A supervised Machine Learning approach for DASH video QoE prediction in 5G networks

Raza Ul Mustafa
Department of Computer Engineering
and Industrial Automation (DCA)
School of Electrical and Computer
Engineering (FEEC), Unicamp
Brazil
r211580@dac.unicamp.br

Simone Ferlin
Ericsson AB
Sweden
simone.ferlin@ericsson.com

Christian Esteve Rothenberg
Department of Computer Engineering
and Industrial Automation (DCA)
School of Electrical and Computer
Engineering (FEEC), Unicamp
Brazil
chesteve@dca.fee.unicamp.br

Darijo Raca
Faculty of Electrical Engineering,
University of Sarajevo, Sarajevo, BiH
draca@etf.unsa.ba

Jason J. Quinlan
School of Computer Science &
Information Technology,
University College Cork, Ireland
j.quinlan@cs.ucc.ie

ABSTRACT

Future fifth generation (5G) networks are envisioned to provide improved Quality-of-Experience (QoE) for applications by means of higher data rates, low and ultra-reliable latency and very high reliability. Proving increasing beneficial for mobile devices running multimedia applications. However, there exist two main co-related challenges in multimedia delivery in 5G. Namely, balancing operator provisioning and client expectations. To this end, we investigate how to build a QoE-aware network that guarantees at run-time that the end-to-end user experience meets the end users' expectations at the same that the network's Quality of Service (QoS) varies.

The contribution of this paper is twofold: First, we consider a Dynamic Adaptive Streaming over HTTP (DASH) video application in a realistic emulation environment derived from real 5G traces in static and mobility scenarios to assess the QoE performance of three state-of-art Adaptive Bitrate Streaming (ABS) algorithm categories: Hybrid - Elastic and Arbiter+; buffer-based - BBA and Logistic; and rate-based - Exponential and Conventional. Second, we propose a Machine Learning (ML) classifier to predict user satisfaction which considers network metrics, such as RTT, throughput, and number of packets. Our proposed model does not rely on knowledge about the application or specific traffic information. We show that our ML classifiers achieves a QoE prediction accuracy of 87.63 % and 79 % for static and mobility scenarios, respectively.

KEYWORDS

5G, QoE prediction, QoS, Machine Learning, video streaming, DASH

Conference'17, July 2017, Washington, DC, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Raza Ul Mustafa, Simone Ferlin, Christian Esteve Rothenberg, Darijo Raca, and Jason J. Quinlan. 2020. A supervised Machine Learning approach for DASH video QoE prediction in 5G networks. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The steady growth of Internet data services drove the development of third (3G) and fourth (4G) generations of the mobile communications standard. Now, the technology is evolving towards its fifth-generation (5G), motivated by similar traffic demands. 5G [23] is expected to support significantly higher throughput (10 Gbps), 1-millisecond end-to-end over-the-air latency, real-time information processing and transmission, and lower network management operation complexity. In video streaming, HTTP Adaptive Streaming (HAS) is the de-facto choice of popular services such as YouTube and Netflix for Internet video distribution. Fuelled by the recent coronavirus pandemic, mobile video streaming surged, increasing its share by 60% in the first few months of 2020 [9].

With video streaming expected to account for 82% of all network traffic within the next three years [1], compared to previous generations, one of the main challenges in 5G will be the ability to effectively manage the increased growth in traffic and the corresponding Quality of Experience (QoE) demands. Various techniques have been proposed to meet end-users' video QoE, such as splitting traffic into voice, video, and control flows, but these techniques come at the cost of in-network optimization and maintenance [21].

Variation in network Quality of Service (QoS) metrics such as RTT and bandwidth, also play a significant role in determining the end-users' video QoE satisfaction. In HAS, besides network QoS, other factors such as the choice of Adaptive Bitrate Streaming (ABS) algorithm, plays a major role. Thus, the main goal of this paper is the proposal of an in-network mechanism able to estimate end-user video streaming (QoE) experience agnostic to the ABS algorithm while not relying on knowledge about the application. In other words, we do not depend on specific characteristics about the application.

In this work, we select six state-of-the-art ABS algorithms for video quality adaptation, categorized into three main groups: i) Hybrid (Arbiter+ [38] and Elastic [8]), ii) Buffered (Logistic [32] and BBA [14]) and iii) Rate-based (Conventional [16] and Exponential (exponential growth of past throughput)).

We assess the behaviour of the aforementioned state-of-the-art ABS algorithms with godash [29]. *godash* is an open-source headless DASH player written in Google GO, which implements five well-known QoE models from the literature. In this paper we focus primarily on one of the QoE model, namely the ITU-T Rec. P.1203 QoE standard [27, 31] (mode 0 considering metadata only, bitrate, frame rate, and resolution). In addition, *godash* also provides logs for Claey [24], Dunamu [12], Yin [36], and Yu [37]. Finally, we propose an ML-based QoE estimator derived from different classifiers, with a specific focus on 5G networks.

The contributions of this work can be therefore summarized as:

- An in-depth analysis of six state-of-art ABS algorithms streaming with varying bandwidth in static and mobile 5G scenarios. The analysis is undertaken through the assessment of associated QoE models such as the P.1203 QoE standard.
- A proposal for a Machine Learning classifier to estimate QoE based on RTT, number of packets and throughput.

The rest of the paper is structured as follows: Section 2 presents background and related work. Section 3 describes the experimental setup followed by Section 4, which gives a brief introduction into ML classifiers and pre-processing in Sec 4.1. Section 5 shows how QoS impacts on QoE, while Section 6 discusses our results. Section 7 concludes our paper and considers future work.

2 BACKGROUND AND RELATED WORK

In HTTP Adaptive Streaming (HAS), a video file is divided into different small chunks no longer than a few seconds each, where each chunk is encoded with different quality levels (representations) and respective transmission sizes. This permits the client to select the appropriate chunk at a transmission size that suits the throughput of the network the client is connected to. Thus permitting an adaptive mechanism through which QoE can increase when network conditions improve. The structure of each video stream is described in a Media Presentation Description (MPD) file. When a video client wants to play a specific video stream, the client first downloads the MPD file, then the client's video adaptation algorithm plays a key role and is responsible for requesting the most appropriate representation for each segment based on the clients play-out estimation. The adaptation to network conditions is one of the reasons that Adaptive HTTP Streaming (DASH) clients experience better resolution and quality [35]. Incorrect choice in representation bitrate can cause the end-user to experience a series of stall events. Stalling describes video playback interruption due to buffer under-run. When the buffer level is beneath a given threshold, insufficient data for playback is available, and the video playback has to be stopped until the buffer is refilled.

Quality of experience (QoE) assessment can be divided into two main categories: i) Subjective, and ii) Objective. Subjective QoE assessment utilizes end-users who grade video quality at the end of a video session using perceiving video quality so-called Mean Opinion Score (MOS). The ITU-T recommendations [30] for subjective quality evaluation follow strict setup and testing conditions. However, subjective QoE is expensive, time-consuming, and doesn't scale very well. Moreover, there are many other factors such as psychological or psycho-physiological, e.g., age, mood, time of day, gender, and socio-economic status [15] that may influence the results. For this reason, objective QoE assessment has gained more popularity [13], with some models that directly map objective QoE to well-known metrics such as MOS, Peak Signal to Noise Ratio (PSNR), and Structural Similarity Index Metrics (SSIM). Variance in the results of these metrics can be tied directly to the quality of the original video stream, and as such more ground truth is needed to improve objective QoE values.

The first study that investigates video quality, freezing stall events, and stall's duration is proposed in [39]; and was adopted by the ITU [25]. [18] reviews more recent methodologies to predict QoE. Similarly, [6] has recently proposed a means to cope with complex QoS and QoE mapping. [34] investigates stall, initial delay, and visual quality for end-user QoE. All of these aforementioned contributions agreed on three factors that influence video quality during transmission: 1) Stall, 2) Representation Stream Bitrate, and 3) Visual Quality [20]. However, how different adaptation algorithms influence the end-user's QoE by varying network QoS also needs to be taken into account. Therefore, these factors are discussed in Sec 5 in more detail as well as their impact on the standardized QoE model P.1203.

Finally, Dimopoulos et al. [11] leverage ML to evaluate the correlation between QoS and QoE to overcome the challenge of measuring end-user satisfaction. Another area where ML techniques were recently shown a lot of promise is improving and designing ML-backed ABR schemes. Authors in [2, 10] use ML to compute parameters of the existing ABR scheme to adapt to dynamically changing network conditions. Unlike these approaches, several approaches train ML model as the replacement for ABR algorithm [7, 19, 33]. In comparison to the literature, our work considers a linear regression ML model that considers RTTs, throughput, and number of packets as input to classifiers to map QoS to QoE. Also, it is important to note that our ML model is unaware of the underlying algorithm type (and its characteristics) used for streaming the HAS video and mandates reasonable realistic results.

3 EXPERIMENTAL SETUP

In this section, we present the experimental setup for the experiments performed using the network emulator *mininet*[1] and *godash* [22] the headless DASH video player (go accelerated framework). Figure 1 depicts the topology used in the experiments. We assess the impact of concurrent video streaming clients, keeping client 1 (1st client) as a reference, and varying the number of concurrent clients between (1,...,2), (1,...,3), (1,...,5); with all of the clients streaming from the same server. Video quality metrics are assessed in real-time from the *godash* player. The network bandwidth sample values are based on the 5G trace parameters [28], where we select 10 different combinations[2] (in Mbps): Mobility (driving) — (38.26 to 10.33), (29.33 to 10.55), (0.5 to 3) and (6 to 14), and static — (72.42 to

[1] http://mininet.org

[2] Value combinations taken from: https://github.com/uccmisl/5Gdataset

9), (70 to 20), (52.06 to 0.5), (4.19 to 8), (0.5 to 6) and (8.29 to 57.15). These bandwidth combination values consist of different variations of (static and mobility) network throughput extracted from the 5G network traces such that the video clients stream from very high bandwidth to low and moderate and vice versa.

In our experiments, we use a well-known video sample named *Sintel*, from the publicly available video dataset for Advanced Video Coding (AVC) H.264 [26]. Sintel is encoded in thirteen different representation rates across eight resolutions[3]. The total stream duration of Sintel is over 14 minutes. Furthermore, Sintel is divided into chunks/segments of two seconds (2s), from which we stream 60 segments in each experiment, i.e., 60×2s = 120s. The bandwidth during each experiment is changed after every 4 seconds, i.e., with 4s sampling interval two video segments can be downloaded before a new bandwidth value is sampled from the 5G trace files. While not shown in this submission, please note that we also evaluated other sampling intervals such as (1s, 2s), without noticing a significant impact in our evaluation results. The experiments are run on a Intel Core i7, SSD Linux machine. The bandwidth is changed at the bottleneck between S_1 and S_2, at run-time using Linux Traffic control (TC) and Hierarchical Token Bucket (HTB) [4]. The bottleneck router buffer is set to be one Bandwidth Delay Product (BDP). Also, a parameterized script is used to run each experiment, resulting in one *godash* logfile per client and a corresponding *pcap* file per run. Table 1 shows parts of the content of a *godash* log file, composed of the following information: Seg_# as (segment number), Algorithm (Elastic, Arbiter+, etc.), Seg_Dur (video segment duration, i.e, 2s), Codec (h264), Width - Height (width and height of the corresponding video segment), FPS (Frame Rate per Second), Play_Pos (current play back position (ms)), RTT (Packet level (ms) - determined using HTTP's head request), and the output of the five QoE models (Yin, Duanmu, Yu, Clae, P.1203). Details from each feature is available in *godash* [29]. For each experiment, we collect per-segment QoS information, e.g., RTT, throughput and number of packets per video segment, from the *pcap* files with a Python script[4]. For each of the clients' 60 video segments (2 minutes of video in total), the network QoS information, and each corresponding *godash* logfile, a PHP script is used to store the experiments' data in a database. Example output is shown in Table 2, and is described as per the following: (Host, Stall, Bitrate, Segment) are indicated as (host number, stall and bitrate (during the video segment), and segment number) followed by (Total_Users, Buffer, and Algorithm) as (total user during the experiment, buffer level on the corresponding segment and adaptation algorithm). The QoS collected from the *pcap* files of each segment is indicated as (RTT, throughput, number of packets) for all five QoE models. We open-source all scripts used for collection and pre-processing of our generated dataset[5].

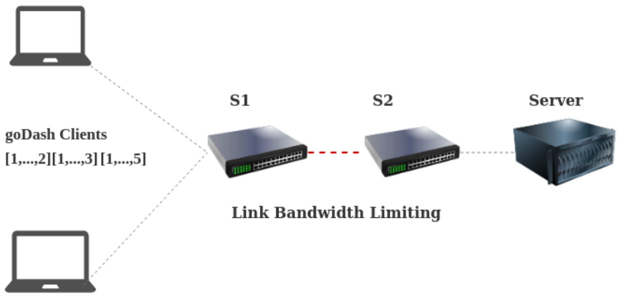


Figure 1: Experiment topology

4 MACHINE LEARNING: METHODOLOGY

In this section we introduce the various Machine Learning models we use in our work. Typically, ML problems can be classified as the following: i) Supervised, and ii) Un-supervised. Supervised can be further divided into two types. i) regression, and ii) classification. During a typical classification problem, the ML models try to predict certain categorical classes, but, in regression, the model outputs real value such as integers or floating-point numbers based on input variables. For a comprehensive analysis of the dataset, that requires less computational overhead during the pre-processing phases, such as scaling and normalization, we selected Decision Tree Regression (DTR), Multi-linear Regression (MLR) and Random Forest Regression (RFR).

DTR utilises a tree-like structure in its models, and is popular in both regression and classification problems. Departing from the root (parent) node, child nodes are decided by the largest Information Gain (IG) [5], and the iterative process terminates when the leaves are so-called *pure*. MLR is a statistical technique used to predict correlation between variables from independent predictors. MLR, or simply Multiple Regression (MR), is used to explain the relationship between one continuous dependent variable and two or more independent variables [3]. The relationship between variables can tell the change in the target value, by fitting a line through the observations. In our experiments, we have three main input variables (network QoS parameters) to the MLR classifier: RTT, number of packets for each video segment, and throughput.

Finally, we also apply RFR, which is used for classification and regression by building multiple DTs. RFR is also commonly known as Bagging, because it trains each DT on different data samples and, finally, instead of relying on a single tree, it merges all of them together before taking the final decision [17]. In our experiments using different regression classifiers we find the strength of independent variable on dependent variable Y (P.1203) [27, 31], in other words, how RTT, number of packets per video segment and throughput have impact P.1203's score.

$$\mathrm{P.1203}_t = \alpha + \beta_1 \mathrm{RTT}_t + \beta_2 \mathrm{Packets}_t + \beta_3 \mathrm{Throughput}_t + \epsilon \qquad (1)$$

In Equation (1), β_1 , β_2 and β_3 are regression coefficients associated with RTT, number of packets per video segment, and throughput, respectively. *epsilon* is the random error component reflecting the difference between the observed and fitted linear relationship.

4.1 Pre-processing

By inspecting the pattern changes in RTT and throughput, we split 60 video segments from the log file into 4 equal parts, i.e, 30s each. Note that the length of the video file in each experiment run is 2

[3] http://cs1dev.ucc.ie/misl/4K_non_copyright_dataset/2_sec/x264/sintel/DASH_Files/full/sintel_enc_x264_dash.mpd

[4] https://github.com/razaulmustafa852/5G/blob/master/updated-per-segment-others.ipynb

[5] https://github.com/razaulmustafa852/5G

Table 1: *godash* log file, 1st client: First 5 video segments of 2s video segment duration with BBA algorithm

Seg_#	Algorithm	Seg_Dur	Codec	Width	Height	FPS	Play_Pos	RTT	P.1203	Claey	Duanmu	Yin	Yu
1	BBA	2000	H264	320	180	24	0	50.649	1.878	0.000	1726.005	239515.000	233515.000
2	BBA	2000	H264	384	216	24	4000	88.149	1.890	0.419	2305.465	309046.000	467030.000
3	BBA	2000	H264	384	216	24	6000	60.046	1.903	0.477	2467.704	332223.000	845607.000
4	BBA	2000	H264	384	216	24	8000	33.467	1.766	0.509	2560.764	343791.373	5028184.000
5	BBA	2000	H264	384	216	24	10000	20.293	1.649	0.529	2616.268	350744.524	9576761.000

Table 2: Features stored in the database for offline investigation

Host	Stall	Bitrate	Segment	Total_Users	Buffer	Algorithm	RTT	Throughput	Packets	P.1203	Claey	Duanmu	Yin	Yu
1	0	2441	60	2	30704	exponential	11.24	899614.28	194	4.88	5.11	77.34	252609.84	4.32
1	0	5314	59	2	30479	exponential	8.17	2034351.78	313	4.88	5.10	77.34	248234.83	4.31
1	0	218	37	2	2543	arbiter	42.67	197013.62	20	1.24	0.41	46.88	9641.51	0.28
1	240	173	34	2	2000	arbiter	58.19	84982.20	16	1.24	0.41	47.55	11335.03	0.29

minutes. Together with this processed data, we take all comprehensive information available in *godash* log files, i.e., aggregated RTT, throughput, number of packets per video segment, and P.1203 score as shown in Table 3. The three columns in the middle of Table 3 are used as input for the for ML classifiers to predict P.1203 scores.

Table 3: Processed dataset used in the ML classifiers

Column	User	Algorithm	RTT	Throughput	Packets	P1203
2	1	Arbiter	3.76	2584825.53	126.46	3.12
5	2	Elastic	0.23	7682269.18	65	3.02
2	2	BBA	0.58	2866549.71	64	2.94
5	1	Logistic	0.16	7212008.60	17	1.87
4	1	Conventional	0.66	6377796.25	87.73	3.56
4	2	Exponential	8.65	1077560.73	291.86	4.84

The first three columns (Column, User, Algorithm) are used to differentiate each trace fed into the ML model separately. For instance, (Column=1) means the first scenario from the 5G trace parameters described in Section 3. To train a single model for static and mobility scenarios, we use *pandas.get_dummies* to convert categorical algorithm names into dummy, or indicator, variables. The proposed ML methods, i.e, DTR, MLR and RFR using Python's *scikit-learn* library were trained on 80 % of data, while the remaining 20 % used for testing trained ML models.

5 QOS IMPACT ON QOE

In this section, we assess the impact of network QoS on client QoE. We illustrate the relationship between QoS and QoE for each of following ABS algorithms: Rate-based — Conventional [16] and Exponential, Buffer-based — Logistic [32] and BBA [14], and Hybrid — Arbiter+ [38] and Elastic [8]. The adaptation mechanism of each of the selected algorithms is explained in Table 4.

Table 4: ABS Algorithm Adaptation Mechanisms

Algorithm	Type	Mechanism
Arbiter [38] +	Hybrid	The quality selection policy includes targeting a reduction of the stall risk by performing smooth representation switches.
Elastic [8]	Hybrid	Is designed to ensure application-level fairness in the case of sharing a bottleneck.
BBA [14]	Buffer-based	Represents the class of algorithms that solely depends on buffer-level in the adaptation decision.
Logistic [32]	Buffer-based	This model is able to find the optimal buffer required for any given set of video quality levels.
Conventional [16]	Rate-based	The TCP download throughput observed by a client is directly taken as its fair share of the network bandwidth.)
Exponential	Rate-based	Exponential growth of past throughput.

We evaluate all ABS algorithms in Table 4 in two test scenarios: a) Good, and b) Moderate. In both cases, three key metrics are analyzed (RTT, number of stalls, and the P.1203 standardized scores). For scenario a) the bandwidth range is (6 − 14) Mbps and scenario b) the range is (0.5 − 3) Mbps[6] After each experiment, *godash* generates QoE metrics, from which we select the P.1203, mode 0, standardized scores (which we denote as P.1203). P.1203 generates a value in a scale ranging from 0 to 5 (where a higher score values denotes a better QoE value for the client). For example, 0 means the lowest and worst P.1203 score, which, in other words describes a scenario when a video is streamed under poor network conditions. Next let us consider two concurrent streaming clients, based on the topology as shown in Figure 1. In each of the following figures, we group each of the algorithms per HAS category, and illustrate a range of their outputs in both test scenarios (across two figures - the first figure illustrating adaptation in segment bitrate and the second figure showing changes in RTT, Stalls and P.1203 values). This gives a clear overview of how these algorithms adapt in different scenarios and also which of the algorithms performs better in each scenario (typically, the higher the segment bitrate, the better the overall QoE - two items to note: 1) stalls are not shown here, and stalls dramatically decrease user QoE and 2) constant switching between bitrates (assuming changes in different resolutions) also reduces overall user enjoyment.

Figure 2 shows that the rate-based Exponential and Conventional algorithms perform similarly over the experiment time in both Good (a) and Moderate (b) scenarios. In scenario a) Good: up to 50s, both algorithms deliver on average 1 Mbps. Around (20s − 30s) and (50s − 60s), higher segment bitrate can be translated directly into higher video resolution. Figure 5 shows similar trends in RTT, number of stalls, and P.1203 score, with an overall slight advantage for Conventional in scenario a). In the Moderate scenario, Exponential achieves a small advantage in overall segment bitrate, i.e., achieving higher visual video quality. It is important to note that

[6] Due to space constraints, we select the mobility scenario from the whole range of our experiments, which contain the most interesting results in our opinion, and not all possible combinations for static and mobility scenarios from Section 3.

in Figure 5, both algorithms constantly make the wrong choice of representation bitrate (achieving an overall higher representation level of video quality), which can be seen in the levels of stalls in both test scenarios.

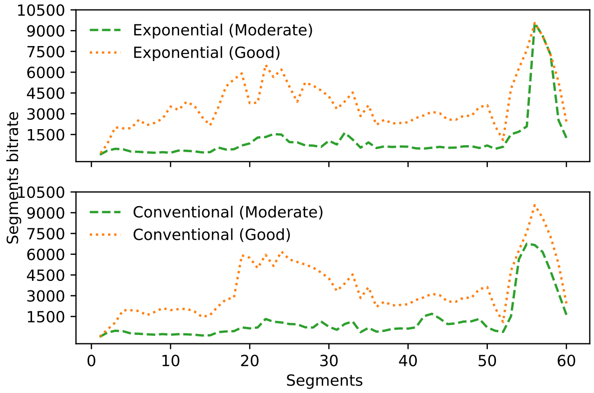


Figure 2: Rate-based

Figure 3 presents the output of the buffer-based BBA and Logistic algorithms. These algorithms perform very similarly over the experiment time, with a small advantage in terms of segment bitrate for BBA in scenario a). Figure 6 shows that although Logistic does not report any stall in scenario a), video segments have considerably higher RTT and overall lower P.1203 score. However, scenario b) reports that both algorithms are comparably similar, with BBA reporting overall lower P.1203 scores thus translated into lower video resolution. It can also be seen in Figure 6, that in moderate bandwidth scenarios, fluctuations in the bandwidth can cause the buffer-based algorithms to make the wrong choice. Highlighting the need for a predictive model for adaptive algorithms in cellular networks.

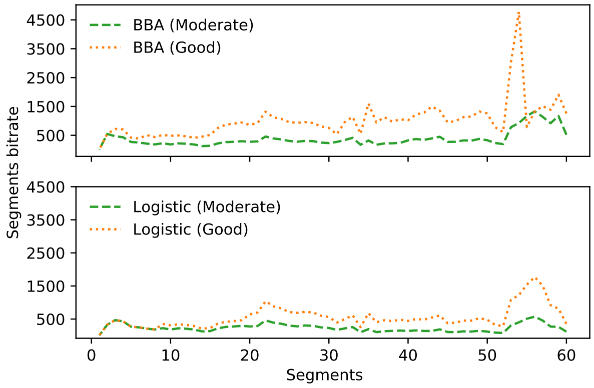


Figure 3: Buffer-based

Finally, Figure 4 shows that the hybrid algorithm, Arbiter+ consistently outperforms Elastic in scenarios a) and b) with higher segment bitrate. However, looking at Figure 7 higher segment bitrate come at the cost of more stalls that also result in lower P.1203 scores translated into lower video quality. In other words, one can conclude in both scenarios, a) and b), that Arbiter+ tries to maintain higher segment bitrate, even though it experiences more stall events and lower P.1203 scores as direct consequence.

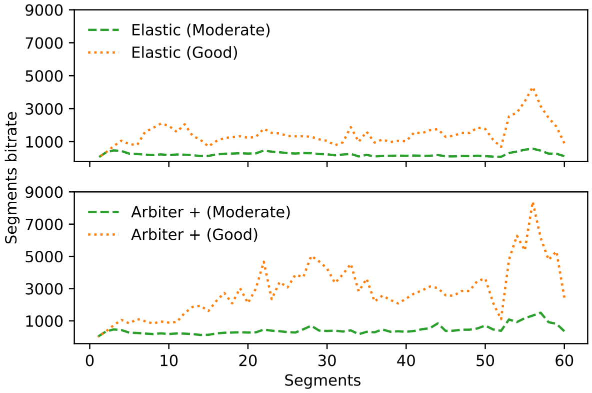


Figure 4: Hybrid

These controlled experiments and single runs of each ABS algorithm in Table 4 give us a bird view of the impact of bandwidth variation, described as scenarios a) Good and b) Moderate. We note that for each of the ABS algorithms, they follow quite different strategies to adjust the video bitrate. Following the methodology described in Section 4, we would like to use next network QoS information (RTT, number of packets for each video segment, throughput) and apply different ML algorithms to estimate the resulting P.1203 score. The result of the predicted P.1203 scores will be then compared against our ground-truth, namely, the values reported by *godash*.

6 VIDEO QOE PREDICTION WITH ML

In Section 4, we describe the methodology to apply ML to predict the P.1203 score as a function of varying network QoS. In our analysis, we use a total of 13,547 observations (approximately 225 client runs - across 2-client, 3-client and 5-client experiments) as input and evaluate three regression classifiers, namely, DTR, MLR, and RFR. The input dataset used in our experiments describe a static and a mobility (driving) scenario as already discussed in Section 3.

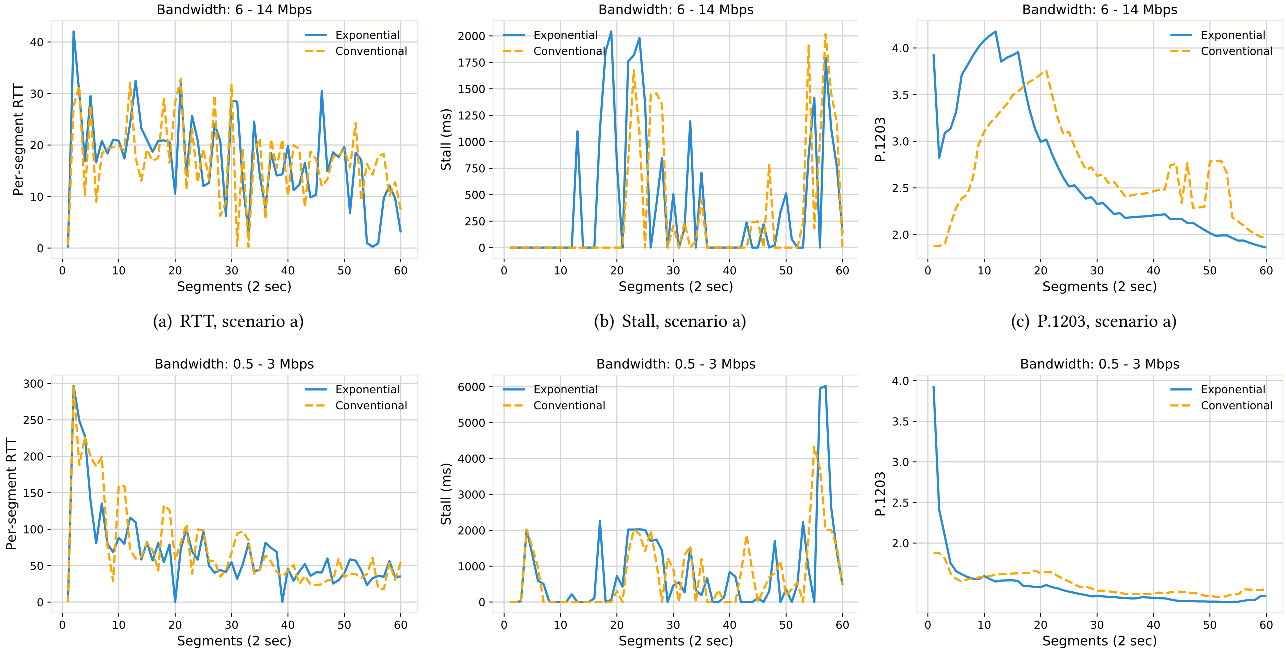
Initially, each classifier is separately trained with each of the ABS algorithm categories, namely: Rate-based, buffer-based and hybrid. In other words, each classifier is trained with data from both Arbiter+ and Elastic for the Hybrid category, and the same is done for rate- and buffer-based ABS algorithms. Table 5 shows for the static scenario that RFR achieves much higher accuracy compared to DTR and MLR.

To quantify the predicted error of the P.1203 values to the ground-truth P.1203 scores, we use the Mean Absolute Error (MAE):

$$MAE = (\frac{1}{n})\sum_{i=1}^{n}|y_i - \overline{x}_i| \qquad (2)$$

MAE is a metric used to find the similarity between two sets. The absolute error is the absolute difference, whereas the error is the difference of two numbers. In order to find MAE, we first need to find the absolute error between two values, and then find the mean of these values. In Equation (2), y and $\overline{x}$ correspond to the actual and predicted value, respectively.

Similarly, Table 6 shows the classifiers' results for the mobility (driving) scenario, where RFR has a considerably lower MAE and much better accuracy see Table 7. In the static scenario, the same regression classifier RFR has an accuracy of 87.63 % whereas



(a) RTT, scenario a)

(b) Stall, scenario a)

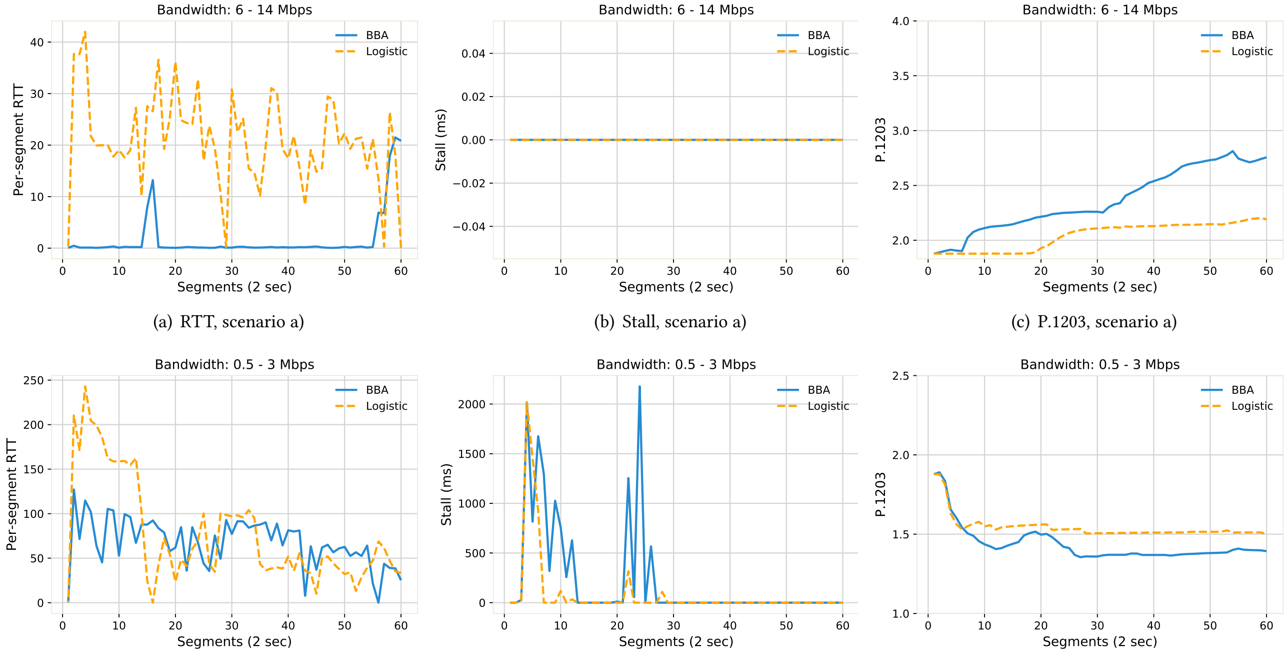
(c) P.1203, scenario a)

(d) RTT, scenario b)

(e) Stall, scenario b)

(f) P.1203, scenario b)

Figure 5: Rate-based, scenarios a) and b) (1st video client): RTT, stalls, and P.1203 score per video segment for 60 video segments



(a) RTT, scenario a)

(b) Stall, scenario a)

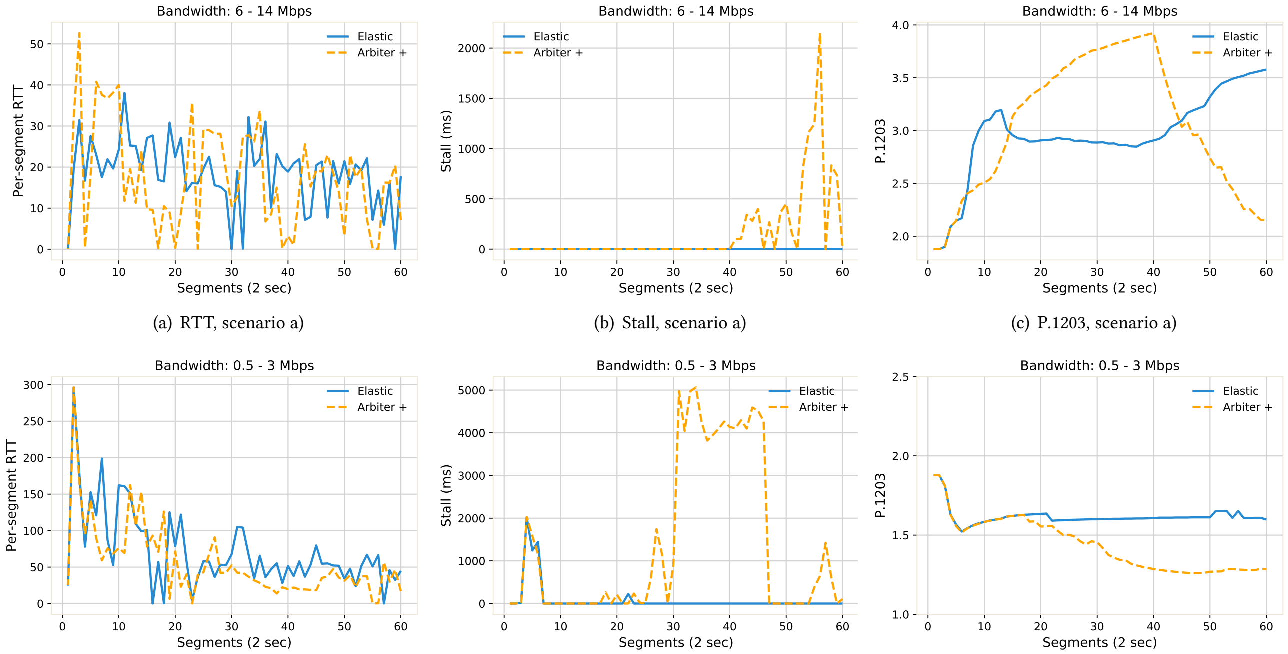
(c) P.1203, scenario a)

(d) RTT, scenario b)

(e) Stall, scenario b)

(f) P.1203, scenario b)

Figure 6: Buffer-based, scenarios a) and b) (1st video client): RTT, stalls, and P.1203 score per video segment for 60 segments



(a) RTT, scenario a)

(b) Stall, scenario a)

(c) P.1203, scenario a)

(d) RTT, scenario b)

(e) Stall, scenario b)

(f) P.1203, scenario b)

Figure 7: Hybrid, scenarios a) and b) (1st video client): RTT, stalls, and P.1203 score per video segment for 60 segments

Table 5: Static scenario: Rate-based, Buffer-based and Hybrid: Classifiers' accuracy with MAE

Algorithm	Classifiers	MAE [%]
Arbiter +, Elastic	DTR	0.20
	RFR	0.17
	MLR	0.55
BBA, Logistic	DTR	0.12
	RFR	0.07
	MLR	0.12
Conventional, Exponential	DTR	0.23
	RFR	0.10
	MLR	1.03

Table 6: Mobility scenario: Rate-based, Buffer-based and Hybrid: Classifiers' accuracy with MAE

Algorithm	Classifiers	MAE [%]
Arbiter +, Elastic	DTR	0.31
	RFR	0.31
	MLR	0.55
BBA, Logistic	DTR	0.01
	RFR	0.01
	MLR	0.19
Conventional, Exponential	DTR	0.13
	RFR	0.07
	MLR	0.70

Table 7: Classifiers' accuracy in static and mobility scenarios

Case	Classifiers	Accuracy
Static	DTR	78.68 %
	RFR	87.63 %
	MLR	40.01 %
Driving	DTR	72.37 %
	RFR	79.00 %
	MLR	58.67 %

DTR has 78.68 % and 72.37 % in the static and mobility scenarios, respectively.

7 CONCLUSION

In this paper, we leverage Machine Learning (ML) to train an algorithm agnostic classifier that predicts user quality-of-experience of video session based on network quality-of-service metrics, RTT, a number of packets and measured throughput.

To generate a dataset upon which we train our ML models, we investigated the achievable video quality of different state-of-art Adaptive Bitrate Streaming (ABS) algorithms commonly found in HTTP Adaptive Streaming (*HAS*), by varying the network quality-of-service (bandwidth) sampled from real 5G traces. From this investigation, we understand the impact of bandwidth variation on each category of ABS algorithm (Rate-based, Buffer-based and Hybrid) and how they follow different strategies to adapt the video bitrate and achievable quality-of-experience. After collecting the resulting P.1203 scores in these experiments, we applied three different machine learning classifiers on the same data to verify the predicted P.1203 score compared to our ground-truth, namely, the P.1203 reported by *godash* log files. We find out that Random Forest Regression (RFR) outperforms both Decision Tree (DTR) and Multilinear Regression (MLR), achieving higher accuracy of 87.63% and 79.00% in static and mobility scenarios, respectively.

For future work, we would expect to extend our study in different directions: First, we would like to analyse the impact of both bandwidth and throughput, i.e., vary both values simultaneously, in the video quality of experience of the end-user. Also, we would like to improve our analysis by selecting more video segments per sampled value as well as more concurrent video clients. Finally, we would like to investigate online real-time quality-of-experience prediction with at least one of our selected classifiers in this study, i.e., RFR, in both static and low/high mobility scenarios.

REFERENCES

[1] Adnan Ahmed, Zubair Shafiq, Harkeerat Bedi, and Amir Khakpour. 2017. Suffering from buffering? detecting qoe impairments in live video streams. In *2017 IEEE 25th International Conference on Network Protocols (ICNP)*. IEEE, 1–10.

[2] Zahaib Akhtar, Yun Seong Nam, Ramesh Govindan, Sanjay Rao, Jessica Chen, Ethan Katz-Bassett, Bruno Ribeiro, Jibin Zhan, and Hui Zhang. 2018. Oboe: Auto-Tuning Video ABR Algorithms to Network Conditions. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication* (Budapest, Hungary) *(SIGCOMM '18)*. Association for Computing Machinery, New York, NY, USA, 44–58. https://doi.org/10.1145/3230543.3230558

[3] Sana Aroussi, Thouraya Bouabana-Tebibel, and Abdelhamid Mellouk. 2012. Empirical QoE/QoS correlation model based on multiple parameters for VoD flows. In *2012 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 1963–1968.

[4] Doru Gabriel Balan and Dan Alin Potorac. 2009. Linux HTB queuing discipline implementations. In *2009 First International Conference on Networked Digital Technologies*. IEEE, 122–126.

[5] Neeraj Bhargava, Girja Sharma, Ritu Bhargava, and Manish Mathuria. 2013. Decision tree analysis on j48 algorithm for data mining. *Proceedings of International Journal of Advanced Research in Computer Science and Software Engineering* 3, 6 (2013).

[6] Yanjiao Chen, Kaishun Wu, and Qian Zhang. 2014. From QoS to QoE: A tutorial on video quality assessment. *IEEE Communications Surveys & Tutorials* 17, 2 (2014), 1126–1165.

[7] Y. Chien, K. C. Lin, and M. Chen. 2015. Machine learning based rate adaptation with elastic feature selection for HTTP-based streaming. In *2015 IEEE International Conference on Multimedia and Expo (ICME)*. 1–6.

[8] L. De Cicco et al. [n.d.]. ELASTIC: A Client-Side Controller for Dynamic Adaptive Streaming over HTTP (DASH). In *2013 20th International Packet Video Workshop*. https://doi.org/10.1109/PV.2013.6691442

[9] Conviva. 2020. *Conviva's State of Streaming Q1 2020*. https://www.conviva.com/state-of-streaming/

[10] Luca De Cicco, Giuseppe Cilli, and Saverio Mascolo. 2019. ERUDITE: A Deep Neural Network for Optimal Tuning of Adaptive Video Streaming Controllers. In *Proceedings of the 10th ACM Multimedia Systems Conference* (Amherst, Massachusetts) *(MMSys '19)*. Association for Computing Machinery, New York, NY, USA, 13–24. https://doi.org/10.1145/3304109.3306216

[11] Giorgos Dimopoulos, Ilias Leontiadis, Pere Barlet-Ros, and Konstantina Papagiannaki. 2016. Measuring video QoE from encrypted traffic. In *Proceedings of the 2016 Internet Measurement Conference*. ACM, 513–526.

[12] Z. Duanmu et al. 2018. A Quality-of-Experience Database for Adaptive Video Streaming. *IEEE Transactions on Broadcasting* 64, 2 (June 2018), 474–487. https://doi.org/10.1109/TBC.2018.2822870

[13] Christian Esteve Rothenberg, Danny Alex Lachos Perez, Nathan F. Saraiva de Sousa, Raphael Rosa, Raza Ul Mustafa, Md Tariqul Islam, and Pedro Henrique Gomes. 2020. Intent-based Control Loop for DASH Video Service Assurance using ML-based Edge QoE Estimation. In *2020 6th IEEE International Conference on Network Softwarization (NetSoft) (NetSoft 2020)*. Ghent, Belgium.

[14] T. Huang et al. 2014. A Buffer-based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service. In *Proceedings of the 2014 ACM Conference on SIGCOMM* (Chicago, Illinois, USA) *(SIGCOMM '14)*. ACM, New York, NY, USA, 187–198. https://doi.org/10.1145/2619239.2626296

[15] Péter András Kara, László Bokor, Andreas Sackl, and Mariana Mourão. 2015. What your phone makes you see: Investigation of the effect of end-user devices on the assessment of perceived multimedia quality. In *2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX)*.

[16] Z. Li et al. 2014. Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale. *IEEE Journal on Selected Areas in Communications* 32, 4 (April 2014), 719–733. https://doi.org/10.1109/JSAC.2014.140405

[17] Andy Liaw, Matthew Wiener, et al. 2002. Classification and regression by randomForest. *R news* 2, 3 (2002), 18–22.

[18] Orlewilson Bentes Maia, Hani Camille Yehia, and Luciano de Errico. 2015. A concise review of the quality of experience assessment for video streaming. *Computer Communications* 57 (2015), 1–12.

[19] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. 2017. Neural Adaptive Video Streaming with Pensieve. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication* (Los Angeles, CA, USA) *(SIGCOMM '17)*. Association for Computing Machinery, New York, NY, USA, 197–210. https://doi.org/10.1145/3098822.3098843

[20] Maria G Martini, CTER Hewage, Moustafa M Nasrall, and Ognen Ognenoski. 2016. QoE control, monitoring, and management strategies. *Multimedia Quality of Experience (QoE): Current status and future requirements* (2016), 149–167.

[21] James Nightingale, Pablo Salva-Garcia, Jose M Alcaraz Calero, and Qi Wang. 2018. 5G-QoE: QoE modelling for ultra-HD video streaming in 5G networks. *IEEE Transactions on Broadcasting* 64, 2 (2018), 621–634.

[22] John O'Sullivan, Darijo Raca, and Jason J Quinlan. 2020. godash 2.0-The Next Evolution of HAS Evaluation. (2020).

[23] Nisha Panwar, Shantanu Sharma, and Awadhesh Kumar Singh. 2016. A survey on 5G: The next generation of mobile communication. *Physical Communication* 18 (2016), 64–84.

[24] S. Petrangeli et al. 2015. QoE-Driven Rate Adaptation Heuristic for Fair Adaptive Video Streaming. *ACM Trans. Multimedia Comput. Commun. Appl.* (Oct. 2015).

[25] Margaret H Pinson, Nicolas Staelens, and Arthur Webster. 2013. The history of video quality model validation. In *2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 458–463.

[26] Jason J Quinlan, Ahmed H Zahran, and Cormac J Sreenan. 2016. Datasets for AVC (H. 264) and HEVC (H. 265) evaluation of dynamic adaptive streaming over HTTP (DASH). In *Proceedings of the 7th International Conference on Multimedia Systems*. 1–6.

[27] Alexander Raake et al. 2017. A bitstream-based, scalable video-quality model for HTTP adaptive streaming: ITU-T P.1203.1. In *Ninth International Conference on Quality of Multimedia Experience (QoMEX)*. https://doi.org/10.1109/QoMEX.2017.7965631

[28] Darijo Raca, Dylan Leahy, Cormac J Sreenan, and Jason J Quinlan. 2020. Beyond throughput, the next generation: a 5G dataset with channel and context metrics. In *Proceedings of the 11th ACM Multimedia Systems Conference*. 303–308.

[29] Darijo Raca, Maëlle Manifacier, and Jason J Quinlan. 2020. goDASH-GO accelerated HAS framework for rapid prototyping. (2020).

[30] Ulrich Reiter, Kjell Brunnström, Katrien De Moor, Mohamed-Chaker Larabi, Manuela Pereira, Antonio Pinheiro, Junyong You, and Andrej Zgank. 2014. Factors influencing quality of experience. In *Quality of experience*. Springer, 55–72.

[31] Werner Robitza et al. 2018. HTTP Adaptive Streaming QoE Estimation with ITU-T Rec. P. 1203: Open Databases and Software. In *9th ACM Multimedia Systems Conference* (Amsterdam, Netherlands) *(MMSys '18)*. 466–471. https://doi.org/10.1145/3204949.3208124

[32] Y. Sani et al. 2015. Modelling Video Rate Evolution in Adaptive Bitrate Selection. In *2015 IEEE International Symposium on Multimedia (ISM)*. 89–94. https://doi.org/10.1109/ISM.2015.65

[33] Y. Sani, D. Raca, J. J. Quinlan, and C. J. Sreenan. 2020. SMASH: A Supervised Machine Learning Approach to Adaptive Video Streaming over HTTP. In *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. 1–6.

[34] Michael Seufert, Nikolas Wehner, and Pedro Casas. 2018. Studying the impact of HAS qoe factors on the standardized qoe model P. 1203. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 1636–1641.

[35] Thomas Stockhammer. 2011. Dynamic adaptive streaming over HTTP–: standards and design principles. In *Proceedings of the second annual ACM conference on Multimedia systems*. ACM, 133–144.

[36] Xiaoqi Yin et al. 2015. A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP. In *2015 ACM Conference on Special Interest Group on Data Communication* (London, United Kingdom) *(SIGCOMM '15)*. 14.

[37] L. Yu et al. 2017. QoE-Driven Dynamic Adaptive Video Streaming Strategy With Future Information. *IEEE Transactions on Broadcasting* 63, 3 (Sep. 2017), 523–534. https://doi.org/10.1109/TBC.2017.2687698

[38] A. H. Zahran et al. [n.d.]. ARBITER+: Adaptive Rate-Based InTElligent HTTP StReaming Algorithm for Mobile Networks. *IEEE Transactions on Mobile Computing* ([n. d.]).

[39] Fan Zhang, Long Xu, and Qian Zhang. 2013. Maximum-likelihood visual quality based on additive log-logistic model. In *2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 470–475.