state of the art in many classification tasks, especially when the structure of the domain is difficult to characterize. The problem of automatic video processing is closely related to that of image processing, with the additional complexity given by the temporal dimension of the data. In the so-called "content-based" video retrieval [42], for instance, a range of different techniques can be applied depending on the task of interest, e.g., video indexing, scene recognition and/or classification, object tracking, and motion detection. In recent years, advances in the theory and practice of probabilistic graphical models and statistical learning led to the development of extremely powerful deep learning systems, which achieve state-of-the-art performance in several machine vision tasks [43], [44]. Although the main application of these systems has been primarily focused on still frames, there have also been successful extensions to the temporal domain [45], [46].

All the above-mentioned machine learning methods, however, are usually applied at the pixel level, or to some higher-level representations obtained after additional pre-processing of the raw images. Nevertheless, for the task of classifying different videos depending on the dynamics of their content, we assume that the relevant information is still preserved after the video has been encoded to be sent on a transmission channel.

In [6] we showed that SSIM can be compactly represented by means of polynomial curves that can be associated to each video. Tagged videos can then be handled by simple traffic shaping mechanisms in case of network congestion or under-provisioned network resources. The idea of representing the Q-R curve as a polynomial function is well known in the literature. For example, considering the distortion expressed as the PSNR, the Bjøntegaard model [47] approximates a Q-R curve by a third order logarithmic polynomial fitting, based on experimental observations [48]. Another polynomial fitting based on PSNR and MPEG-2 encoding is described in [49].

In this work, which builds on our preliminary position paper in [50], we therefore propose to automatically extract a set of features that can be used to describe the relevant characteristics of the original videos, using only information available at the network level. To the best of our knowledge, this is the first attempt to apply machine learning algorithms on this type of data for such a purpose.

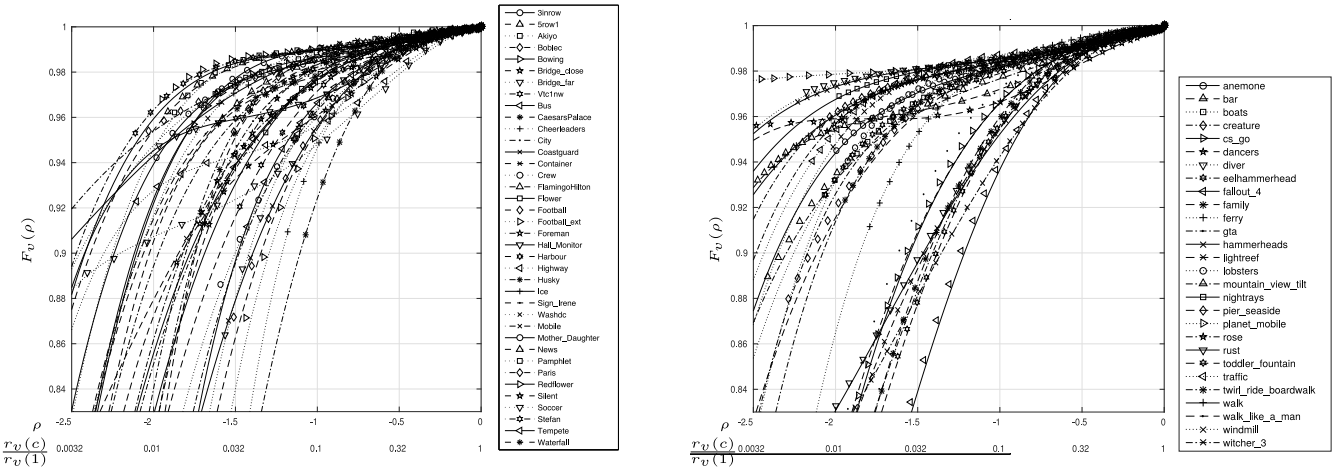
III. Video Analysis

In this study, we have expanded the video dataset used in [6] with a set of additional HD video clips. For the reader's convenience, we report here the video analysis framework described in [6].

We evaluate the objective QoE of the videos with the SSIM index, which is a full reference metric that measures the image degradation in terms of perceived structural information change, thus leveraging the tight inter-dependence between spatially close pixels that contain the information about the objects in the visual scene [8]. SSIM is calculated via statistical metrics (mean, variance) computed within a square window of size $N \times N$ (typically 8×8), which moves pixel-by-pixel over the entire image. The measure between the corresponding windows X and Y of two images is computed as follows:

$$SSIM(X, Y) = \frac{(2\mu_X\mu_Y + c_1)(2\sigma_{XY} + c_2)}{\left(\mu_X^2 + \mu_Y^2 + c_1\right)\left(\sigma_X^2 + \sigma_Y^2 + c_2\right)} \qquad (1)$$

where μ and σ^2 denote the mean and variance of the luminance value in the corresponding window, and c_1 and c_2 are



(a) CIF video set

(b) HD video set

Fig. 2. SSIM of the different video clips when varying the RSF: markers show empirical values, lines are obtained by the 4-degree polynomial approximation $F_v^{(4)}(\rho)$.

TABLE I
MAPPING SSIM TO MEAN OPINION SCORE SCALE

SSIM	MOS	Quality	Impairment
≥ 0.99	5	Excellent	Imperceptible
$[0.95, 0.99)$	4	Good	Perceptible but not annoying
$[0.88, 0.95)$	3	Fair	Slightly annoying
$[0.5, 0.88)$	2	Poor	Annoying
< 0.5	1	Bad	Very annoying

variables to stabilize the division with weak denominator (we refer the interested reader to [8] for additional details).

The range of the SSIM index goes from 0 to 1, which represent the extreme cases of totally different or perfectly identical frames, respectively. Tab. I shows the mapping of SSIM to Mean Opinion Score (MOS), which assesses the subjective perceived video quality on a scale of 5 values, from 1 (bad) to 5 (excellent), as reported in [51].

The analysis of the SSIM has been first applied to a pool of $V = 38$ CIF video clips, taken from standard reference sets.[2] Successively, we replicated the analysis on a set of 28 HD videos. Each video has been encoded into H.264-AVC format. To test the robustness of the proposed approach to the specific encoding algorithm, we used the Joint Scalable Video Model (JSVM) reference software [53] for CIF videos and the x264 encoder [54] for HD videos. The encoding has been done at $C = 18$ increasing compression levels (i.e., quantization points) for the CIF videos, and $C = 33$ levels for the HD videos, which correspond to as many quality levels. Note that there are no scene transitions inside each video sequence. The SSIM of a frame encoded at compression level c is obtained by comparing the decoded frame with the full quality version of the same frame. For practical reasons, we take the average values of the SSIM index computed for all frames of each video.

We denote by $r_v(c)$ the transmit rate of video $v \in \{1, \ldots, V\}$ encoded at rate $c \in \{1, \ldots, C\}$, with $r_v(1)$ being the maximum (i.e., full quality) rate. To ease the comparison between different video clips, it is convenient to normalize the video rates to the full quality rates. Moreover, following the Weber-Fechner's law that postulates a logarithmic relation between the intensity and the subjective perception of a stimulus, we introduce a logarithmic measure of the normalized rate, here named *Rate Scaling Factor* (RSF) and defined as

$$\rho_v(c) = \log(r_v(c)/r_v(1)). \tag{2}$$

The dynamics of the video content impact the perceived QoE for a certain RSF value, as clearly shown in Fig. 2 (on the next page) where markers correspond to the average SSIM of each video clip when varying the RSF ρ, while lines represent a 4-degree polynomial interpolation of such points. More generally, we observe that the SSIM characteristics of a video v can be approximated by an n-degree polynomial expression, which takes the form

$$F_v^{(n)}(\rho) \simeq 1 + a_{v,1}\rho + a_{v,2}\rho^2 + a_{v,3}\rho^3 + \ldots + a_{v,n}\rho^n. \tag{3}$$

The vector of coefficients $\mathbf{a}_v = \{a_{v,i}\}$, called *SSIM coefficients* in the following, provides a compact description of the relation between the perceived QoE and the RSF of a video v.

From Fig. 2, we observe that, in general, the 4-degree polynomial $F_v^{(4)}(\rho)$ provides a quite accurate approximation of the SSIM values in the range of ρ of practical interest, for both the CIF and the HD videos in the test set. We observe that the relationship between QoE and Quality of Service (QoS) parameters is, in general, very complex, depending (among other factors) on metrics such as GOP size/structure, frame-rate, resolution, etc. (see [55]). The curves reported in Fig. 2 have been obtained for a certain combination of parameters (frame rate, GOP structure, resolution), only changing the quality factor (c) of the H.264 encoder. Nonetheless, we obtained similar Q-R curves by changing the encoding parameters, i.e., considering different combinations of the GOP length and composition, frame rate, and resolution (not reported here for space constraints). In a real setting, most of these parameters will remain fixed within each video segment, so that the proposed approach is valid for each specific DASH

[2]Video traces can be found in [52], ftp://132.163.67.115/MM/cif.

request. It is hence conceivable to tag each video segment with the SSIM coefficients which provide a compact representation of its QoE characteristics that, in turn, can be used by RM and VAC algorithms, as discussed in the next section.

IV. Machine Learning Approach to Video Classification

The exact SSIM characterization of a video sequence using (1) is computationally demanding and infeasible in many practical cases. Following the rationale described in [1] and [56], to overcome this limitation we propose a machine learning approach that provides a fairly accurate estimation of the SSIM characteristics of a video from the *size* of the frames coded in a GOP. As previously mentioned, we postulate that the SSIM characteristics of a video are closely related to the dynamics of its content, and that this information is preserved in the structure of the corresponding sequence of frame sizes after the encoding. However, extracting the SSIM characteristics of a video directly from the raw data, i.e., the frame sizes, is problematic because of the non-linear and hidden interrelations between the two quantities.

The fundamental idea behind our approach is to learn a generative model to capture these non-linearities, providing an alternative representation of the input data that is amenable to classification even by means of linear discrimination methods. More specifically, our learning framework consists of two main phases. First, *unsupervised learning* is used to extract an abstract representation of the raw data that captures descriptive features of the video. A subsequent *supervised learning* phase is then performed to create a mapping between the abstract representations and the corresponding SSIM coefficients of the related videos. These two learning phases are detailed in the following.

A. Unsupervised Phase: The Restricted Boltzmann Machine

Our approach relies on a powerful family of generative models which can be implemented as stochastic recurrent neural networks known as Boltzmann Machines [57]. They can be interpreted as probabilistic graphical models, where connections between units are symmetric, i.e., with equal weight in either direction. The input to the network is given through a layer of visible (i.e., observed) units, which are fully connected to another layer of hidden units that are used to model the latent features of the data. If there are no connections among units of the same layer, we obtain the so-called Restricted Boltzmann Machine (RBM) [7], which is graphically represented in Fig. 3.

The behavior of the network is driven by an energy function E, which implicitly defines the joint distribution of the units by assigning a probability value to each of their possible configurations:

$$p(v, h) = \frac{e^{-E(v,h)}}{Z} \tag{4}$$

where v and h are column vectors containing the values of the visible and hidden units, respectively, and Z is a normalizing factor known as partition function. The energy function

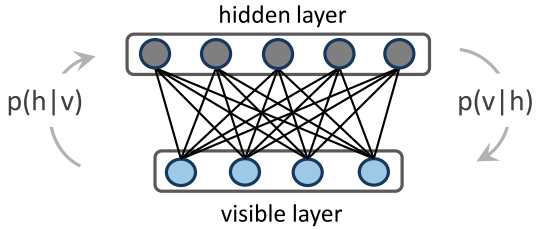


Fig. 3. Graphical representation of a Restricted Boltzmann Machine.

is parameterized according to the weights of the connections between visible and hidden units:

$$E(v, h) = -b^\top v - c^\top h - h^\top W v, \tag{5}$$

where W is the matrix of connections weights and b and c are two additional parameters known as unit biases.

RBMs can be efficiently trained by using the contrastive divergence algorithm [58], which consists in alternating a positive and a negative phase. During the positive phase (*inference*), visible units are clamped to the values of the data observed in the training set. The network then propagates activations to hidden units, according to the weights of the connections. If we consider binary units for simplicity, during the positive phase the network observes the values of the visible units and activates each hidden unit h_j according to the conditional probability:

$$p(h_j = 1|v) = \sigma\left(c_j + \sum_i v_i w_{ij}\right), \tag{6}$$

where σ is the sigmoid logistic function, c_j is the bias term of the hidden unit h_j, and w_{ij} is the weight of its connection with the visible unit v_i. The entire vector of hidden unit activations constitutes an *internal representation* of the pattern observed in the visible units. During the negative phase, instead, hidden units are fixed and activations are propagated backward to the visible units in a similar fashion, in order to accurately *reconstruct* the original input vector. Each visible unit v_i is therefore activated according to the conditional probability:

$$p(v_i = 1|h) = \sigma\left(b_i + \sum_j h_j w_{ij}\right). \tag{7}$$

The objective of the learning process is to find a good set of weights W, so that the function E will assign low energy (i.e., high probability) to configurations of units that allow to obtain accurate reconstructions of the input patterns (i.e., maximum likelihood learning). This can be accomplished by performing gradient descent over the likelihood function of the training data. It turns out that the derivative of the log-probability of a training vector v with respect to a particular weight w_{ij} is surprisingly simple:

$$\frac{\partial \log p(v)}{\partial w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}, \tag{8}$$

where the first term on the right-hand side of (8) represents the empirical expectations computed on the training data, while the second term refers to the expectations computed according

to the actual model distribution. We can use this quantity to compute how each weight should be changed at each learning step:

$$\Delta w_{ij} = \eta\big(\langle v_i h_j\rangle_{data} - \langle v_i h_j\rangle_{model}\big) \tag{9}$$

where η is a small constant called learning rate. Due to the stochastic dynamics of RBMs, computing model expectations requires to gradually change the state of the network until it settles to *thermal equilibrium*, usually by running computationally expensive Gibbs sampling algorithms [59]. However, contrastive divergence makes it possible to efficiently train large-scale RBMs by approximating the log-probability gradient. The reader could refer to [60] and [61] for more details about learning in RBMs and for the explanation of important additional parameters of the algorithm (e.g., weight decay and momentum).

In our case, the training set consists of vectors of frame sizes for each GOP of the videos in the dataset. Unsupervised learning tunes the RBM model parameters (i.e., the connections weights) with the objective of reproducing the patterns presented in the visible layer, thereby minimizing the reconstruction error. At the beginning, weights are randomly initialized to small values (close to zero) and the reconstruction will be very poor. However, the learning process iteratively adapts the weights until the network is able to accurately reproduce the observed patterns. At the end of this unsupervised learning phase, the values taken by the units in the hidden layer provide an alternative and, hopefully, more expressive representation of the input vector, i.e., of a certain sequence of frame sizes in a GOP.

B. Supervised Phase: The Linear Classifier

After a good model of the data has been learned, an additional *read-out* module can be put on top of the hidden layer of the RBM to perform a supervised classification task, which in our case consists in estimating the SSIM coefficients $\mathbf{a}_v$ for each new GOP. The idea is that some characteristics of the data are not directly visible in the raw input patterns, but can be discovered by the feature extraction process during the unsupervised learning phase. Once the RBM has learned good internal representations of the patterns by modeling their underlying causes, it should be easier to perform a supervised classification task starting from those abstract representations.

We use a simple linear classifier as read-out module. The discrimination between the possible classes is therefore performed by exploiting a linear combination of the data features. This choice is motivated by observing that the non-linearities of the data should be captured by the generative model during the unsupervised learning phase, which creates more separable representations that could be easily read out even by a linear method. In many machine learning scenarios, this strategy has shown to be very effective and is usually adopted by the so-called "kernel methods" exploited in *Support Vector Machines* [62], which first perform a non-linear projection of the data into a different (usually higher-dimensional) feature space, and then apply a linear optimization method to compute the maximum margin separating hyperplane.

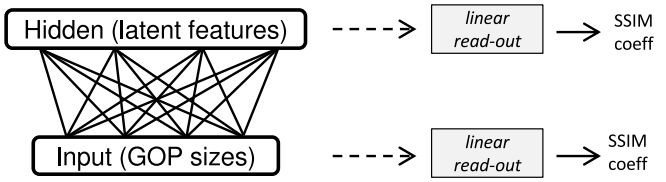


Fig. 4. Scheme of the proposed learning framework, on which unsupervised feature extraction (left) is followed by supervised linear read-out (right).

Within this perspective, the accuracy of linear read-out can be considered as a coarse measure of how well the relevant features of the data are explicitly captured by the generative model [60]. Therefore, the use of a linear classifier makes it easier to understand the quality of the internal representations learned by the RBM, because we can directly compare the classification accuracy obtained using the raw input patterns with that obtained from the internal representations of the RBM. Moreover, a linear classifier is preferred in our case due to its greater generalization ability even in the presence of a limited training set. Indeed, a more powerful, non-linear algorithm would be more prone to overfitting. A schematic representation of this process is given in Fig. 4.

It is worth remarking that, once the unsupervised training phase is completed, the internal representation of the input data provided by the RBM can be used to perform supervised training of multiple read-out modules, with different purposes. For instance, it is possible to train a linear classifier that recognizes the GOPs belonging to the same video, or that classifies the GOPs according to the similarity of the video dynamics, and so on [56]. This is indeed one of the major advantages of combining unsupervised and supervised learning approaches, with the former providing an alternative representation of the input data that eases the selection of useful features by the latter.

V. Learning Framework Performance

In this section we evaluate the performance of the proposed RBM-based learning framework with respect to a linear classifier that acts directly on the raw data, i.e., the frame sizes contained in a GOP.

A. Dataset and Learning Parameters

The system is tested on the video dataset described in Section III. In order to make the size of the data uniform, we considered the first 15 GOPs of each CIF video, and 13 GOPs for HD videos, thereby discarding shorter videos. Thus, we used 34 CIF videos, for a total of 510 data patterns (GOPs), and 28 HD videos, for a total of 364 data patterns. The quality of learning in a RBM gets worse when the patterns in the training set are drawn from very different, heterogeneous distributions. In particular, in our case we observed that by merging the GOP patterns corresponding to both CIF videos and HD videos resulted in the emergence of a much less effective set of features. The reason is that the sole frame size is likely insufficient to capture the complex Q-R relationships for generic encoding parameters, while it is sufficient when

the other parameters (namely, the GOP structure and size, and the video resolution) are fixed. A possible solution to overcome this problem is to train a different learning model for each representative combination of video encoding parameters. Another possibility may consist in expanding the input patterns by also explicitly including some information about the encoding parameters, such as the resolution of the video or the GOP structure. In this work we considered the first solution, leaving the latter for future studies.

We therefore created two different training sets, one containing samples derived from CIF videos and the other containing samples derived from HD videos, and separate RBMs were trained on each dataset. The encoding format for input patterns consisted of GOPs formed by a single inter-coded frame (I) followed by 15 predicted frames (P), which is a common format for GOPs of 16 frames. However, control simulations (not reported here) showed that our approach still works if we consider other GOP formats, e.g., with a different number of frames and/or a different pattern (sequence of I and P frames within a GOP), provided that the RBM is adapted to the new input and properly trained.

Due to the limited size of the datasets, we tested the performance of the system using a *k-fold cross-validation* technique [63]. To this aim, we partitioned the dataset of CIF videos into 34 subsets (folds), each including all the 15 GOPs of a specific video. The RBM was then trained using 33 folds (training set), and its generalization performance was computed on the left-out fold (test set). This way, 34 different RBMs were trained, each time changing the left-out video to be used as test, and we report the mean estimation accuracy over all the 15 GOPs. The input to the RBM consisted of 32 visible units, which represented the sizes of the 16 frames in a GOP, coded with two different compression levels $c = 1$ (full quality) and $c = 9$ (intermediate quality). We only included these two levels in order to limit the amount of patterns in the training set, with the goal of more clearly establishing how well the system could generalize to previously unseen compression levels. Furthermore, we did not consider the lowest qualities in place of the intermediate one because we aimed at estimating with greater accuracy the high SSIM region of the Q-R curves rather than the low-quality tail, considering that in practical applications the latter region is of scarce interest because of the very poor visual quality of the videos.

The same procedure was applied for the dataset of 28 HD videos clips, where however the intermediate quality corresponded to a parameter $c = 18$, since the number of available quality layers for HD videos was 33, against the 18 levels of the CIF videos.

The I and P frame sizes of each GOP were normalized between 0 and 1, which corresponded to the minimum and maximum frame sizes, as this is the usual format of the input patterns used for training neural networks. The size of the hidden layer determines the complexity of the generative model, since the number of free parameters in the model is given by the number of connection weights. We tested different layer sizes, with a number of units varying between 50 and 200, finding that our results are robust with respect to this parameter. Results presented in the following have been obtained with a network having 70 hidden units.

We use a publicly available efficient implementation of RBMs that exploits Graphic Processing Units (GPUs) to parallelize the learning algorithm [64]. Unsupervised learning occurred using a mini-batch scheme with mini-batch size of 13, learning rate of 0.001, weight decay of 0.00001, and a momentum coefficient of 0.9. With the current settings of the machine learning parameters, the learning phase converged after about 50 epochs without exceeding one minute of running time. Regarding the supervised phase, a linear classifier can be implemented as a single layer perceptron, on which iterative learning was performed using the delta-rule. We used an equivalent but computationally more efficient method, which relies on the calculation of a pseudo-inverse matrix and is readily available in some high-level programming languages such as Python or MATLAB [60].

We remark that the unsupervised and supervised learning processes are performed only once. Once the RBM and the coupled linear classifier are trained, the estimation of the SSIM coefficients for unknown videos is extremely simple, and can be performed online in negligible time.

B. Coefficients Estimation Accuracy

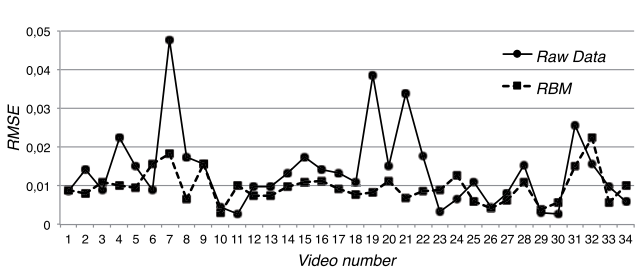
We assessed whether the internal representation learned by the RBM allowed to estimate the n SSIM coefficients for each video in the test set. The quality of the estimation was evaluated in terms of Root Mean Square Error (RMSE) between the exact and the estimated SSIM-rate characteristics, i.e.,

$$\mathrm{RMSE} = \sqrt{\frac{1}{\rho_{min}} \int_{\rho_{min}}^{0} \left(F_v^{(4)}(\rho) - \tilde{F}_v^{(n)}(\rho) \right)^2 d\rho}$$

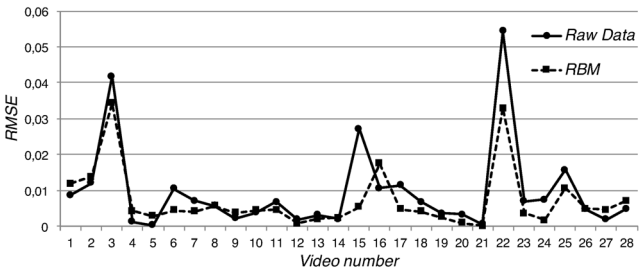
where $\rho_{min} \simeq -3$ is the minimum value of RSF of interest, while $F_v^{(4)}(\rho)$ is the reference SSIM-rate curve, and $\tilde{F}_v^{(n)}(\rho)$ is the n-degree polynomial (3), with coefficients estimated by the classifier.

The dashed line with square markers in Fig. 5 shows the mean estimation accuracy on the 15 GOPs contained in each of the 34 videos of the CIF test set (a), and on the 13 GOPs of the 28 videos in the HD test set (b). To better appreciate the performance of the RBM-based learning architecture, we also report the RMSE for the SSIM curves obtained by applying the linear classifier directly on the raw data patterns (solid line with circle markers). We see that the internal representation learned by the RBM model is indeed capable of capturing critical features of the data, thereby allowing to increase the estimation accuracy for almost all test videos.

Fig. 6 offers a visual comparison between the exact and estimated SSIM curves for two different videos with prototypical trends (see corresponding points in Fig. 5 to have an idea of their average RMSE error). In particular, Fig. 6(a) shows that the curve estimated using the RBM internal representations (solid line) clearly exhibits a better alignment with the exact SSIM curve (dashed line) than the curve obtained directly from raw data (dotted line). Even in the few cases where the RMSE is worse for RBM prediction, as that reported in Fig. 6(b), the RBM estimation of the SSIM curve still remains good.

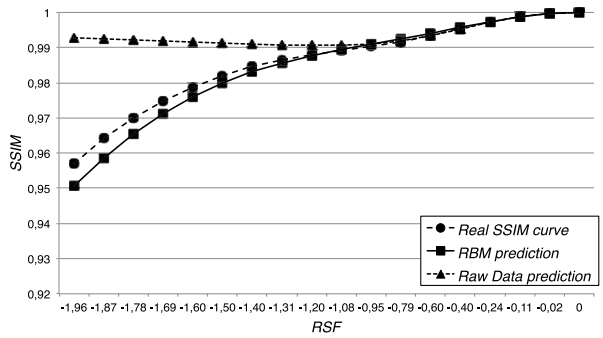


(a) CIF video set

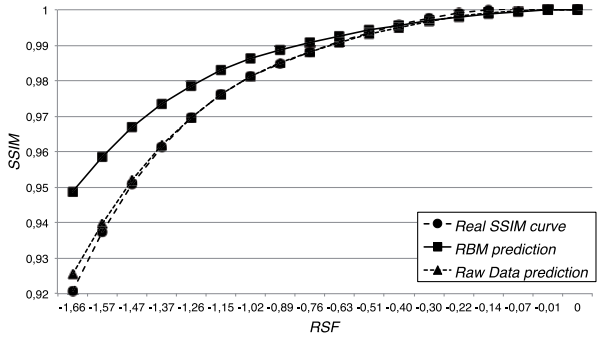


(b) HD video set

Fig. 5. Root Mean Squared Error (RMSE) of the estimated SSIM-rate curve for each video in the CIF video test set (a) and HD video test set (b), with $n = 4$. Polynomial coefficients estimation is given by applying a linear classifier on raw input data (circle markers) or on the hidden layer of the RBM (square markers).



(a) Predicted and real curves for video number 2: RBM prediction shows better precision.



(b) Predicted and real curves for video number 11: raw data prediction shows better precision, but RBM prediction is still acceptable.

Fig. 6. Examples of predicted polynomial curves with respect to the ideal curve, for two different videos.

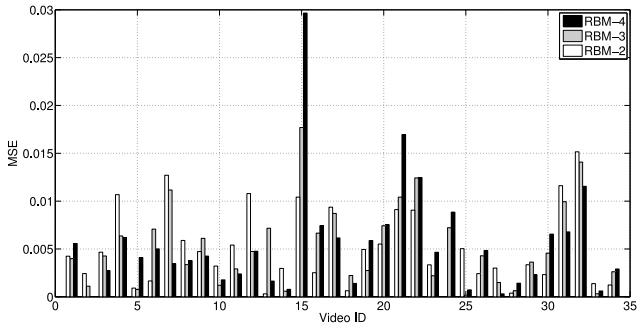


Fig. 7. 2, 3 and 4-degree prediction error for each video of the dataset.

As explained, the complexity of the coefficients estimation increases with the degree n of the polynomial. On the other hand, high-degree polynomials offer a better approximation of the actual SSIM-rate characteristics. It is therefore interesting to investigate the accuracy of the SSIM estimation when varying the degree n of the polynomial. To this end, for each video in the CIF dataset, we report in Fig. 7 the RMSE of the SSIM estimation obtained by considering 2, 3 and 4-degree polynomials. Similar results were obtained for HD videos. Quite interestingly, we observe that there is no absolute winner: the optimal choice of n depends on the characteristics of each video. In the next section, we will investigate the practical impact of such estimation differences in the performance of video admission control and resource allocation algorithms.

VI. Performance Analysis of Cognitive RM and VAC Algorithms

In this section, we first revisit the approach presented in [6], which in this paper is used in conjunction with the learning framework of Section V. Then, we discuss the role of the play-out buffer and derive a simple analysis to determine the amount of pre-buffered content that guarantees a freezing probability lower than a given threshold.

A. SSIM-Based RM and VAC Algorithms

Given a mechanism to infer the QoE characteristics of a video, we develop VAC and RM mechanisms that can make use of such information. We consider a framework where different video clips are multiplexed into a shared link of capacity R by the Cognitive HTTP Proxy (CHP) that performs VAC and RM (see Fig. 1). In general, the RM module should detect changes of the link capacity (e.g., due to concurrent data flows or fading phenomena in wireless channels) and trigger an optimization procedure that adapts the video rates to maximize a certain utility function. In this work, we consider a more favorable (but still practical) scenario, in which a fixed and constant capacity is reserved to video flows, which are then isolated from best-effort traffic. In Section VIII we will discuss possible extensions of the work to more challenging scenarios.

The VAC module determines whether or not a new video request can be accepted without decreasing the QoE of any

video below a threshold F^* negotiated, for instance, between the operator and video consumers. To this end, the VAC invokes the RM module to get the best resource allocation for all the videos potentially admitted into the system and, then, computes the expected SSIM of each video by using (3). If the estimated SSIM is below F^* the last video admission request is refused, otherwise the video is accepted and the rates of the videos in the system are adapted to the new allocation of the transmission resources determined by the RM module. To avoid sharp quality changes in the ongoing video streams, the video rates can be adapted progressively, with a step that depends on the actual gap between the current and the target SSIM of each video. Such smoothing techniques will be briefly discussed in Section VIII, though a detailed analysis of these and other possible improvements is left to future work.

Formally, let R denote the average available transmission capacity of the link that can be allotted to the videos, and let $\Gamma = \{\gamma_v\}$ be an allocation vector that assigns to the vth video a fraction γ_v of R, with $\gamma_v = 0$ indicating that the video is not accepted into the system. Although the H.264 encoding can only offer a discrete set of transmit rates, in the formulation of the optimization problem we temporarily assume that video encoding rates can be tuned in a continuous manner.[3] Under this assumption, the RSF of the vth video can be expressed as

$$\tilde{\rho}_v = \log\left(\frac{\gamma_v R}{r_v(1)}\right). \tag{10}$$

The optimization problem addressed by the RM module can then be defined as follows:

$$\Gamma_{\text{opt}} = \arg\max_{\Gamma} U(\Gamma, R, \{F_v\}) \quad \text{s.t.} \quad \sum_v \gamma_v \le 1, \tag{11}$$

where $\{F_v\}$ denotes the set of SSIM functions of the videos, while $U(\cdot)$ denotes the *utility function* considered by the optimization algorithm. We consider two baseline utility functions that reflect different optimization purposes.

Rate Fairness (RF): Resources are distributed to all active videos proportionally to their full quality rate, without considering the impact on the perceived QoE. In this case, the optimal rate allocation for the ith video is simply given by

$$\gamma_{\text{opt},v} = \frac{r_v(1)}{\sum_j r_j(1)}, \tag{12}$$

so that the RSF of each video equals $\tilde{\rho} = \log(R/\sum_j r_j(1))$.

SSIM Fairness (SF): Resources are allocated according to a max-min fairness criterion with respect to the SSIM of the different videos:

$$U(\Gamma, R, \{F_v\}) = \min_v F_v(\tilde{\rho}_v). \tag{13}$$

Note that under the assumption of continuous rate adaptation, the SF criterion yields the same SSIM, say φ, to all active videos. Given this target SSIM, the RSF for each video can be easily found as $\tilde{\rho}_v = F_v^{-1}(\varphi)$, where F_v^{-1} is the inverse of the QoE function F_v (which is monotonic in the range of interest). Therefore, the optimization problem can be solved by searching for the maximum φ that satisfies the rate constraint in (11), i.e.,

$$\varphi^* = \max\left\{\varphi : \frac{1}{R}\sum_v r_v(1) 10^{F_v^{-1}(\varphi)} \le 1\right\}. \tag{14}$$

and the associated rate-allocation vector is given by

$$\gamma_v = 10^{F_v^{-1}(\varphi^*)}\frac{r_v(1)}{R} \quad \text{for all } v \in V. \tag{15}$$

Mapping to Admissible Encoding Rates: Once the target allocation vector $\Gamma = \{\gamma_v\}$ has been determined under the assumption of continuously encoding rates, we need to find a feasible allocation vector $\Gamma^\circ = \{\gamma_v^\circ\}$ such that, for each video v, there exists an encoding rate $r_v(c) = \gamma_v^\circ R$. The solution is obtained through the following recursive policy. For each video v, we find the minimum compression level $\hat{c}$ for which the encoding rate does not exceed the allotted capacity, i.e.,

$$\hat{c} = \min\{c : r_v(c) \le \gamma_v R\}.$$

We then select the video v for which the gap between $r_v(\hat{c})$ and $\gamma_v R$ is minimum, and set $\gamma_v^\circ = r_v(\hat{c})/R$. Hence, we update the amount of available resources as $R \leftarrow R - r_v(\hat{c})$ and repeat the process iteratively over the remaining videos.

B. Play-Out Buffer Analysis

We observe that the considered RM algorithms always guarantee that the aggregate bitrate of the downloaded video segments does not exceed the available channel capacity. Consequently, the *size*[4] of the play-out buffer at the client side will also remain approximately constant in time, except for small oscillations due to the variations of the GOP rates around their mean, which can be smoothed out by buffering a few GOPs of video before starting the playback. In this way, it is possible to avoid freezing events, while guaranteeing quick starting of the video play. In the following, we propose an approximate analysis of the play-out buffer size that guarantees a smooth video playback with low probability of freezing and rebuffering events.

Let τ_v be the time duration of each GOP in the video sequence v. Furthermore, let $s_v^h(c)$ be the size of the hth GOP of the video, when encoded at compression level c. In principle, these values can be determined by the video server and passed to the client (and the CHP) through the MPD descriptor. However, for the sake of simplicity and generality, we model these values as independent and identically distributed random variables, with mean $s_v(c) = \mathrm{E}[s_v^h(c)]$ and standard deviation $\sigma_v(c)$, and we assume that only these two parameters are passed to the client/CHP.

Let n_0 be the number of GOPs that are buffered by the client before starting the playback. When the playback starts, a GOP is fetched from the buffer every τ_v seconds, while new GOPs arrive into the buffer from the network at uneven intervals. A freezing event occurs whenever the time to download n new GOPs exceeds the time to play n_0+n GOPs or, in other terms,

[3]This assumption will be removed in the simulations.

[4]As customary, the size of the play-out buffer is here intended in terms of playing time of the buffered video content, whose size in bytes depends on the compression level of the video sequence.

when the aggregate size of n GOPs, $S_v(n;c)$, exceeds the total number of bits $D_v(n)$ that can be downloaded by the client in the period $(n+n_0)\tau_v$. Assuming that the RM determines the source rates by conservatively considering only a fraction $\alpha \in [0,1]$ of the available link rate R, we have that $s_v(c) = \alpha\tau_v\gamma_v^\circ R$, so that the aggregate size of the n GOPs is $S_v(n;c) = \sum_{h=1}^{n} s_v^h(c)$, with mean $\mu = ns_v(c) = n\alpha\tau_v\gamma_v^\circ R$, while the total amount of data that can be downloaded in the playing time of $n+n_0$ GOPs is $D_v(n) = \tau_v\gamma_v^\circ R(n_0+n)$. The freezing probability can then be expressed as $P_f(n;c) = \Pr[S_v(n;c) \geq D_v(n)] = \Pr[S_v(n;c) \geq \mu(1+\delta)]$, where $\delta = \frac{n+n_0}{n\alpha} - 1$. We wish to determine the value of n_0 such that $P_f(n;c) \leq P_f^*$ for all n, where P_f^* is the maximum acceptable freezing probability. Applying the Chernoff bound, we then get

$$P_f(n;c) \leq \exp\left(-\frac{2\delta^2\mu^2}{n\Delta_v(c)^2}\right), \tag{16}$$

where $\Delta_v(c)$ is the difference between the max and the min GOP sizes. Posing the right-hand side of (16) lower than or equal to P_f^* we get the following conservative criterion to choose the size of the play-out buffer:

$$\begin{aligned} n_0 \geq f_0(n;\alpha) &= \frac{\alpha\Delta_v(c)}{s_v(c)}\sqrt{n\log\left(\frac{1}{\sqrt{P_f^*}}\right)} - n(1-\alpha) \\ &= \beta\sqrt{n} - (1-\alpha)n, \end{aligned} \tag{17}$$

where, for ease of writing, we set

$$\beta = \frac{\Delta_v(c)}{\tau_v\gamma_v R}\sqrt{\log\left(\frac{1}{\sqrt{P_f^*}}\right)}. \tag{18}$$

The right-hand side of (17) reaches it maximum for $n^* = \frac{\beta^2}{4(1-\alpha)^2}$, for which we get $f_0(n^*;\alpha) = \frac{\beta^2}{4(1-\alpha)}$. Denoting by $n_{\max}$ the maximum number of GOPs in a video stream, we can then set

$$n_0 = \beta\sqrt{\min\{n_{\max}, n^*\}} - (1-\alpha)\min\{n_{\max}, n^*\}. \tag{19}$$

Using this approximation, it is possible to tune the play-out buffer size to the characteristics of the specific video stream. Note that, the smaller α (i.e., the larger the fraction of the link rate that is not allocated to the sources to leave some capacity in case of need), the smaller the play-out buffer required to avoid freezing events. However, the value of c will also be affected by α, since the RM will choose more compressed versions of the video streams to fit into the shrunk channel capacity αR. For a given P_f^*, there is then a tradeoff between the delay to start the play out, which is approximately equal to $\alpha n_0\tau_v$, and the quality of the streamed video.

Considering the test videos used in this study, by setting $\alpha = 1$ (which allows for maximum video quality), we obtained $\Delta_v(c)/s_v(c) \leq 0.35$ for all videos and all values of c. With such values, eq. (19) returns a buffer size of $n_0 \simeq 10$ GOPs (about 3.6 seconds with GOP of 12 frames) when considering video sequences of up to $n_{\max} = 500$ GOPs (about 3 minutes) and a freezing probability threshold $P_f^* = 5\%$, while $n_0 = 12$ GOPs for $P_f^* = 1\%$.

VII. Simulation Results

Here we present the results of our simulation study, which show the potential benefits, in terms of QoE and blocking probability of the video connections, that can be achieved by adopting the proposed mechanisms.

A. Simulation Scenario

To compare the performance of the VAC and RM algorithms described above, we simulate a scenario where a transmission link is shared among the users, e.g., the outbound link towards the public Internet of a LAN. The VAC mechanism (running in the edge router/proxy) intercepts all requests for new video streaming sessions, and checks whether the additional traffic flow can be accommodated without dropping the QoE of the active videos below a certain SSIM threshold that we set to $F^* = 0.95$, which corresponds to good quality (MOS of 4, see Tab. I).

The video generation process is simulated as a Poisson process with $\lambda = 0.66$ requests/s, where each video request refers to a video randomly picked from the dataset. The simulation provides a high-level picture of the system, neglecting the low-level details of the HTTP protocol. Each new video request triggers the VAC and RM modules, which use the Q-R curve for that video as estimated by the RBM algorithm to perform their decisions. When a new video is admitted into the system, or an active video completes its playback, the RM algorithm reassigns the resources, according to the chosen policy. Note that, while the VAC and RM operate on the *estimated* Q-R curves, the performance shown in the result section refers to the *actual* SSIM of the active videos. Denoting by T the average duration of a video sequence, we then have an offered load of $\lambda T \simeq 11$ videos, which corresponds to an aggregate rate request for full video quality of about $G \simeq 161$ Mb/s.

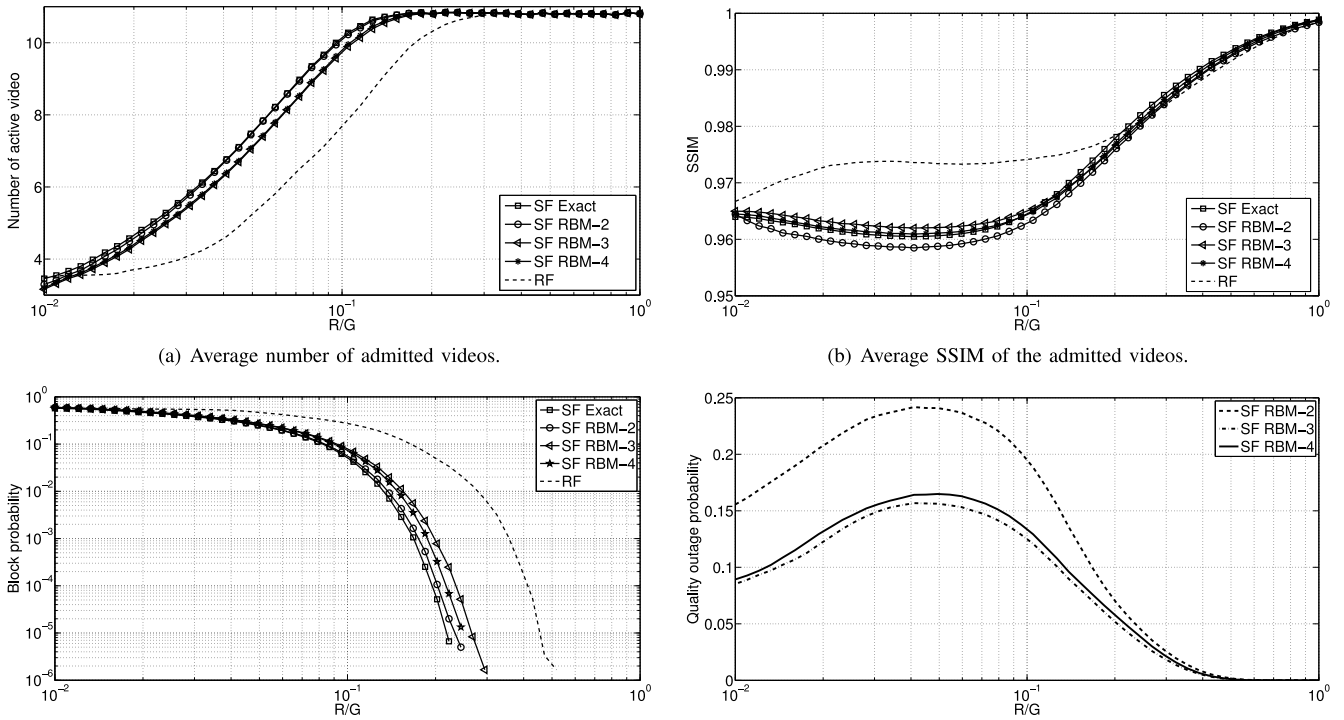
Video requests are processed by the VAC algorithms described in Section VI, and resources are allocated accordingly. In particular, we consider four different flavors of the SF algorithm, corresponding to different choices of the SSIM function $F_v(\rho)$, namely:

- *SF-Exact* based on the exact SSIM curve, i.e., $F_v(\rho) = F_v^{(4)}(\rho)$;
- *SF-RBM-n* based on the n-degree polynomial estimation given by the RBM model, i.e., $F_v(\rho) = \tilde{F}_v^{(n)}(\rho)$, with $n \in \{2,3,4\}$.

The simulation has been implemented using MATLAB, without the use of external libraries. We considered a practical, but somehow favorable scenario, where the link capacity is stable and known and the Q-R characteristic of each video is fixed in time.

B. Results

We compare the algorithms in terms of: (i) average number of admitted videos, (ii) average SSIM of admitted videos, (iii) blocking probability of a video request, and (iv) quality outage probability, i.e., probability that the quality of an accepted video drops below the minimum threshold F^* during the session. Note that with SF-Exact there is no



(a) Average number of admitted videos.

(b) Average SSIM of the admitted videos.

(c) Probability that a video request is blocked.

(d) Probability that the video quality is below threshold at least once during the video session.

Fig. 8. Performance comparison of our proposed algorithms *RF* and *SF* when varying the channel capacity, where *SF-Exact* is the result based on the exact SSIM curve, while *SF-RBM-n* is based on the *n*-degree polynomial estimation given by the RBM model.

quality outage, therefore this performance index captures the impact of the SSIM estimation errors of the RBM-based methods.

Fig. 8 shows the performance indices when varying the channel rate R with respect to the nominal average rate request G for full-quality videos. At first glance, we observe that the SF policies always perform better than RF, and accept more videos with above-threshold quality. This confirms that content-aware admission and resource allocation policies are much more effective than traditional content-agnostic policies in a QoE framework. It is interesting to observe in Fig. 8(b) that the average SSIM of the active videos is well above the minimum required quality threshold F^*. The reason is that we considered the actual video rates obtained with the different compression levels, so that resource allocation is not able to use all the channel capacity, leaving part of it unused. This effect is minimized when $R/G \simeq 0.05$. If the video coder were able to provide any desired bitrate value, the quality for all video would have been equal to F^*, when considering a sufficiently large G. From Fig. 8(d) we also note that the smaller the margin between the mean SSIM and F^*, the larger the quality outage probability of the SF-RBM schemes. Having a smaller margin, in fact, offers less protection to SSIM estimation errors. When the average SSIM is way larger than F^*, instead, the probability that a SSIM estimation error causes the actual video quality to drop below F^* is very low.

For what concerns the SF algorithms, we observe in Fig. 8(a) that, on average, the SF-RBM polynomial approximations perform quite closely to the SF-Exact scheme. Hence, the RBM-based prediction is nearly optimal and proves the goodness of the training phase. A closer look at the results reveals that SF-RBM-2 is slightly looser than the other SF schemes in the admission process, allowing a moderately larger number of videos in the system, with a little lower average SSIM, as shown in Fig. 8(b). From Fig. 8(d), however, we note that the degree-2 approximation exhibits the largest quality outage probability, which negatively impacts the system performance due to the aforementioned nearly optimal number of admitted videos. Conversely, the SF-RMB-3 and SF-RMB-4 schemes perform in a comparable manner, with a very small advantage of SF-RBM-3 over SF-RBM-4 in terms of quality outage probability. Thus, we might suggest the use of degree-3 predictions due to the slightly lower computational complexity and amount of signaling required in the system.

VIII. Improvements and Open Challenges

The study presented in the previous sections was mainly intended to prove the effectiveness of the machine-learning approach to gain knowledge on the Q-R characteristics of a video sequence from high-layer parameters and to show how such a knowledge can be exploited by network management algorithms to improve the service offered to the users. The analysis has been carried out by considering a practical, but somehow favorable scenario, in which we assumed homogeneous video sequences, with fixed and known Q-R characteristics and stable communication resources. Furthermore, we

neglected other important QoE metrics, such as the effect of sharp quality variations.

In this section we provide a preliminary discussion of some possible extensions of the proposed approach to overcome these limits, leaving a more detailed analysis to future work. Given its superior performance, we only consider the SSIM-fairness RM criterion. As a first step, we relax some of the assumptions regarding the Q-R characteristics of the video sequences and the QoE metrics, by still assuming that the multimedia flows are guaranteed a constant bitrate R. Then, we address the case where the channel capacity may vary over time.

A. Limiting Video Quality Variations

To avoid sharp variations of the video quality due to the adaptation mechanisms, it is possible to resort to the smoothing/hysteresis techniques proposed in the DASH literature. However, the knowledge of the Q-R characteristics of each video sequence makes it possible to choose the step of the rate adaptation in a way that makes the quality variation less perceivable. Consider, for example, the reduction of the SSIM of current videos from φ to φ' to make space for a newcomer. If the quality variation $\varphi - \varphi'$ is small, so that the SSIM gap is barely perceivable, then the rate change can be performed immediately, irrespective of the actual rate gap, and the new video can be directly admitted with quality φ'. If, instead, the SSIM gap is perceivable, then the rates should be smoothly changed and the new video might be admitted with some delay and/or with a lower initial quality which is progressively and smoothly increased till φ'. We observe that the proper implementation of these mechanisms would require the definition of a function $d(\varphi, \varphi', t)$ that quantifies the quality degradation due to variations of the SSIM from φ to φ' in a time t. To the best of our knowledge, the identification of such a function is still an open and interesting research challenge.

B. Varying Q-R Characteristics

The video clips considered in our analysis were homogeneous in terms of Q-R characteristics. In general, however, the Q-R curve may vary in consecutive video segments, e.g., because of scene changes. In this case, the VAC becomes more complex. If the Q-R curve is known in advance for all the video segments, the VAC can potentially predict the resource assignments for the whole duration of the video sequences (assuming the current system conditions would not further change) and check whether the SSIM would always be satisfactory. Moreover, it is possible to design rate adaptation algorithms that temporarily increase the resource share assigned to a flow (or reduce the video quality of that flow) in order to fill the play-out buffer in prevision of future segments of the same video with higher rate requests.

To formalize these concepts, we can define $g_v^\ell(\varphi)$ as the size of the ℓ segment of video v, when encoded at a level that yields SSIM φ. Adopting a conservative approach, we may replace the feasibility condition in (14) with the following

$$\frac{1}{n_s}\sum_{\ell=1}^{n_s}\sum_{v} g_v^\ell(\varphi) \le RT_s, \quad \text{for } n_s = 1, 2, \ldots, N_{s,\max}, \tag{20}$$

where T_s is the time duration of a video segment, and $N_{s,\max}$ is an acceptable time horizon (e.g., the least number of residual segments for the ongoing flows). Therefore, (20) is satisfied when the aggregate bitrate required to download each of the video segments at quality φ never exceeds the link capacity. A new video is accepted into the system only if the maximum φ that satisfies (20) is not lower than the threshold F^*. A more aggressive (and resource-efficient) strategy may consider a dynamic adaptation of φ, while avoiding sharp quality variations. In this case, the feasibility condition can be expressed as

$$\begin{aligned}&\frac{1}{n_s}\sum_{\ell=1}^{n_s}\sum_{v} g_v^\ell(\varphi_\ell) \le RT_s, \quad \text{for } n_s = 1, 2, \ldots, N_{s,\max},\\ &\text{s.t. } d(\varphi_\ell, \varphi_{\ell+1}, T_s) \le d^*.\end{aligned}$$

where $d(\cdot)$ is the function described in Section VIII-A, and d^* is the maximum acceptable degradation due to quality variations. The analysis of these approaches, however, is left for future work.

C. Variable Link Capacity

The analysis carried out so far assumes that the link capacity reserved to multimedia flows is constant over time. In many practical cases, however, the multimedia contents share the channel with other flows, so that the capacity available to video flows may vary in time. In this case, the RM algorithm should be able to estimate the new available rate and adapt the quality of the on-going flows accordingly. Since the capacity estimate is generally noisy, however, it is not possible to guarantee a minimum SSIM, or to completely avoid the risk of freezing or sharp quality variations.

To gain insights on the possible effects of noisy channel estimates, we model the link rate experienced when downloading the hth GOP of video v as $r_v^h = r_v(c) + w_v^h$, where $r_v(c)$ is the link capacity estimated by the RM algorithm and w_v^h is an estimate error term, which we assume to be random, with zero mean and variance $\sigma_{r,v}^2$. Building upon the analysis developed in Section VI-B, we can now express the freezing probability as $P_f(n) = \Pr[S_v(n) \ge D'(n)]$ with $D'(n) = \tau_v(\gamma_v^\circ R(n+n_0) + \sum_{h=1}^{n+n_0} w_v^h) = D_v(n) + Y(n)$ where $Y(n)$ has zero mean, so that $\mathrm{E}[S_v(n) - Y(n)] = \mu$, as in Section VI-B. Then, repeating the steps of Section VI-B, we get

$$n_0 \ge \frac{\alpha\Delta_v'(c)}{s_v(c)}\sqrt{n\log\left(\frac{1}{\sqrt{P_f^*}}\right)} - n(1-\alpha) = \beta'\sqrt{n} - (1-\alpha)n$$

where $\Delta_v'(c) \ge \Delta_v(c)$ because of the additional variance due to rate estimation errors. Approximating $\Delta_v(c)'$ as

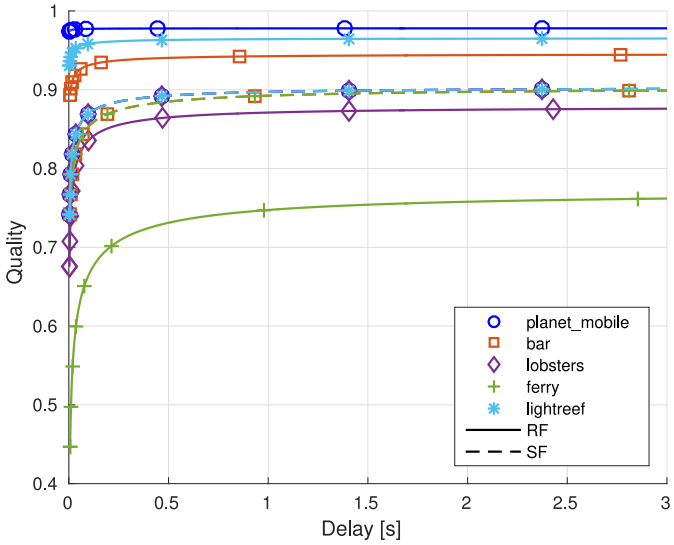


Fig. 9. Video quality and initial playback delays for different values of α, using RF and SF.

$k\sqrt{\sigma_v^2(c)+\tau_v^2\sigma_{r,v}^2}$ (i.e., increasing the variance of the GOP size to account for the channel capacity fluctuations), we get

$$\beta' = \frac{\alpha\Delta_v'(c)}{s_v(c)}\sqrt{\log\left(\frac{1}{\sqrt{P_f^*}}\right)} \simeq \frac{k\sqrt{\sigma_v^2(c)+\tau_v^2\sigma_{r,v}^2}}{\gamma_v^\circ(c)R\tau_v}\sqrt{\log\left(\frac{1}{\sqrt{P_f^*}}\right)}$$

$$n_0 = \beta'\sqrt{\min\{n^{*\prime}, n_{\max}\}} - (1-\alpha)\min\left\{n^{*\prime}, n_{\max}\right\} \quad (21)$$

with $n^{*\prime} = \frac{\beta'^2}{4(1-\alpha)^2}$.

Clearly, the size of the play-out buffer impacts the initial delay τ_0. A rough estimate of τ_0 can be obtained by assuming that the aggregate size of the initial n_0 GOPs is equal to $n_0 s_v(c) = n_0\alpha\tau_v\gamma_v^\circ R$ and that these GOPs are downloaded at the assigned share of the nominal link rate, i.e., $\gamma_v^\circ R$, so that we get $\tau_0 = \alpha n_0\tau_v$. From this result and (21), we see that the smaller α, the lower τ_0. On the other hand, the smaller α, the lower the quality of the segments downloaded by the CHP. There exists then a tradeoff between the initial playback delay and the average quality of the video when varying α. Fig. 9 shows such a tradeoff for a few sample videos, when using both the SF (dashed line) and RF (solid lines) RM algorithms. The plot has been obtained by setting $k = 7$, $\sigma_v(c)/s_v(c) = 5\%$, $P_f^* = 5\%$, and $\sigma_{r,v} = 0.01$. The results show that SF makes it possible not only to offer the same quality to all video sequences, but also to provide the same playback delay for a certain quality level. RF, instead, can give better quality (or lower playback delay) to certain videos, while others will suffer very poor quality, even when the initial delay is allowed to be large.

IX. Conclusions and Future Directions

We designed a framework for video admission control in wireless systems that exploits machine learning algorithms to optimize resources management. By means of simulation, we showed that our proposal outperforms offline video analysis techniques in terms of the trade-off between QoE delivered and computational costs.

One promising future direction to further improve the proposed method could be to extend the unsupervised learning phase by using a richer input vector, including other encoding parameters, and a deeper architecture, thereby considering a hierarchical generative model of the data distribution [43]. However, more complex models usually need larger training datasets, which must provide enough statistical information to extract a good set of descriptive features. An important step would therefore be to also increase the amount of data used to train the generative model, which can be accomplished by collecting more videos or integrating other available datasets into the framework. Finally, exploiting unsupervised learning to build an expressive set of high-level features allows great flexibility to the proposed framework, which can be used to transfer knowledge across several tasks [65].

References

[1] A. Testolin *et al.*, "A machine learning approach to QoE-based video admission control and resource allocation in wireless systems," in *Proc. IEEE Med-Hoc-Net*, Piran, Slovenia, Jun. 2014, pp. 31–38.

[2] N. Amram *et al.*, "QoE-based transport optimization for video delivery over next generation cellular networks," in *Proc. IEEE ISCC*, 2011, pp. 19–24.

[3] *DASH*. Accessed: Dec. 22, 2017. [Online]. Available: http://www-itec.uni-klu.ac.at/dash/

[4] T. Stockhammer, "Dynamic adaptive streaming over HTTP—Standards and design principles," in *Proc. ACM MMSys*, San Jose, CA, USA, Feb. 2011, pp. 133–144.

[5] D. Munaretto, F. Giust, G. Kunzmann, and M. Zorzi, "Performance analysis of dynamic adaptive video streaming over mobile content delivery networks," in *Proc. IEEE ICC Commun. QoS Rel. Model. Symp. (ICC CQRM)*, Sydney, NSW, Australia, 2014, pp. 1059–1064.

[6] M. Zanforlin, D. Munaretto, A. Zanella, and M. Zorzi, "SSIM-based video admission control and resource allocation algorithms," in *Proc. IEEE WiVid Workshop WiOpt*, Hammamet, Tunisia, May 2014, pp. 656–661.

[7] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," in *Parallel Distributed Processing: Explorations on the Microstructure of Cognition. Volume 1: Foundations*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA, USA: MIT Press, 1986.

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

[9] "Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference," ITU, Geneva, Switzerland, ITU-T Recommendation J.144, Mar. 2004.

[10] K. Seshadrinathan and A. C. Bovik, "Motion tuned spatio-temporal quality assessment of natural videos," *IEEE Trans. Image Process.*, vol. 19, no. 2, pp. 335–350, Feb. 2010.

[11] M. A. Saad and A. C. Bovik, "Blind quality assessment of videos using a model of natural scene statistics and motion coherency," in *Proc. Conf. Rec. 46th Asilomar Conf. Signals Syst. Comput. (ASILOMAR)*, Pacific Grove, CA, USA, Nov. 2012, pp. 332–336.

[12] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE," in *Proc. 8th Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, Nice, France, Dec. 2012, pp. 97–108.

[13] Z. Li *et al.*, "Probe and adapt: Rate adaptation for HTTP video streaming at scale," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 4, pp. 719–733, Apr. 2014.

[14] S. Petrangeli, J. Famaey, M. Claeys, S. Latré, and F. De Turck, "QoE-driven rate adaptation heuristic for fair adaptive video streaming," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, no. 2, Oct. 2015, Art. no. 28.

[15] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 187–198, Oct. 2014.

[16] D. Stohr *et al.*, "QoE analysis of dash cross-layer dependencies by extensive network emulation," in *Proc. Workshop QoE Based Anal. Manag. Data Commun. Netw. (Internet-QoE)*, Florianópolis, Brazil, 2016, pp. 25–30.

[17] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over HTTP," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 325–338, 2015.

[18] A. Bokani, M. Hassan, and S. Kanhere, "HTTP-based adaptive streaming for mobile clients using Markov decision process," in *Proc. 20th Int. Packet Video Workshop*, San Jose, CA, USA, Dec. 2013, pp. 1–8.

[19] C. Zhou, C.-W. Lin, and Z. Guo, "mDASH: A Markov decision-based rate adaptation approach for dynamic HTTP streaming," *IEEE Trans. Multimedia*, vol. 18, no. 4, pp. 738–751, Apr. 2016.

[20] M. Gadaleta, F. Chiariotti, M. Rossi, and A. Zanella, "D-DASH: A deep Q-learning framework for DASH video streaming," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 703–718, Dec. 2017.

[21] M. Claeys *et al.*, "Design of a Q-learning-based client quality selection algorithm for HTTP adaptive video streaming," in *Proc. Adapt. Learn. Agents Workshop (ALA)*, St. Paul, MN, USA, May 2013, pp. 30–37.

[22] M. Claeys *et al.*, "Design and optimisation of a (FA)Q-learning-based HTTP adaptive streaming client," *Connection Sci.*, vol. 26, no. 1, pp. 25–43, 2014.

[23] V. Martín, J. Cabrera, and N. García, "Q-learning based control algorithm for HTTP adaptive streaming," in *Proc. Int. Conf. Vis. Commun. Image Process. (VCIP)*, Singapore, Dec. 2015, pp. 1–4.

[24] L. Yu, T. Tillo, and J. Xiao, "QoE-driven dynamic adaptive video streaming strategy with future information," *IEEE Trans. Broadcast.*, vol. 63, no. 3, pp. 523–534, Sep. 2017.

[25] J. Chen, M. Ammar, M. Fayed, and R. Fonseca, "Client-driven network-level QoE fairness for encrypted 'DASH-S,"' in *Proc. Workshop QoE Based Anal. Manag. Data Commun. Netw. (Internet-QoE)*, Florianópolis, Brazil, 2016, pp. 55–60.

[26] A. S. Abdallah and A. B. MacKenzie, "A cross-layer controller for adaptive video streaming over IEEE 802.11 networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, U.K., Jun. 2015, pp. 6797–6802.

[27] J. Kua, G. Armitage, and P. Branch, "A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1842–1866, 3rd Quart., 2017.

[28] L.-Q. Xu and Y. Li, "Video classification using spatial–temporal features and PCA," in *Proc. IEEE ICME*, Baltimore, MD, USA, Jul. 2003, pp. 485–488.

[29] S. Latré and F. De Turck, "Joint in-network video rate adaptation and measurement-based admission control: Algorithm design and evaluation," *J. Netw. Syst. Manag.*, vol. 21, no. 4, pp. 588–622, Dec. 2013.

[30] S. Qadir and A. A. Kist, "Video-aware measurement-based admission control," in *Proc. Aust. Telecommun. Netw. Appl. Conf. (ATNAC)*, Nov. 2013, pp. 178–182.

[31] B. Feitor, P. Assuncao, J. Soares, L. Cruz, and R. Marinheiro, "Objective quality prediction model for lost frames in 3D video over TS," in *Proc. IEEE ICC*, Budapest, Hungary, Jun. 2013, pp. 622–625.

[32] M. Naccari, M. Tagliasacchi, and S. Tubaro, "No-reference video quality monitoring for H.264/AVC coded video," *IEEE Trans. Multimedia*, vol. 11, no. 5, pp. 932–946, Aug. 2009.

[33] P. Seeling, M. Reisslein, and B. Kulapala, "Network performance evaluation using frame size and quality traces of single-layer and two-layer video: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 6, no. 3, pp. 58–78, 3rd Quart., 2004.

[34] M. Katsarakis, R. C. Teixeira, M. Papadopouli, and V. Christophides, "Towards a causal analysis of video QoE from network and application QoS," in *Proc. Workshop QoE Based Anal. Manag. Data Commun. Netw. (Internet-QoE)*, Florianópolis, Brazil, 2016, pp. 31–36.

[35] G. Dimopoulos, I. Leontiadis, P. Barlet-Ros, and K. Papagiannaki, "Measuring video QoE from encrypted traffic," in *Proc. Internet Meas. Conf. (IMC)*, Santa Monica, CA, USA, 2016, pp. 513–526.

[36] I. Orsolic, D. Pevec, M. Suznjevic, and L. Skorin-Kapov, "A machine learning approach to classifying YouTube QoE based on encrypted network traffic," *Multimedia Tools Appl.*, vol. 76, no. 21, pp. 22267–22301, Nov. 2017.

[37] M. Ries, M. Slanina, and D. M. Garcia, "Reference free SSIM estimation for full HD video content," in *Proc. 21st Int. Conf. Radioelektronika*, Brno, Czech Republic, Apr. 2011, pp. 1–4.

[38] T.-L. Lin *et al.*, "NR-bitstream video quality metrics for SSIM using encoding decisions in AVC and HEVC coded videos," *J. Vis. Commun. Image Represent.*, vol. 32, pp. 257–271, Oct. 2015.

[39] T. Shanableh, "Prediction of structural similarity index of compressed video at a macroblock level," *IEEE Signal Process. Lett.*, vol. 18, no. 5, pp. 335–338, May 2011.

[40] P. Goudarzi, "A no-reference low-complexity QoE measurement algorithm for H.264 video transmission systems," *Scientia Iranica*, vol. 20, no. 3, pp. 721–729, Jun. 2013.

[41] A. Rossholm and B. Lövströem, "A new low complex reference free video quality predictor," in *Proc. IEEE 10th Workshop Multimedia Signal Process.*, Cairns, QLD, Australia, Oct. 2008, pp. 765–768.

[42] S. Antani, R. Kasturi, and R. Jain, "A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video," *Pattern Recognit.*, vol. 35, no. 4, pp. 945–965, Apr. 2002.

[43] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 24, Dec. 2012, pp. 1–9.

[45] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Modeling human motion using binary latent variables," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 19, Jan. 2007, pp. 1345–1352.

[46] A. Testolin, I. Stoianov, A. Sperduti, and M. Zorzi, "Learning orthographic structure with sequential generative neural networks," *Cogn. Sci.*, vol. 40, no. 3, pp. 579–606, Apr. 2016.

[47] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," Video Coding Experts Group, Austin, Texas, USA, Rep. VCEG-M33, Apr. 2001.

[48] P. Hanhart and T. Ebrahimi, "Calculation of average coding efficiency based on subjective quality scores," *J. Vis. Commun. Image Represent.*, vol. 25, no. 3, pp. 555–564, Apr. 2014.

[49] J. M. Libert, C. P. Fenimore, and P. Roitman, "Simulation of graded video impairment by weighted summation: Validation of the methodology," in *Proc. SPIE*, vol. 3845. Boston, MA, USA, Nov. 1999, pp. 254–265.

[50] L. Badia *et al.*, "Cognition-based networks: Applying cognitive science to multimedia wireless networking," in *Proc. Video Everywhere (VidEv) Workshop IEEE (WoWMoM)*, Sydney, NSW, Australia, Jun. 2014, pp. 1–6.

[51] T. Zinner, O. Hohlfeld, O. Abboud, and T. Hossfeld, "Impact of frame rate and resolution on objective QoE metrics," in *Proc. Workshop Qual. Multimedia Exp. (QoMEX)*, Trondheim, Norway, Jun. 2010, pp. 29–34.

[52] *Test Media Repository*. Accessed: Dec. 22, 2017. [Online]. Available: http://media.xiph.org/video/derf/

[53] *Joint Scalable Video Model—Reference Software*. Accessed: Dec. 22, 2017. [Online]. Available: https://www.hhi.fraunhofer.de/en/departments/vca/research-groups/image-video-coding/research-topics/svc-extension-of-h264avc/jsvm-reference-software.html

[54] *x264 Encoder*. Accessed: Dec. 22, 2017. [Online]. Available: http://www.videolan.org/developers/x264.html

[55] M. Fiedler, T. Hossfeld, and P. Tran-Gia, "A generic quantitative relationship between quality of experience and quality of service," *IEEE Netw.*, vol. 24, no. 2, pp. 36–41, Mar./Apr. 2010.

[56] M. Zorzi, A. Zanella, A. Testolin, M. D. F. D. Grazia, and M. Zorzi, "Cognition-based networks: A new perspective on network optimization using learning and distributed intelligence," *IEEE Access*, vol. 3, pp. 1512–1530, 2015.

[57] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines," *Cogn. Sci.*, vol. 9, no. 1, pp. 147–169, Jan./Mar. 1985.

[58] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, Aug. 2002.

[59] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984.

[60] M. Zorzi, A. Testolin, and I. P. Stoianov, "Modeling language and cognition with deep unsupervised learning: A tutorial overview," *Front. Psychol.*, vol. 4, p. 515, Aug. 2013.

[61] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," in *Neural Networks: Tricks of the Trade*. Heidelberg, Germany: Springer, 2012, pp. 599–619.

[62] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[63] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. Int. Joint Conf. Artif. Intell.*, vol. 14. Montreal, QC, Canada, 1995, pp. 1137–1145.
[64] A. Testolin, I. Stoianov, M. De Filippo De Grazia, and M. Zorzi, "Deep unsupervised learning on a desktop PC: A primer for cognitive scientists," *Front. Psychol.*, vol. 4, p. 251, May 2013.
[65] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, Bellevue, Washington, USA, Jul. 2012, vol. 27, pp. 17–36.

Michele De Filippo De Grazia is a Researcher with the Department of General Psychology, University of Padova, Italy. His research interests are focused on machine learning techniques like neural networks and SVM algorithms to develop intelligent systems to, for example, fault detection and diagnosis of HVAC systems, in time series analysis and for rehabilitation videogame. He is also involved in the development of computational models (as deep networks) for cognitive processes.

Daniel Zucchetto (S'17) received the bachelor's degree in information engineering and the master's degree in telecommunication engineering from the University of Padova, Italy, in 2012 and 2014, respectively, where he is currently pursuing the Ph.D. degree with the Department of Information Engineering. His research interests include low-power wide-area network technologies and next generation cellular networks (5G), with particular focus on their application to the Internet of Things.

Alberto Testolin received the M.Sc. degree in computer science and the Ph.D. degree in cognitive science from the University of Padova, in 2011 and 2015, respectively, where he is currently a Post-Doctoral Researcher focusing on computational modeling of cognitive processes. His main interests include deep learning, recurrent neural networks and probabilistic generative models, which are applied to investigate visual processing and attentional mechanisms.

Andrea Zanella (S'98–M'01–SM'13) received the Laurea degree in computer engineering and the Ph.D. degree in electronic and telecommunications engineering from the University of Padova, Italy, in 1998 and 2011, respectively, where he is an Associate Professor with the Department of Information Engineering. In 2000, he was a Visiting Scholar with the Department of Computer Science, University of California, Los Angeles. He is one of the coordinators of the Signals and Networking Research Laboratory. His long-established research activities are in the fields of protocol design, optimization, and performance evaluation of wired and wireless networks. He is an Associate Editor of the IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, the IEEE INTERNET OF THINGS JOURNAL, and the *Digital Communications and Networks*.

Marco Zorzi is a Full Professor of cognitive psychology and artificial intelligence with the Department of General Psychology and the Padova Neuroscience Center, University of Padova, Italy. He is also a Senior Researcher and a Consultant with San Camillo Neurorehabilitation Hospital, Venice. Research in his laboratory, the Computational Cognitive Neuroscience Laboratory, is at the frontiers between cognitive science, computer science and neuroscience, with a focus on the computational bases of cognition, from development to skilled performance and to breakdowns of processing in atypical development or after brain damage. His research on generative models of human cognition has been supported by an award from the European Research Council. Machine learning techniques, particularly deep neural networks, are used in his lab for modeling human behavior and cognition, as well as for neuroinformatics and industry-related applications.

Michele Zorzi (S'89–M'95–SM'98–F'07) received the Laurea and Ph.D. degrees in electrical engineering from the University of Padova in 1990 and 1994, respectively. From 1992 to 1993, he was on leave from the University of California at San Diego (UCSD). In 1993, he joined the Faculty of the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy. He was with the Center for Wireless Communications, UCSD, for three years. In 1998, he joined the School of Engineering, University of Ferrara, Italy, where he became a Professor in 2000. Since 2003, he has been with the faculty of the Information Engineering Department, University of Padova. His current research interests include performance evaluation in mobile communications systems, WSN and Internet-of-Things, cognitive communications and networking, 5G mmWave cellular systems, and underwater communications and networks. He is currently the Editor-in-Chief of the IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, and was the Editor-in-Chief of the IEEE WIRELESS COMMUNICATIONS from 2003 to 2005 and the IEEE TRANSACTIONS ON COMMUNICATIONS from 2008 to 2011. He served as a Member-at-Large of the Board of Governors of the IEEE Communications Society from 2009 to 2011, and as its Director of Education from 2014 to 2015.