

A Review of Predictive Quality of Experience Management in Video Streaming Services

Maria Torres Vega, *Member, IEEE*, Cristian Perra, *Senior Member, IEEE*, Filip De Turck, *Senior Member, IEEE*, and Antonio Liotta, *Senior Member, IEEE*

Abstract—Satisfying the requirements of devices and users of online video streaming services is a challenging task. It requires not only managing the network Quality of Service but also to exert real-time control, addressing the user’s Quality of Experience (QoE) expectations. QoE management is an end-to-end process that, due to the ever-increasing variety of video services, has become too complex for conventional ‘reactive’ techniques. Herein, we review the most significant ‘predictive’ QoE management methods for video streaming services, showing how different machine learning approaches may be used to perform proactive control. We pinpoint a selection of the best suited machine learning methods, highlighting advantages and limitations in specific service conditions. The review leads to lessons learned and guidelines to better address QoE requirements in complex video services.

Index Terms—Machine Learning, Quality of Experience Management, Video Streaming Services

I. INTRODUCTION

In the current complex and broad wireless environment, satisfying the users’ and devices’ requirements arising from digital services becomes fundamental [1]. In this situation, being able to monitor applications, and act upon them when improvements are required, is essential. Particularly crucial is the case of video streaming services. New streaming protocols increase bandwidth requirements and transmission complexity, which are critical elements for service and network providers. This task has traditionally been studied in the context of network Quality-of-Service (QoS) management. However, due to the variability in channel conditions, streaming over wireless networks incurs quality degradation even when there is sufficient nominal capacity [2]. This degradation cannot be assessed merely by means of QoS factors, which only reflect the status of individual networks but do not comprehensively capture the end-to-end features that affect the overall quality delivered to the user. To address these elements, Quality of Experience (QoE) management has been recognized as a much more effective proposition [3].

Today, humans are quality meters [4], and their expectations, perceptions and needs carry a great value in the context of digital service delivery [5]. QoE is defined as the degree of delight or annoyance of the user of an application or service, based on both objective and subjective psychological measurements [6]. QoE has therefore multiple dimensions,

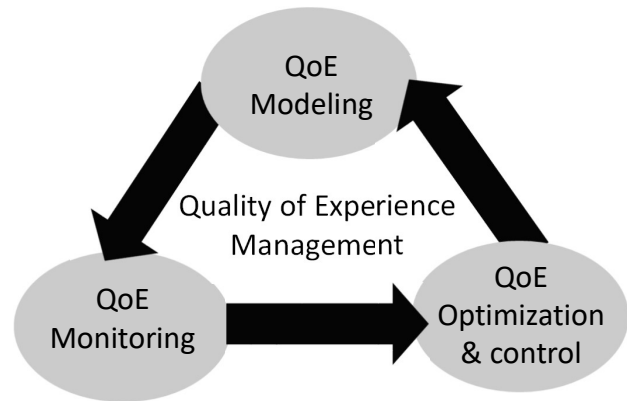


Fig. 1: General block diagram to perform QoE management.

encompassing both objective (e.g., performance related) and subjective (e.g., user related) aspects [7]. The overall goal of QoE management is to optimize the end-user QoE (end-user perspective), while making efficient use of network resources (current and future ones) and maintaining a satisfied customer base (provider perspective) [6]. Thus, successfully managing QoE for any specific application requires understanding, identifying and monitoring the multiple factors (both subjective and objective ones) that affect the quality, considering the various actors in the service provisioning chain. This will lead to QoE models that capture the parameters to be monitored and ultimately, put in place effective QoE optimization strategies.

The QoE management process may be broken down into three general steps (Figure 1): QoE modeling, QoE monitoring and measurements, and QoE optimization and control. Traditionally, QoE management has been performed by means of subjective evaluations and deterministic adaptation. Therefore, a selected group of users would rate the received video content (using the Mean Opinion Score scale [6]). From the server side, the users’ feedback would be monitored and the service would slowly be adapted to the users’ requirements. Due to the time required by the subjective evaluations and adaptations, this procedure can only be performed in an offline manner. However, with the exponential increase in number and variety of video streaming services, client devices and network conditions require real-time, accurate and adaptable QoE management. Thus, performing QoE management via traditional methods becomes unfeasible. To solve this, in the last years, several research lines have followed the paths of artificial intelligence and Machine Learning (ML). ML allows improving the accuracy of quality models [8], helping the

Pre-printed version.

Please cite as: *M. Torres Vega, C. Perra, F. De Turck and A. Liotta: A Review of Predictive Quality of Experience Management in Video Streaming Services. IEEE Transactions on Broadcasting (2018).*

monitoring process [9] or providing a fast optimization feedback loop for adaptive streaming applications [10]. However, picking ML model best suited to the type of application and situation is in itself an open research topic.

In this paper, we review the most significant prediction-based approaches to improve the QoE management loop. We split our analysis into the three components of Figure 1, to comparatively analyze each of the approaches, pinpointing major advantages as well as drawbacks. With this, we aim to provide a set of guidelines and lessons learned from our experience with predictive QoE management of video streaming services.

The remainder of this paper is organized as follows. Section II introduces machine learning, its main categories and type of models. Section III presents the core contribution of this paper, the review of machine learning-based approaches that aim to tackle challenges the QoE management loop. While Section III-A focuses on the approaches that apply machine learning techniques to improve the prediction of video quality models, Section III-B covers the state-of-the-art solutions on predictive video quality optimization and control techniques. Finally, Section III-C focuses on predictive models which complement and enhance quality monitoring. In addition to the state-of-the-art review, in this paper we provide a discussion on pros and cons, technicalities and preferences on selecting ML models depending on the situation, system and problem to solve (Section IV). We provide final remarks and conclusions in Section V.

II. MACHINE LEARNING TECHNIQUES

Machine learning (ML) is the field of computer science that gives computers the ability to derive models directly from data, without having to explicitly program them, making inference and predictions from input samples [11]. ML is employed in a range of computing tasks where designing and programming explicit algorithms with good performance is difficult or impossible. Example applications include email filtering, fraud or intrusion detection, optical character recognition (OCR), automatic ranking or classification, and computer vision, to mention but a few.

ML approaches are classified in three broad categories, depending on whether there is a learning signal or a feedback available to a learning system. 1) in supervised learning (SL), the model is presented with example inputs and their desired outputs; the goal is to learn a general rule that maps the inputs into the outputs. 2) in reinforcement learning (RL), the training data (in form of rewards and punishments) is given only as feedback to the programs actions in a dynamic environment, such as when driving a vehicle or playing a game against an opponent. 3) in unsupervised learning (UL), no labels are given to the algorithm, leaving it on its own to find the structure embedded in the input data. UL can be a goal in itself (e.g., discovering hidden patterns in data) or a means towards an end (e.g., feature learning). Next, we provide further clarifications on each of these categories and introduce the algorithms that are used the most in QoE management.

A. Supervised Learning

A supervised ML model (SL-ML) is a function which maps input-output pairs of data [11]. Formally, a dataset \mathcal{D} consists of a set of pairs $\{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_n, \mathbf{y}_n)\}$. For each specific pair i , \mathbf{x}_i is the vector of input features and \mathbf{y}_i is the vector of outputs. The pairs contained in \mathcal{D} are sampled from an unknown distribution (e.g. a real system) in which the exact function $f : \mathbf{X} \rightarrow \mathbf{Y}$ is unknown (and it cannot be determined). \mathbf{X} represents the input space, i.e. the union of all possible vectors with input features ($\cup_{i=1}^n \mathbf{x}_i \subset \mathbf{X}$) and \mathbf{Y} represents the output space, i.e. the union of all possible vectors with output features ($\cup_{i=1}^n \mathbf{y}_i \subset \mathbf{Y}$). The goal of the ML algorithm is to find (based on the knowledge extracted from \mathcal{D}) a function $\hat{f} : \mathbf{X} \rightarrow \mathbf{Y}$ which is capable to approximate (as accurately as possible) the functionality of the f function (the unknown distribution of the real system). The process of finding the \hat{f} function is usually referred to as training. In general, for validation purposes, the dataset \mathcal{D} is split into two parts: one part (the training dataset) is used to find \hat{f} , while the other part (the testing dataset) is used to measure how well the \hat{f} function approximates f . Formalizing this mathematically, this means that $\forall (\mathbf{x}, \mathbf{y}) \in (\mathbf{X}, \mathbf{Y}), d(f(\mathbf{x}), \mathbf{y})$ is minimized, where $d(\cdot, \cdot)$ can be a suitable distance measure, such as root mean square error or the PCC. The obtained \hat{f} function is, in fact, the ML model trained in a supervised fashion [11].

Depending on the nature of the output provided by the approach, SL models are classified in two categories: 1) classification, where inputs are divided into two or more classes, and the learner must produce a model that assigns unseen inputs to one or more (multi-label classification) of these classes; and 2) regression, where the outputs are continuous rather than discrete. In addition, depending on how easy it is to capture a comprehensible relation between inputs and output, the models are either white or black boxes. White boxes provide an understanding of the model behind, while black boxes do not offer such relation. Table I provides a description of the most significant supervised learning approaches, focusing on the ones which have been used in the state-of-the-art for QoE Management.

One of the most known and simplest white boxes is linear regression [12], which attempts to model the relationship between a scalar (output) and one or more independent variables by means of a linear multidimensional model of the input data. Decision trees learning uses a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value [13]. They are classified according to the type of output provided. On the one hand, tree models, where the target variable takes a value from a finite set, are called classification trees. Leaves represent class labels and branches, conjunctions of features that lead to those class labels. On the other hand, decision trees, where the target variable can take continuous values (typically real numbers), are called regression trees. The performance of regression and decision trees can be further improved by means of an ensemble approach. Ensembles use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning

TABLE I: Representative SL models (name, acronym, categories and short description) used for QoE management.

Name	Ac.	Output	Box	Description
Multiple Linear Regression	LR	Regression	White	Linear model relation: independent variables. to scalar (output).
Standard Decision Trees	DT	Classification	White	Map observations to target value on binary tree.
Ensemble Decision Trees	EDT	Classification	White	Multiple DT to improve performance.
Gaussian Process Regression	GPR	Regression	Black	Values prediction by interpolation on a Gaussian distr.
Support Vector Machines	SVM	Classification	Black	Finding a hyperplane that optimally classify outputs.
Artificial Neural Networks	ANN	Classification	Black	Estimate functions dependent on unknown inputs.
		Regression		

algorithms [14]. Unlike a statistical ensemble in statistical mechanics, which is usually infinite, a ML ensemble refers only to a concrete finite set of alternative models, but typically allows for a much more flexible structure to exist among those alternatives. Evaluating the prediction of an ensemble typically requires more computation than for a single model. Thus ensembles are mostly used as a way to compensate for poor learning algorithms by performing extra computation. For this reason, fast (less accurate) algorithms such as decision trees are commonly used with ensembles.

While simpler, white boxes have also been demonstrated to have limited predictive capacity or to be inflexible. Therefore, the best classification and regression accuracy is typically achieved by black-box models [15], which do not, however, provide a clear explanation of the reasons as to how they have come to a certain prediction. One prominent example is the Gaussian Process Regression (or Kriging) [16]. This aims to predict values by means of interpolation in which the interpolated values are modeled by a Gaussian process governed by prior co-variances. Support Vector Machines (SVMs) [17] use associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training samples (belonging to one out of two categories), an SVM training algorithm builds a model that assigns new samples into categories, making a non-probabilistic binary linear classification. Finally, the artificial neural networks (ANNs) [18] are a family of models inspired to biological neural networks, used to estimate or approximate functions that can depend on a large number of generally unknown inputs. ANNs are generally presented as systems of interconnected nodes or "neurons". The connections among these neurons have numeric weights that are tuned based on various optimization methods, making neural nets adaptive to inputs and capable of learning.

B. Reinforcement Learning

The Reinforcement Learning (RL) paradigm [11] takes its inspiration from the field of behaviorist psychology. In it, an agent, which does not possess a complete model of the surrounding environment and, thus, does not know the

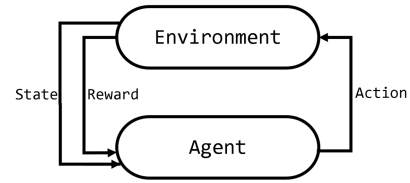


Fig. 2: Reinforcement learning conceptual data flow

effects of an action, adopts a trial-and-error mechanism. By trying different actions, the environment changes his state and generates a reward information. The agent then dynamically adapts its actions on the basis of the received state and reward, attempting to maximize the reward of his actions (Figure 2).

We can formulate the RL problem as a 4-tuple Markov Decision Problem (MDP) [19], $(S, A_s, P_a(s_t, s_{t+1}), R_a(s_t, s_{t+1}))$. In it, S is a finite state of states; A_s is a finite set of actions available to the agent in state $s \in S$; $P_a(s_t, s_{t+1})$ is the transition probability from state s_t to state s_{t+1} when the agent chooses action $a \in A_{s_t}$; and $R_a(s_t, s_{t+1})$ is the *immediate reward* that the agent gets when it chooses action $a \in A_{s_t}$ and the state changes from s_t to s_{t+1} . The transition probabilities of the Markov chain depends on the learner's action and on the external environment. The success of every action a is measured by a reward r .

The agent aims to optimize a policy $\pi(s)$ (i.e., a function that links every state to an action, and depends on the expected long-term reward of each action that the agent already estimated). Learning to act in an environment makes the agent to choose actions to maximize future rewards, defining, in this way, the agent's behavior in the environment. The long-term reward is defined as the *value* of a state ($V^\pi(s)$), i.e. the total amount of reward an agent can expect to accumulate in the future, starting from that state ($V^\pi(s) = \sum_{\tau=t}^{\infty} \gamma^{\tau-t} R_\tau$). The variable $\gamma \in [0, 1]$ is called discounting factor and R is the immediate reward corresponding to an action. The long-term reward is an exponentially weighted sum of the rewards that will be obtained in the future evolution of the MDP.

RL methods are classified in model-based (RLMB) and model-free (RLMF) (Table II). While an RLMB algorithm depends on the knowledge of an explicit model of the environment and of the agent, an RLMF algorithm lacks of such specific model and evaluates the performance based on trial-and-error learning.

In addition, the agent can estimate the value of a given policy either by an On-policy or an Off-policy approach. In a On-policy scenario, on the one hand, both the generation of the data and the evaluation are performed using the same policy. On the other hand, in an Off-line policy scenario, the policy to be evaluated is different from the policy that generates the data.

Online and offline RL are related to the availability or not of the training set. An online RL algorithm does not have any offline pre-training processing and learns the policy online. An offline RL algorithm uses a training dataset for computing the policy.

Common RLMF methods are Q-learning, SARSA (State-

TABLE II: Representative RL classes and methods used in QoE optimization and control.

RL Method	Acronym	Class	Description
Markov Decision Process	MDP	RLMB	MDP model learning and solving concurrently
Q-learning	QL	RLMF	Off-Policy algorithm for temporal difference learning of an MDP
State-action-reward-state-action	SARSA	RLMF	On-policy algorithm for temporal difference learning of an MDP.
Post decision state	PDS	RLMF	Off-policy algorithm for temporal difference learning to reduce RL complexity

Action-Reward-State-Action), and PDS (Post decision state). Both Q-learning and SARSA are RLMF algorithms used to find an optimal action-selection policy for any given (finite) MDP. They are different in the way they estimate the value of their policy, being Q-learning an off-policy model and SARSA an on-policy model. In addition, SARSA takes into account the control policy by which the agent is moving, and incorporates that into its update of action values, where Q-learning simply assumes that an optimal policy is being followed. Post decision states (PDSs) variables [20] are used as a tool for the reduction of dynamic programming algorithm complexity. PDSs have been used for reducing the complexity of the learning agent in RL online algorithm applied to the problem of adaptation strategy for DASH clients [21]. The general idea is that introducing intermediate states through PDS variables in the transition between temporally adjacent steps can reduce the convergence time to the optimal solution.

C. Unsupervised Learning

Unsupervised Learning (UL) is the ML task of inferring a function to describe hidden structure from unlabeled data [11]. Since the training samples given to the learner are not benchmarked (i.e., they are unlabeled), it is not possible to evaluate the accuracy of the structure. A central case of unsupervised learning is the problem of density estimation in statistics, though unsupervised learning encompasses many other problems (and solutions) involving summarizing and explaining key features of the data. Due to not requiring labels for training, unsupervised learning suites the best cases in which the target is unknown or when scalability is essential.

Given the broad variety of UL methods, Table III aims to summarize the most used methods in QoE management that we will name in the following Sections. UL methods can be classified according to their purpose. Cluster analysis or clustering is the task of grouping a set of objects together in such a manner that the objects within a group (cluster) are more similar among one another than to the ones in other groups. Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. For example, K-means clustering aims to partition observations into clusters in which each observation belongs to the cluster

TABLE III: Representative UL and UDL models used in QoE management.

Type	Name	Acronym	Description
Clustering	K-Means	CKM	Clusters on central vector.
	Hierarchical Distribution	CHI	Clusters on their distance.
		CDI	Clusters on prob.distribution.
ANN	Hebbian	HNN	Neurons weights adaptation.
	Generative	GAN	2 ANN contest.
Latent Variable Learning	Expectation-Maximization	EM	Maximum likelihood iteration.
	Moments	MM	Moments derivation.
	Principal Component Analysis	PCA	Orthogonal transform. of possibly correlated into linearly uncorrelated variables.
Deep Learning	Restricted Boltzmann Machines	RBM	Probability distribution over its set of inputs

with the nearest mean, serving as a prototype of the cluster. Hierarchical clustering, is based on the core idea of objects are more related to nearby objects than to objects farther away. Thus, these algorithms group objects in clusters based on their distance and a cluster can be described largely by the maximum distance needed to connect parts of the cluster. At different distances, different clusters are formed, which can be represented using a dendrogram. Therefore, these algorithms do not provide a single partitioning of the dataset but provide an extensive hierarchy of clusters that merge with one another at certain distance. Another example of clustering is the distribution-based approach. Clusters are defined by the objects that most likely belong to the same distribution. Clustering techniques have shown outstanding capabilities to classify video content for prediction purposes, as we see in Section III-C.

Certain types of ANNs can be considered belonging to the UL category. Based on Hebb's Theory [22], Hebb's principle can be described as the method of determining how to alter the weights between model neurons. The weight between two neurons increases if the two neurons activate simultaneously, and reduces if they activate separately. Nodes that tend to be either both positive or both negative at the same time have strong positive weights, while those that tend to be opposite have strong negative weights. The theory attempts to explain associative or Hebbian learning, in which simultaneous activation of cells leads to pronounced increases in synaptic strength between those cells. It also provides a biological basis for error-free learning methods for education and memory rehabilitation. Goodfellow et al. [23] introduced the Generative adversarial networks (GANs), a type of artificial intelligence algorithms that implement a system of two neural networks contesting with each other in a zero-sum game framework. GANs have been used to produce samples of photo-realistic images for visualization purposes or for pattern generation in 2D and 3D videos.

Latent Variable is a statistical model that relates a set of observable variables (so-called manifest variables) to a set of latent variables. It is assumed that the responses on the manifest variables are the result of an individual's position on the latent variable(s), and that the manifest variables have nothing in common after controlling for the latent variable

(local independence). Different types of the latent variable models have appeared. A first example is the Expectation–Maximization (EM) algorithm, which iterates to find maximum likelihood or maximum a posteriori (MAP) estimates of input parameters in statistical models, where the model depends on unobserved latent variables [11]. The method of moments (MM) is a method of estimation of population parameters by means of derivations that relate the population moments (i.e., the expected values of powers of the random variable under consideration) to the parameters of interest [24]. Finally, one of the most known and used methods is the Principal Component Analysis (PCA) [25]. PCA uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.

It is possible to enhance the UL performance by combining the models with Deep Learning (DL) techniques [26], which attempt to model high-level abstractions in data by using a deep graph with multiple processing layers (multiple linear and non-linear transformations). The combined action of UL and DL (UDL) becomes a very powerful tool for quality prediction and monitoring. One example of such approaches are the Restricted Boltzmann Machines (RBMs) [27]. RBMs are generative stochastic ANNs that can learn a probability distribution over its set of inputs by means of only inter-layer connections. It distributes its neurons in two layers. The visible layer corresponds to the input features. In the hidden layer, hidden features are automatically extracted by the RBM model from the input data. Every visible neuron is thus connected to every hidden network and has an associated weight that is modeled according to the input (visible features).

III. PREDICTIVE VIDEO QOE MANAGEMENT

This Section provides an extensive analysis of the state-of-the-art on predictive QoE management. In order to give a comprehensive overview, we have classified the predictive approaches according to the specific aspect of the QoE management loop that they aim to solve. Using Figure 1 as the reference, we have upgraded the QoE management loop to include the predictive nature (Figure 3).

In the remainder of this Section, the different aspect-related approaches are presented. While Section III-A deals with approaches using prediction for QoE modeling, Section III-B focuses on predictive QoE optimization and, Section III-C discusses the usage of ML to enhance the QoE monitoring.

A. Client-based Predictive QoE models

The purpose of video QoE models is to assess the degree of degradation of the received video data as fast as possible (in real-time or near real-time) and, in the most accurate manner (i.e. most in-line with the human perception of the service). From the accuracy point of view, the legitimate judges of visual quality are the humans, whose opinion can be gathered through subjective analyses [6]. The subjects of such tests are presented stimuli (*i.e.* impaired video sequences), which they rate typically using the Mean Opinion Score (MOS) [28]. This type of analysis is, however, incompatible with real-time

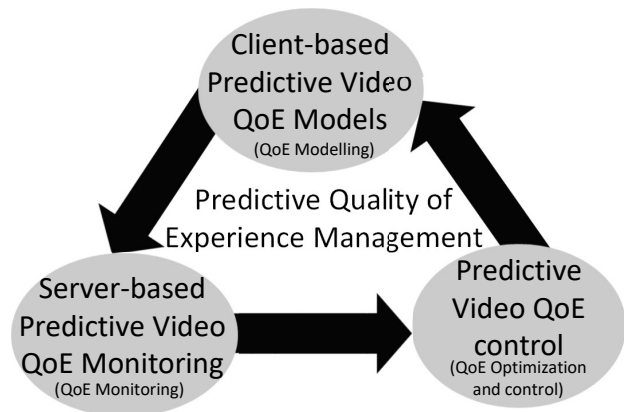


Fig. 3: General block diagram to integrate prediction within the QoE management loop.

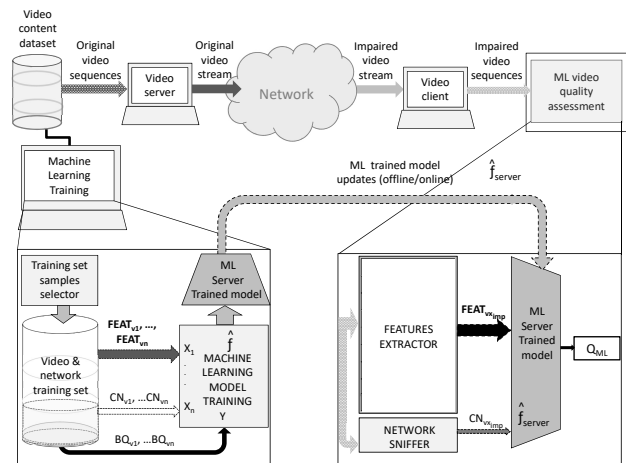


Fig. 4: General block diagram to perform predictive Video Quality Assessment.

assessment, due to its time-consuming nature. Thus, in the last years, a great effort has been put on to developing a number of new objective quality metrics which could provide a valid alternative to subjective evaluation.

Objective metrics are classified, according to the material they require in order to perform the assessment, into three categories, namely Full Reference (FR), Reduced Reference (RR) and No Reference (NR) [29]. FR metrics, which perform a frame-to-frame comparison between the original and the received (impaired) data, have shown the best performance in terms of accuracy to human perception. Examples of such metrics are the Structural Similarities (SSIM) [30] or the Video Quality Model (VQM) [31]. Yet, these metrics require access to the original material and, they tend to be very computationally heavy. Thus, they are unfit for real-time evaluation and better suited for benchmarking purposes. RR and NR metrics, on the other hand, perform their assessment only based on the received material and the network conditions. Therefore, they are the most adequate solution, at least in terms of time-liness and computation efficiency, even allowing deployment

in thin devices, such as smart phones [32], [33]. However, conventional RR and specially NR are not able to provide a sufficiently accurate assessment across the broad range of video conditions (e.g., video types, encoding algorithms, frame rate, bitrate, etc.). Research threads such as [34] or [32] have shown very low correlation of simple RR and NR techniques with the human perception of video services.

Per contra, ML tools have demonstrated to provide the required enhancement of the quality assessment in real-time client based metrics. Figure 4 presents a simplified block diagram of the working principles of this type of prediction based quality assessment algorithms [62]. At service launch, the service provider will already have a representative video type set (e.g., sport, action movies, cartoons, and so forth); thus an initial prediction model can be constructed (and made available to the client side). When a completely new video type is added, the prediction model will be less accurate. Yet, over the time the model will be updated based on new types and, what is more important, the chances of getting new video types will rapidly diminish. The server performs the updates of the model in an offline manner, meaning that it runs a process in the background in which the ML model is trained with the available video samples and new models (\hat{f}_{server}) are uploaded to the clients (on a continuous or periodic basis). On the other end of the transmission link, the video client employs the ML model trained by the server, to generate its prediction-based quality metric (Q_p). During a streaming session, the client characterizes the incoming video in terms of the features required as input to the model, matching this information against the prediction model to generate the quality estimation. Given the functioning mechanism of the prediction system, i.e. training based on prior information saved in the server, SL models will be used the ML class selected, as we will see when discussing the different approaches. Only in the cases when the benchmark was not available, it would be possible to use UL or UDL techniques to enhance the accuracy of the assessment.

The performance of the predictive quality assessment model depends on several characteristics. First, as we introduced in the previous Section, different kinds of ML approaches have their own advantages and disadvantages. This circumstance makes them better or worse fitted to model the problem at hand. For example, while a Fast Forward neural network seems to be the best option to model a system with all the ground truth data, it becomes unfit for a scalable solution (where the training samples need to be labeled before generating the model). The second characteristic is the utilized benchmark. This includes two things: a) the ground truth quality used (in the case of a SL approach); and b) the quality measurement used to assess the accuracy of the model. Third, selecting the features that better characterize the video streams, are effective in the ML training process (in the server) and, ultimately, generate an accurate quality metric (in the clients) is an important decision. In order to keep the calculation of the input features as fast and simple as possible, the predictive QoE models tend to use low-complexity NR features (which can be calculated in real time and with only the client material provided) to input to their models.

With all these performance characteristics in mind, the remaining of this Section reviews the most significant learning based approaches that have appeared in the last 15 years, starting from the first one introduced in 2002 [8]. In Table IV each of the approaches main characteristics as well as their performance evaluation are summarized. To provide a constructive order, the analysis both in the table and the text is performed chronologically.

In the last fifteen years, several researchers have explored the machine learning path in order to improve both the generality and accuracy of their client-based video quality models. Already in 2002, Gastaldo et al. introduced one of the first methods to estimate the video quality using artificial neural networks [8]. They proposed the use of circular back-propagation networks (based on bitstream layer parameters) in order to mimic the users perception of compressed MPEG2 videos. Their approach showed promising results on a 12-video dataset from the motion picture expert group (MPEG). Their study focused on video distortions deriving merely from compression. Also in 2002, Mohamed et al. [36] explored the capabilities of random neural networks for constructing a video quality metric capable of continuous quality monitoring and measuring. They used both bitstream parameters (bitrate, frame rate and ratio of encoded intra to inter macroblocks) and network conditions (packet loss and burst size) as input to the network, and tested on a single low resolution video sequence. They obtained promising results. Yet, since only one video was used, their algorithm was not considering content, implying a lack of generality.

Also working on compressed videos, Le Callet et al. [39] employed an interesting convolutional neural network as a Reduced Reference (RR) method to allow a continuous-time quality estimation and scoring of the video. Their method (as any RR metric) requires the transmission of features extracted from the original video together with the video under scrutiny. El Khatabi et al. [41] also used Neural Networks (Multilayer Perceptron) to predict subjective quality in QCIF and CIF compressed videos. Their inputs consisted on a set of 8 pixel features, making their approach independent from encoding. Their approach was only proven reliable for compression degradation. In addition, they do not report quantitative levels of accuracy.

In [42], the authors investigated and modeled impairment visibility in HD H.264/AVC encoded video sequences using decision trees. In their paper, they report that it is possible to predict the visibility of different impairments just by a limited number of parameters extracted from the bitstream (39 parameters in total).

Narwaria et al [44] evaluated the performance of Support Vector Regression on two video databases against eight different visual quality predictors. The results show a significant improvement in prediction accuracy.

Khan et al [9] proposed a linear regression model based on bitstream and network characteristics. They presented very strong correlation values for videos subjected to a simulated (NS2) UMTS scenario. To our knowledge, this is the first approach that tested with simulated impaired videos instead of the synthetic solutions. Konuk et al [47] also made use of

TABLE IV: Comparative table of using ML (SL and UL) for video quality prediction. Approaches are organized chronologically.

Approach	ML Tech.	Input Feat.	BM	Eval.	Performance	Distort.	App.
Gastaldo et al. [8] (2002)	Circular Backprop. Neural Networks [35](SL)	Bitstream: 24	MOS	12 MPEG-2	$LCC \geq 0.95$ (low motion)	Compr.	MPEG-2
Mohamed et al. [36] (2002)	Random Neural Network [37](SL)	bistream: 3 Network: 2	MOS	1 video (94 cond.)	High correlation	Compr.& Trans. (synth.)	H.263
Le Callet et al. [39] (2006)	Convol. [38]& Time-delay [40] Neural Networks (SL)	Frame-based: 4	DMOS	17 MPEG-2 (4 seq.)	$LCC \geq 0.92$	Compr.	MPEG-2
El Khattabi et al [41] (2011)	Multilayer Perceptron (MLP) [11](SL)	Frame-based:8	MOS	9 QCIF 11CIF	Same quality interval	Compr.	H.263
Staelens et al [42] (2012)	Decision Trees (SL)	Bitstream: 39	MOS	8 H.264/AVC (48 imp. scen.)	$PCC \geq 0.85$	Compr.& Trans. (Synth.)	H.264/AVC
Narwaria et al. [44] (2012)	Support Vector Regression (SVR)(SL)	Singular value Decomposition (per block) [46]	DMOS MOS	LIVE [43] EPFL [45]	$PCC_{LIVE} \geq 0.75$ $PCC_{EPFL} \geq 0.9$	Compr.& Trans. (synth.)	CIF, QCIF, MPEG-2 MPEG-4
Khan et al. [9] (2012)	Linear& Logarithmic Regression (SL)	Bitstream: 2 Network:2	MOS	LIVE [43] 6 H.264/AVC (48 UMTS)	$PCC_{LIVE} \geq 0.7$ $PCC_{NS2} \geq 0.93$	Compr.& Trans. (sim.)	H.264
Konuk et al. [47] (2013)	Polynomial Linear Regression (SL)	Frame-based: 2 Bitstream:1 Network: 1 (PLR)	DMOS	LIVE [43]	$PCC \geq 0.8$	Compr.& Trans. (synth.)	H.264
Staelens et al. [48] (2013)	Genetic programming based symbolic regression [49](SL)	Bitstream: 34 (8)	MOS	8 H.264/AVC	$PLCC \geq 0.88$	Compr.	H.264/AVC
Zhu et al. [51] (2015)	Multilayer Neural Networks [53] (SL)	Frame-based: 6 DCT transform extr. Temporal polling	MOS	IRCCyN [50] HDTV Pool [52] LIVE mobile [43] LIVE [43]	$LCC \geq 0.75$	Compr.& Trans. (synth.)	MPEG-2 & H.264
Sogaard et al. [54] (2015)	Support Vector Regression (SL)	Codec, I-Frame, Quantization and PSNR pred.	MOS	LIVE [43]	$PLCC \geq 0.9$	Compr.& Trans. (synth)	MPEG-2 & H.264/AVC
Shahid et al. [56] (2015)	(SL) Feedforward Neural Networks [53](SL)	Bitstream:52 (43)	PSNR VQM	30 HEVC [55]	$PCC_{PSNR} \geq 0.98$ $PCC_{VQM} \geq 0.94$	Compr.	HEVC
Pandremmenou et al. [57] (2015)	Least Absolute Shrinkage and Selection Operator (LASSO) [58](SL)	Bitstream: 46 (13,15,12)	MOS SSIM VQM	16 H.264/AVC (4 orig.)	$PCC_{MOS} \geq 0.9$ $PCC_{SSIM} \geq 0.99$ $PCC_{VQM} \geq 0.88$	Compr.& Trans. (synth.)	H.264/AVC
Huang et al [59] (2017)	Elastic Net [60](SL)	Pixel level, quantiz. PSNR pred.	MOS	LIVE [43](train) SJTU(test)	$SRCC \geq 0.9$	Compr.	HEVC
Torres Vega et al. [62] (2017)	9 ML models (LR,RT,ERT, SVR,NN,...)(SL)	Frame:6 Bitstream:3 Network:1	VQM	LIMP [61]	$PCC_{ERT} \geq 0.98$	Compr.& Trans. (emul.)	H.264/MPEG-4
Torres Vega et al. [64] (2017)	Restricted Boltzmann Machines(RBM)(UL)	Frame:4 Bitstream:4	VQM	ReTRIEVED [63] LIMP [61] (emul.)	$PCC_{RET} \geq 0.85$ $PCC_{LIMP} \geq 0.88$	Compr.& Trans.	MPEG-2& H.264/MPEG4

linear regression tools based on independent features extracted from spatial and temporal quantities derived from the video packet losses, bit rate, and spatio-temporal complexity. They report correlations higher than 0.8 on the LIVE Video quality database [43].

Staelens et al. [48] presented an NR video quality estimation method which uses a symbolic regression framework trained on a large set of parameters extracted from the codec. They obtained very high correlation with subjective tests for H.264 compressed streams.

Zhu et al. [51] proposed the use of neural networks and features extracted from the analysis of Discrete Cosine Transform (DCT) coefficients of each decoded frame from a video

sequence to predict its quality. Their approach showed good correlation results in compressed videos of four different well-known datasets. However, the complexity of the approach makes it unsuitable for real-time deployments. Similar principles were proposed in [54] by using features extracted from specific codecs (MPEG or H.264/AVC), the analysis of DCT coefficients, the estimation of the quantization level used in the I-frames to measure quality of videos distorted through the compression process. They show higher correlation to subjective studies than some state-of-the-art metrics (FR, RR and NR) making this a very promising solution for H.264/AVC compressed streams.

Shahid et al. [56] proposed a model combining different

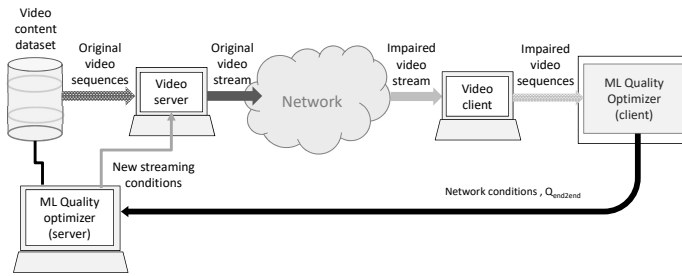


Fig. 5: General block diagram to perform prediction for optimization and control of the video QoE management process.

bitstream-layer features using an Artificial Neural Network to estimate the quality. They tested their method on compressed videos and benchmarked against PSNR. Pandremmenou et al. [57] employed the Least Absolute Shrinkage and Selection Operator (LASSO) regression method for assessing the accuracy of bitstream parameters to full reference metrics and subjective analysis in videos affected by compression and synthetic impairments.

The impairments derived from compressing videos into HEVC is the focus of the NR metric presented by Huang et al. [59]. They proposed to model Elastic Nets [60] with pixel level features to assess the perceived degradation when HEVC compression. They trained their network on the LIVE [43], testing it on the SJTU videoset. They obtained a 90% spearman correlation to subjective studies. They show a very promising approach for pinpointing the degradation inflicted on videos by the HEVC videos.

In our previous work [62], [64] we proposed to use a set of 8 NR features (on the pixel, the bitstream and the network) to explore the capabilities of a broad range of machine learning models. In [62] we put to test 9 models (ranging from linear regression to support vector machines) and assessed the performance of our NR predictive method benchmarking against VQM [31] in a network impaired video set. We obtained accuracies close to 100% with ensemble regression trees. In addition, in [64] we used unsupervised learning techniques, namely restricted Boltzmann machines, to predict the degradation of videos inflicted by networks. We obtained an accuracy higher than 80% while not requiring a ground truth to train our model (given the unsupervised essence of the approach).

B. Video Quality Optimization and Control

Another fundamental element of the QoE management loop is the control and optimization of quality. This element of control is present in multiple streaming techniques, especially the ones based on adaptive streaming. Adaptive video streaming is a well-known concept adopted in several video streaming services. The main function of an adaptive video streaming service is to provide the same content as a set of video streams encoded with different parameters (e.g. bitrate, resolution). The client can dynamically adapt to the video stream that best

matches the available network resources (network bandwidth) in order to maximize the user QoE. A conventional approach to adaptive video streaming makes use of deterministic heuristics to dynamically adapt the required quality on the basis of the perceived network and device state. Such approaches are designed to fit specific network configurations. For this reason, these approaches do not fit a large set of applications running on different network condition.

Per contra, predictive-based adaptation algorithms can prevent buffer underflow on the client device (e.g., on the decoder), which would otherwise lead to unrecoverable QoE impairments. This method would require the client to rely on intelligent models that can anticipate butter starvation events and act proactively.

ML can be employed to develop proactive QoE control loops, as depicted in Figure 5. Both server and client adopt predictive tools. While the client agent uses its tool to sense the received quality and provide a feedback loop, the server agent employs this feedback to perform the required adjustments to the streaming service. Given the required real-time adaptability to new conditions and the feedback loop, RL methods are best suited to perform the control tasks.

RL methods have recently been proposed for adaptive control of video streaming bitrates with the objective of improving the quality of experience, by selecting the video representation that maximizes the overall user's QoE. In the RL paradigm, this objective is achieved by maximizing the long term reward. Table V shows a summary of works that exploit RL for controlling QoE in video streaming applications.

An approach based on RL and providing a flexible solution to the problem of adaptive streaming was proposed in [65]. The main contribution of this research is the proposal of a QoE estimation function incorporating the subjectively perceived quality, the duration and frequency of playback freeze. The framework considers an adaptive streaming application over HTTP. A frame size distribution was computed for the simulation of the proposed method.

A dynamic learning behavior on how to respond to the network condition was presented by Claeys et al. in [66]. In particular, an adaptive RL strategy based on Q-learning was proposed. The technique exploits the knowledge of a set of video quality parameters (current video quality, oscillation in quality during playout, buffer starvation) for a tunable reward function and for the evaluation of different aspects of QoE. The experimental evaluation was conducted using NS-3, comparing the RL method with Microsoft ISS Smooth Streaming service (MSS). Their results showed a nearly 11% improvement of the proposed quality metric of the RL implementation with respect to the reference streaming service in a simulated environment.

The same authors proposed an improved solution in [10]. There, a frequency adjusted Q-learning approach was introduced for improving the QoE in strongly variable environments. This was done by means of an initialization phase of the Q-Tables aiming at adding a preliminary knowledge of

TABLE V: RL for video quality of experience control. Approaches are listed chronologically.

Approach	ML Technique	Quality	Eval. set	Video streams	Experimental setup	Performance	Application
Menkovski et al. [65] (2013)	SARSA (RL)	QoE Model	Simulated source	1: Variabl Bitrate (VBR)	Simulation (Matlab)	Not available	HTTP Adaptive Streaming
Claeys et al. [66] (2013)	Q-learning (RL)	QoE model	Big Buck Bunny (1)	7: Constant Bitrate (CBR)	Simulation (NS3)	11% improv.	HAS
Claeys et al. [10] (2014)	Q-learning (RL)	QoE model	Big Buck Bunny(1)	7:Constant Bitrate (CBR)	Simulation (NS-3)	18% improv.	HAS
Petrangeli et al. [67] (2014)	Multi-agent Q-learning (RL)	QoE model	Big Buck Bunny (1)	7: Constant Bitrate (CBR)	Simulation (NS-3)	60% improv. MSS	HAS
Torres Vega et al. [68] (2015)	Q-learning (RL)	Quality Bitrate	Shields (1, LIVE [43])	10:Constant Bitrate (CBR)	Lab test	Max. quality selection	DASH
Chiarotti et al. [21] (2016)	Post decision state (RL)	SSIM	EvalVid CIF (5) [69]	9:Constant Bitrate (CBR)	Simulation (Matlab)	Not available	DASH
Yu et al. [70] (2017)	MDP (RL)	QoE model	Big Buck Bunny (1)	4: Variable Bitrate (VBR)	Simulation (Matlab)	27% improv.	DASH

the adaptive streaming over HTTP behavior. The simulation was again conducted using NS-3 and benchmarked against the Microsoft ISS Smooth Streaming Service. The results showed an improvement from 11% to 18% of the proposed quality metric of the RL implementation with respect to the reference streaming service.

Petrangeli et al. [67] built on the two previous approaches to propose a Multi-agent Q-learning video streaming player. A proxy is in charge of facilitating the coordination among clients. The algorithm not only learns and adapts its policy depending on the network conditions but also does so in a fair manner without any explicit communication among agents. They evaluated their approach in NS-3 benchmarking both against MSS and against the one agent Q-learning variant [10]. Their algorithm achieves 60% and 48% improvement on fairness compared to the MSS and the one agent Q-learning variant respectively.

With the inspiration of the work performed by both Claeys and Petrangeli, in [68] we proposed a Q-learning algorithm for Android wireless devices. Thus, we could study the RL based adaptive streaming beyond simulated conditions, on real-world devices. Our results showed the great potential of using RL in lightweighted devices as well as pinpointing at the most affecting impairments for adaptive streaming services.

An online adaptation logic for Dynamic Adaptive Streaming over HTTP (DASH) [71] clients was introduced by Chiarotti et al. [21]. In it, each client chooses a streaming adaptation process that maximizes the long-term expected reward. A Markov Decision Process (MDP) optimization was used for modeling the selection of the optimal stream representation. The expected reward for an action is defined as a combination of the decoded quality, the quality fluctuations, and the re-buffering events triggered during the playback. Each client selects the representation that maximizes the long-term expected reward. In particular, the research presented in [21] aims at addressing the problem of large quality variation and playback underflow buffer at the receiver which affects the quality of experience of the final user. The proposed MDP model and

learning technique resulted in a fast-learning technique with respect to previous methods such as [10].

Yu et al. [70] introduced a method for adapting variable bitrate video streaming in [70]. The bitrates of incoming segments are sent in advance to the client carried by the media presentation description XML document of DASH standard [71]. The adaptation problem tries to maximize the global QoE by dividing this problem in sub-optimal problems to meet the real time constraints.

C. Predictive Server-based Monitoring Methods

In addition to predictive modeling and optimization, monitoring the services (from the server side), classifying and understanding the behavioral patterns of different videos is a fundamental part of the management loop. Making use of a good and accurate monitoring tool will heavily influence the performance of the models and control loops. This task has traditionally been performed by means of static, adaptive techniques, such as setting quality thresholds. By means of ML tools, the monitoring and classification capabilities of the server can be improved considerably.

Figure 6 presents a simplified block diagram of how a predictive monitoring tool is to be deployed and used in the video streaming set-up that already introduced in the previous two Sections. The video streaming server is connected to the client through the network and in the client a real-time assessment model (same or similar to the examples described in Section III-A) is running. This assessment is included in the feedback loop that goes back to the video server. The ML Video Monitoring tool, classifies the incoming information based on the prior history of the system, the video content types available, etc. The output of this monitoring tool (a classification, feature extraction, etc.) can be used to improve the prediction model in the client (Section III-A) and to optimize and control the management loop (Section III-B).

ML tools employed to perform the monitoring have to be very powerful classifiers. As such SL Classification and UL are

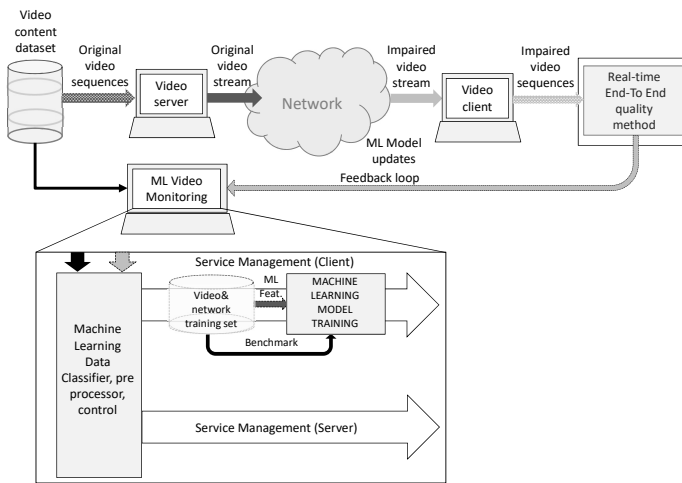


Fig. 6: General block diagram to use prediction to enhance the monitoring of the quality. This process usually happens on the server side based on the feedback provided by the clients combined with the registered prior conditions.

TABLE VI: ML-based approaches to enhance the monitoring in the Server. The approaches appear in alphabetical order.

Approach	ML technique	Input param	Purpose
Menkovski et al. [72] (2010)	Decision Trees (SL)	Bitstream feats.	Determine acceptability of videos
Malekmohamadi et al. [73] (2012)	K-means& Discriminant Analysis (UL)	Spatio-Temp.& Depth	Classify 3D videos for quality
Khan et al. [9] (2012)	K-means (UL)	Content Descriptors	Cluster video content
Wang et al. [74] (2014)	K-means (UL)	Frame level	Extract features for video quality
Jiang et al. [75] (2016)	K-Means (UL)	Technical Quality (TQ)	Isolate unreliable MOS
Konuk et al. [76] (2016)	K-Means (UL)	Spatio-Temporal Feat.	Cluster Video Content

the best suited techniques. Table VI shows the most significant predictive monitoring approaches developed in the last 10 years. Unlike the previous two cases, in this Table we have focused on what each of the approaches aims to achieve, as the performance can only be evaluated by means of the accuracy of the client model that the system uses the clustering for.

Predictive monitoring techniques have focused their attention on classification for two different tasks: 1) enhancement of the prediction capabilities of QoE models deployed in the client; 2) direct classification of users' behavioral patterns.

In general, the clustering technique K-Means is the most used technique to enhance the prediction capabilities of QoE models. Malekmohamadi et al. [73] proposed to use the combined action of K-means and discriminant analysis to classify 3D videos according to their depth's spatial and temporal information. Then, for each of the generated categories, a parametric equation adjusted the quality index. They showed good performance for a small dataset of 3D videos. Khan

et al. [9] pre-classified the videos based on their content descriptors to then apply linear regression to the categories. In a similar approach, Konuk et al. [76] proposed to use K-means to cluster video content according to bitstream and spatio-temporal content descriptors. Finally, Wang et al. in [74], also employed k-means to extract video features that would afterwards be used to measure quality.

Regarding the behavioral prediction task, already in 2010 Menkovski et al [72] made use of binary decision trees to determine if after the effect of the networks, the videos were within the acceptability range (if the users would find them acceptable). In order to classify the impaired videos, they measured the impact of the network on bitstream parameters. Jiang et al. went one step further on the subjective assessment classification. In [75], they propose to use clustering (K-Means) to differentiate between unreliable and reliable participants in a subjective study. In their approach, reliability was defined in terms of criteria such as consistency of rating and ability to distinguish between qualitative differences in level of impairments. Their results showed that clustering a data set that is augmented with unreliable pseudo-participants can provide a new and improved perspective on individual differences in video QoE assessment.

IV. DISCUSSION: WHICH MACHINE LEARNING METHOD IS BEST FITTED?

In the previous Section we have analyzed the different predictive approaches, classifying them according to the element of the QoE management loop they focused on. Based on the surveyed state-of-the-art and the problem they aim to solve, we have hinted at the type of ML better suited to solve each problem.

Predictive client-based video quality modeling (Section III-A) is mostly performed by means of SL metrics, due to the fact that it heavily benefits from offline training and benchmarking. In an extreme case, when either the benchmark is missing or the solution is not scalable, it is possible to use UDL (such as [64]).

The control loop (Section III-B) requires real-time feedback updates. Thus, RL is the best solution. These techniques enable video and network real-time adaptation, such as the one needed by adaptive streaming and MPEG-DASH solutions.

Finally, predictive monitoring (Section III-C) benefits from classification, grouping and pattern recognition techniques. Therefore, UL or Classification SL are the solutions of choice, due to the fact that these techniques allow for fast classification of the videodata according to its characteristics.

In addition to the problem to solve, the dataset's composition, size and characteristics will heavily influence the choice in terms of type of ML class and particular approach. Two golden rules are to be kept in mind for selecting the ML model to apply:

- 1) *Understand your dataset, its strengths and weaknesses.* When trying to solve a QoE management problem by means of prediction, it is often declared that the more data is available for training, the better and more accurate the predictive model will be, regardless of the

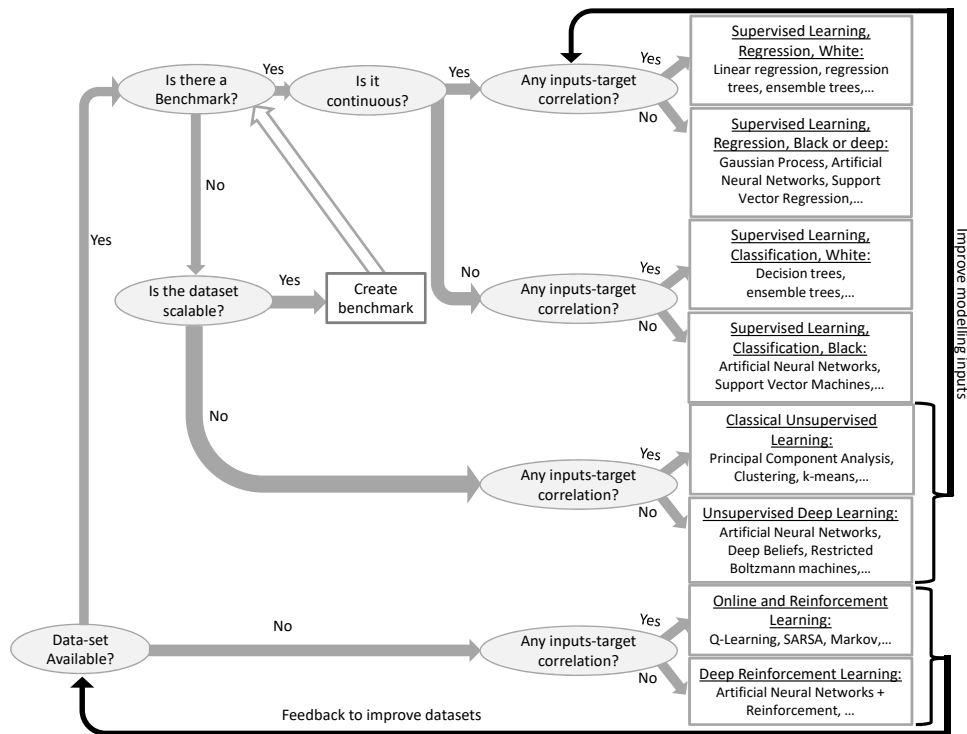


Fig. 7: Flow chart of which ML method to use depending on the available data, the complexity of the problem and the expected performance. General block diagram to perform prediction while monitoring the quality.

model under-scrutiny. This is a wrong assumption since each predictive model learns in a different manner. This means that what may be a strength of a model could represent the weakness of another.

- 2) *The model, the simpler the better.* As shown in this paper, the current state-of-the-art shows a tendency to select the most complicated models to perform prediction. This, while providing the “fanciest” solution, can fall onto a dangerous path, in which it becomes very difficult to visualize and control what the model does with the input data. This means not understanding why the model works with certain samples (or predicted qualities) but fails to provide an accurate assessment with others. In contrast, using simpler but accurate models (whenever the dataset allows) brings with it easier ways to spot the reasons behind inaccuracies and to provide hints to solve issues.

Based on these two guidelines, Figure 7 aims to illustrate the reasoning to be followed in order to select the ML method. When a historic dataset is available, the first step is to see what it looks like, which type of video characteristics are present and the type of benchmark that was used. In the worst-case scenario in which no dataset is available and, there is no easy manner to generate a training set (for example, if there is no prior knowledge of the system or lack of accessibility), the only possible option is online learning, *i.e.* RL. Within the broad range of RL methods, the selection should be done according to the second rule (the simpler the better), but always taking into account the purpose (the output to predict) and the relation of the output to the possible inputs, *i.e.*, how

easy it is to model the target given the input features. If it is possible to find some correlation (even if reduced) between the inputs and target output (such as in the case of video quality prediction based on bitrates and network delays), it is best to start from a simple model (such as Q-Learning or SARSA). Then, based on the performance, the recommendation is to gradually increase the model’s complexity and to adjust the input parameters until satisfactory results are achieved. The outputs and input conditions obtained from the RL process, could, in turn, be used as the starting dataset to solve other prediction problems.

Considering the cases in which a prior dataset is available, either from historic data or collected by means of RL, SL models offer a good trade-off between simplicity and accuracy. The nature of the benchmark (whether discrete or continuous) will lead to either classification or regression approaches. Again, depending on the ease to model the target output given the inputs (*i.e.*, if it is possible to correlate certain behavioral patterns of the inputs to the output), the idea is to start from simple models (white boxes, such as linear regression, regression or decision trees). White boxes have shown very good performance in the literature (Section III-A [47], [62]). If the predictions were still far from the set targets, the inputs could be tuned, and more complicated approaches (black boxes) could be tried out.

If a suitable benchmark is not available, it is best to first ponder whether it would be feasible to create a benchmark, for at least a portion of the training set (either by means of subjective evaluation or RL) and, if the solution would be scalable. This means that by generating the benchmark

for a certain portion of the training set, it would be possible to extrapolate the ground-truth of not benchmarked data belonging to the same system (the remaining portion of the dataset or new samples generated after training). Creating the benchmark, would allow the usage of SL techniques that tend to be easier to tune. Alternatively, UL techniques enable clustering and classification of the data, which is very useful for understanding the effect of impairments on the videos and assessing whether the provided quality can reach acceptable levels. It is also possible to combine UL with Deep Learning, which has already shown promising performance when aiming to predict quality in the absence of benchmarks, as we showed in [64]. The UL outputs could in turn be used to understand the nature and behavioral patterns of the inputs, which is necessary to construct better suited models.

V. CONCLUSION

Quality of Experience management has enormous potential in the context of real-time assessment, monitoring and control of video streaming services. Yet, conventionally this is performed through reactive control and is based on static models. This defeats the very purpose of user-centric management. In this article, we have reviewed proactive QoE management techniques, with particular emphasis on prediction-based methods founded on machine learning. This avenue has been explored only recently, with studies dating no longer than fifteen years. We identify the key challenges and research lines, capturing progress in the areas of QoE modeling, monitoring, and control. For each case, we consider the best suited ML approaches, providing guidelines to match QoE components to ML options.

What emerges is not only a wealth of new possibilities enabled by intelligent methods, but also a substantial immaturity of the tools and platforms available. The lack of large-scale datasets and benchmarks represents a major research and development hurdle, as it is often difficult to carry out comparative studies and reach generalized conclusions. This explains why considerable efforts are still ongoing to pursue a combined QoS/QoE management in large-scale video streaming services.

ACKNOWLEDGMENT

This work has been partially supported by the Italian Ministry of University and Research (MIUR), within the Smart City framework (project: PON04a2_00381 “CAGLIARI2020”).

REFERENCES

- [1] L. Atzori, C. W. Chen, T. Dagiuklas, and H. R. Wu, “QoE management in emerging multimedia services,” *IEEE Communications Magazine*, vol. 50, no. 4, pp. 18–19, 2012.
- [2] D. Mocanu, A. Liotta, A. Ricci, M. Vega, and G. Exarchakos, “When does lower bitrate give higher quality in modern video services?” in *Network Operations and Management Symposium (NOMS), 2014 IEEE*, Krakow, Poland, May 2014, pp. 1–5.
- [3] T. Hossfeld, R. Schatz, M. Varela, and C. Timmerer, “Challenges of QoE management for cloud applications,” *IEEE Communications Magazine*, vol. 50, no. 4, pp. 28–36, April 2012.
- [4] S. Barakovic and L. Skorin-Kapov, “Survey and Challenges of QoE Management Issues in Wireless Networks,” *Journal Comp. Netw. and Commun.*, vol. 2013, pp. 165 146:1–165 146:28, 2013.
- [5] M. Varela, L. Skorin-Kapov, K. De Moor, and P. Reichl, “QoE - Defining a User-centric Concept for Service Quality,” in *Multimedia Quality of Experience (QoE): Current Status and Future Requirements*, C. Chen, P. Chatzimisios, and T. Dagiuklas, Eds. John Wiley&Sons, 2015, ch. 2.
- [6] P. Le Callet, S. Moeller, A. Perkis, and eds., “Qualinet white paper on definitions of Quality of Experience,” in *COST Action IC 1003*, Lausanne, Switzerland, 2012.
- [7] K. De Moor, W. Joseph, I. Ketykó, E. Tanghe, T. Deryckere, L. Martens, and L. De Marez, “Linking Users’ Subjective QoE Evaluation to Signal Strength in an IEEE 802.11B/G Wireless LAN Environment,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2010, pp. 6:1–6:10, Apr. 2010.
- [8] P. Gastaldo, S. Rovetta, and R. Zunino, “Objective quality assessment of mpeg-2 video streams by using cbp neural networks,” *IEEE Transactions on Neural Networks*, vol. 13, no. 4, pp. 939–947, Jul 2002.
- [9] A. Khan, L. Sun, and E. Ifeachor, “QoE Prediction Model and its Application in Video Quality Adaptation Over UMTS Networks,” *IEEE Transactions on Multimedia*, vol. 14, no. 2, 2012.
- [10] M. Claeys, S. Latré, J. Famaey, T. Wu, W. Van Leekwijck, and F. De Turck, “Design and optimisation of a (FA) Q-learning-based HTTP adaptive streaming client,” *Connection Science*, vol. 26, no. 1, pp. 25–43, 2014.
- [11] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, 1st ed. Springer, Oct. 2007.
- [12] D. Freedman, *Statistical Models : Theory and Practice*. Cambridge University Press, August 2005.
- [13] J. R. Quinlan, “Learning efficient classification procedures and their application to chess end games,” in *Machine Learning. An Artificial Intelligence Approach*, 1983, pp. 463–482.
- [14] D. W. Opitz and R. Maclin, “Popular ensemble methods: An empirical study,” *J. Artif. Intell. Res. (JAIR)*, vol. 11, pp. 169–198, 1999.
- [15] R. Turner, “A model explanation system,” in *Black Box Learning and Inference NIPS Workshop 2015*, Montreal, Canada, Dec. 2015.
- [16] G. Matheron, *Principles of geostatistics*. Society of Economic Geologists, 1963, vol. 58.
- [17] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.
- [18] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.
- [19] R. Bellman, “A markovian decision process,” *Journal of Mathematics and Mechanics*, pp. 679–684, 1957.
- [20] A. Ruszczyński, “Commentary:post-decision states and separable approximations are powerful tools of approximate dynamic programming,” *INFORMS Journal on Computing*, vol. 22, no. 1, pp. 20–22, 2010.
- [21] F. Chiariotti, S. D’Aronco, L. Toni, and P. Frossard, “Online learning adaptation strategy for dash clients,” in *Proceedings of the 7th International Conference on Multimedia Systems*. ACM, 2016, p. 8.
- [22] D. Hebb, *The Organization of Behavior*. Wiley & Sons, 1949.
- [23] I. J. Goodfellow, “NIPS 2016 tutorial: Generative adversarial networks,” *CoRR*, vol. abs/1701.00160, 2017.
- [24] K. O. Bowman and L. R. Shenton, “Estimator: Method of Moments,” *Encyclopedia of statistical sciences*, pp. 2092–2098, 1998.
- [25] H. Abdi and L. Williams, “Principal component analysis,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.
- [26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [27] H. Larochelle and Y. Bengio, “Classification using discriminative restricted boltzmann machines,” in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML ’08. New York, NY, USA: ACM, 2008, pp. 536–543.
- [28] V. Menkovski and A. Liotta, “Adaptive psychometric scaling for video quality assessment,” *Signal Processing: Image Communication*, vol. 27, no. 8, pp. 788 – 799, 2012.
- [29] S. Chikkerur, V. Sundaram, M. Reisslein, and L. J. Karam, “Objective Video Quality Assessment Methods: A Classification, Review, and Performance Comparison,” *IEEE Transactions on Broadcasting*, vol. 57, no. 2, pp. 165–182, 2011.
- [30] Z. Wang, L. Lu, and A. C. Bovik, “Video quality assessment based on structural distortion measurement,” *Signal Processing: Image Communication*, vol. 19, no. 2, pp. 121–132, Feb. 2004.
- [31] M. H. Pinson and S. Wolf, “A new standardized method for objectively measuring video quality,” *IEEE Transactions on broadcasting*, vol. 50, no. 3, pp. 312–322, Sep. 2004.

- [32] M. Torres Vega, V. Sguazzo, D. C. Mocanu, and A. Liotta, "An experimental survey of no-reference video quality assessment methods," *Int. J. Pervasive Computing and Communications*, vol. 12, no. 1, pp. 66–86, 2016.
- [33] M. Shahid, A. Rossholm, B. Löfvström, and H. Zepernick, "No-reference image and video quality assessment: a classification and review of recent approaches," *EURASIP J. Image and Video Processing*, vol. 2014, p. 40, 2014.
- [34] P. Paudyal, F. Battisti, and M. Carli, "Impact of video content and transmission impairments on quality of experience," *Multimedia Tools and Applications*, vol. 2016.
- [35] S. Ridella, S. Rovetta, and R. Zunino, "Circular backpropagation networks embed vector quantization," *IEEE Trans. Neural Networks*, vol. 10, no. 4, pp. 972–975, 1999.
- [36] S. Mohamed and G. Rubino, "A Study of Real-Time Packet Video Quality Using Random Neural Networks," *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, vol. 12, no. 12, 2002.
- [37] E. Gelenbe, "Random Neural Networks with negative and positive signals," *Neural Comput.*, vol. 1, pp. 502–511, 1989.
- [38] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, 1998, pp. 2278–2324.
- [39] P. Le Callet, C. Viard-Gaudin, and D. Barba, "A convolutional neural network approach for objective video quality assessment," *Neural Networks, IEEE Transactions on*, vol. 17, no. 5, pp. 1316–1327, Sept 2006.
- [40] M. Schenkel, I. Guyon, and D. Henderson, "On-line recursive script recognition using time delay neural networks and hidden markov models," in *Proceedings of ICASSP '94: IEEE International Conference on Acoustics, Speech and Signal Processing, Adelaide, South Australia, Australia, April 19-22, 1994*, 1994, pp. 637–640.
- [41] H. El Khattabi, A. Tamtaoui, and D. Aboutajdine, "Video Quality assessment Measure with a Neural Network," *International Journal of Computer and Information Engineering*, vol. 5, no. 3, 2011.
- [42] N. Staelens, G. V. Wallendael, K. Crombecq, N. Vercammen, J. D. Cock, B. Vermeulen, R. V. de Walle, T. Dhaene, and P. Demeester, "No-Reference Bitstream-Based Visual Quality Impairment Detection for High Definition H.264/AVC Encoded Video Sequences," *IEEE TRANSACTIONS ON BROADCASTING*, vol. 58, no. 2, June 2012.
- [43] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack, "Study of subjective and objective quality assessment of video," *Trans. Img. Proc.*, vol. 19, no. 6, pp. 1427–1441, Jun. 2010.
- [44] M. Narwaria and W. Lin, "SVD-Based Quality Metric for Image and Video Using Machine Learning," *IEEE Transactions on Systems, Man, and Cybernetics - PART B: Cybernetics*, vol. 42, no. 2, 2012.
- [45] F. Simone, M. Naccari, M. Tagliasacchi, F. Dufaux, S. Tubaro, and T. Ebrahimi, "Subjective assessment of H.264/AVC video sequences transmitted over a noisy channel," in *Proc. 1st Int. Workshop QoMEX*, Jul 2009, pp. 204–209.
- [46] D. Kalman, "A singularly valuable decomposition: The SVD of a matrix," *College Math. J.*, vol. 27, no. 1, pp. 2–23, 1996.
- [47] B. Konuk, E. Zerman, G. Nur, and G. Akar, "A spatiotemporal no-reference video quality assessment model," in *Image Processing (ICIP), 2013 20th IEEE International Conference on*, Sept 2013, pp. 54–58.
- [48] N. Staelens, D. Deschrijver, E. Vladislavleva, B. Vermeulen, T. Dhaene, and P. Demeester, "Constructing a no-reference h.264/avc bitstream-based video quality metric using genetic programming-based symbolic regression," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 23, no. 8, pp. 1322–1333, Aug 2013.
- [49] K. Vladislavleva, K. Veeramachaneni, M. Burland, J. Parcon, and U. O'Reilly, "Knowledge mining with genetic programming methods for variable selection in flavor design," in *GECCO*. ACM, 2010, pp. 941–948.
- [50] Y. Pitrey, M. Barkowsky, R. Pépion, P. Le Callet, and H. Hlavacs, "Influence of the source content and encoding configuration on the perceived quality for scalable video coding," in *SPIE Human Vision and Electronic Imaging XVII*, vol. 8291, no. 54, San francisco, United States, Jan. 2012, pp. 1–6.
- [51] K. Zhu, C. Li, V. Asari, and D. Saupe, "No-reference video quality assessment based on artifact measurement and statistical analysis," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.
- [52] A. K. Moorthy, L. K. Choi, A. C. Bovik, and G. de Veciana, "Video quality assessment on mobile devices: Subjective, behavioral and objective studies," *J. Sel. Topics Signal Processing*, vol. 6, no. 6, pp. 652–671, 2012.
- [53] R. Rojas, *Neural Networks: A Systematic Introduction*. New York, NY, USA: Springer-Verlag New York, Inc., 1996.
- [54] J. Sogaard, S. Forchhammer, and J. Korhonen, "No-reference video quality assessment using codec analysis," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.
- [55] M. Barkowsky, E. Masala, G. V. Wallendael, K. Brunnström, N. Staelens, and P. L. Callet, "Objective video quality assessment - towards large scale video database enhanced model development," *IEICE Transactions*, vol. 98-B, no. 1, pp. 2–11, 2015.
- [56] M. Shahid, J. Panasiuk, G. V. Wallendael, M. Barkowsky, and B. Löfvström, "Predicting full-reference video quality measures using HEVC bitstream-based no-reference features," in *Seventh International Workshop on Quality of Multimedia Experience, QoMEX 2015, Pilos, Messinia, Greece, May 26-29, 2015*, 2015, pp. 1–2.
- [57] K. Pandremmenou, M. Shahid, L. P. Kondi, and B. Löfvström, "A no-reference bitstream-based perceptual model for video quality estimation of videos affected by coding artifacts and packet losses," in *Human Vision and Electronic Imaging*, San Francisco, US, Feb. 2015.
- [58] R. Tibshirani, "The LASSO method for variable selection in the Cox model," *Statist. Med.*, vol. 16, no. 4, pp. 385–395, 1997.
- [59] X. Huang, J. Sgaard, and S. Forchhammer, "No-reference pixel based video quality assessment for HEVC decoded video," *J. Vis. Commun. Image R.*, vol. 43, 2017.
- [60] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Stat. Soc.: Ser. B (Stat. Methodol.)*, vol. 67, pp. 301–320, 2015.
- [61] M. Torres Vega and A. Liotta, "LIMP-Video Quality Database," <https://www.tue.nl/index.php?id=53688>.
- [62] M. Torres Vega, D. C. Mocanu, S. Stavrou, and A. Liotta, "Predictive No-Reference Assessment of Video Quality," *Signal Processing: Image Communication*, vol. 52, pp. 20–32, March 2017.
- [63] F. Battisti, M. Carli, and P. Paudyal, "ReTRiEVED Video Quality Database," <http://www.comlab.uniroma3.it/retrieved.htm>.
- [64] M. Torres Vega, D. C. Mocanu, and A. Liotta, "Unsupervised deep learning for real-time assessment of video streaming services," *Multimedia Tools and Applications*, pp. 1–25, 2017.
- [65] V. Menkovski and A. Liotta, "Intelligent control for adaptive video streaming," in *2013 IEEE International Conference on Consumer Electronics (ICCE)*, Jan 2013, pp. 127–128.
- [66] M. Claeys, S. Latré, J. Famaey, T. Wu, W. Van Leekwijck, and F. De Turck, "Design of a Q-learning-based client quality selection algorithm for HTTP adaptive video streaming," in *Adaptive and Learning Agents Workshop, part of AAMAS2013 (ALA-2013)*, 2013, pp. 30–37.
- [67] S. Petrangeli, M. Claeys, S. Latré, J. Famaey, and F. D. Turck, "A multi-agent Q-Learning-based framework for achieving fairness in HTTP Adaptive Streaming," in *NOMS*. Krakow, Poland: IEEE, May 2014, pp. 1–9.
- [68] M. Torres Vega, D. C. Mocanu, R. Barresi, G. Fortino, and A. Liotta, "Cognitive streaming on android devices," in *in proc. of the 1st IEEE/IFIP IM 2015 International Workshop on Cognitive Network & Service Management*, Ottawa, Canada, May 2015.
- [69] J. Klaue, B. Rathke, and A. Wolisz, "Evalvid - a framework for video transmission and quality evaluation," in *In Proc. of the 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, 2003, pp. 255–272.
- [70] L. Yu, T. Tillo, and J. Xiao, "Qoe-driven dynamic adaptive video streaming strategy with future information," *IEEE Transactions on Broadcasting*, vol. 63, no. 3, pp. 523–534, Sept 2017.
- [71] I. Sodagar, "The mpeg-dash standard for multimedia streaming over the internet," *IEEE Multimedia*, vol. 18, no. 4, pp. 62–67, 2011.
- [72] V. Menkovski, G. Exarchakos, and A. Liotta, "Machine learning approach for quality of experience aware networks," in *Proc. 2nd Int. Conf. INCOS*, Nov 2010, pp. 461–466.
- [73] H. Malekmohamadi, W. A. C. Fernando, and A. M. Kondoz, "Automatic QOE Prediction in Stereoscopic Videos," in *2012 IEEE International Conference on Multimedia and Expo Workshops*, Melbourne, Australia, July 2012, pp. 581–586.
- [74] J. Xu, P. Ye, Y. Liu, and D. Doermann, "No-reference video quality assessment via feature learning," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct 2014, pp. 491–495.
- [75] J. Jiang, P. Spachos, M. H. Chignell, and L. Zucherman, "Assessing unreliability in OTT video qoe subjective evaluations using clustering with idealized data," in *DMIAF*. IEEE, 2016, pp. 235–239.
- [76] B. Konuk, E. Zerman, G. Nur, and G. B. Akar, "Video content analysis method for audiovisual quality assessment," in *2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX)*, Lisbon, Portugal, June 2016, pp. 1–6.