

PREDICTING QUALITY OF EXPERIENCE FOR ONLINE VIDEO SERVICE PROVISIONING

Utku Bulkan, Tasos Dagiuklas
SuITE Research Group

Division of Computer Science, London South Bank University, UK
Email: {bulkanu, tdagiuklas}@lsbu.ac.uk

Abstract—The expansion of the online video content continues in every area of the modern connected world and the need for measuring and predicting the Quality of Experience (QoE) for online video systems has never been this important. This paper has designed and developed a machine learning based methodology to derive QoE for online video systems. For this purpose, a platform has been developed where video content is unicast to users so that objective video metrics are collected into a database. At the end of each video session, users are queried with a subjective survey about their experience. Both quantitative statistics and qualitative user survey information are used as training data to a variety of machine learning techniques including Artificial Neural Network (ANN), K-nearest Neighbours Algorithm (KNN) and Support Vector Machine (SVM) with a collection of cross-validation strategies. This methodology can efficiently answer the problem of predicting user experience for any online video service provider, while overcoming the problematic interpretation of subjective consumer experience in terms of quantitative system capacity metrics.

Index Terms—Quality of Experience (QoE), Machine Learning, Online Video Services, Content Delivery, QoE Modelling, Subjective QoE Assessment, H.264, HTTP Streaming, MPEG-DASH, VOD.

I. INTRODUCTION

OVER the last decade, video has become the main component of the Web. In today’s world, social media, news channels, conventional television broadcasting and also most of telephony products are all built upon video services [1, 2, 4]. Analysis has shown that [1] whenever the content provider company fails to deliver the content in expected time and quality, the user might tend to cancel their subscription regardless if it is a paid or a free service. According to a recent whitepaper from Akamai [2] “With a 5-second delay in starting a video playing, a publisher may have a probability to lose a quarter of its audience, and with a 10-second delay, nearly half of the audience are eager to leave.”

In an ideal world, where each user sends information about their experience, it would be easy to translate this instant feedback from user’s feelings into system and network parameters to increase customer satisfaction. However, only a very small percentage of the consumers provide instant feedback about the service experience. Yet, this information can be translated into valuable feedback as many of the frontrunner companies like Facebook, Whatsapp and Skype frequently employ these methodologies.

The aim of this paper is to answer the question of measuring user experience and correlating them to objective video

consumption and system capacity parameters. Unlike other research works [3, 6, 8] that use time invariant models, real time Quality of Experience (QoE) of an online video is predicted using the correspondence between quantitative and qualitative observations using machine-learning methods. The capability to measure single user’s QoE concludes on measuring the whole service quality. The ability to compare service QoE delta between two different moments is the key to provision the resources that construct an online video delivery system.

The remainder of the paper is structured as follows: Section II gives a brief description about components of an online video platform, Section III outlines state of art QoE questionnaire and Section IV provides information on related works on the QoE. Section V advertises contributions of the project and Section VI specifies details of the system implementation for online video platform. Section VII clarifies experiment methodology, postulates derivation of overall system QoE from single user QoE and discusses comparison of the models. Section VIII, IX & X discusses supervised machine learning, model performance comparison and online video platform capacity estimation. Finally, conclusions and future works are argued in Section XI.

II. ON-LINE VIDEO PLATFORM AND QOE

As shown in the Figure 1, the generalized view of an online video platform consists of following components; consumer end-user device (mobile, tablet or PC), browsers or other player software, network layer, content delivery network (CDN), load balancer, web services, and video platform.

In such a complex procedure like online video delivery, there can be several bottlenecks that might cause deterioration in delivering the content to consumer. This includes player and consumer device related errors [11], network congestion [5], video encoding and adaptation related quality drops [6, 7], CDN [10] problems or user and social context factors [8, 9].

For any of these scenarios, when users are not satisfied with the service, instant feedback to actual product owner plays a crucial role as the subscribers can submit their experience about the service instantaneously as a subjective QoE survey, which can influence correct changes for the system operator. This will unquestionably save time, profit and practically the entire business. Moreover, following a low consumer experience, only a very small number of users are eager to share their feelings. For instance; users who face a long initial buffering duration while trying to watch a YouTube video [5,

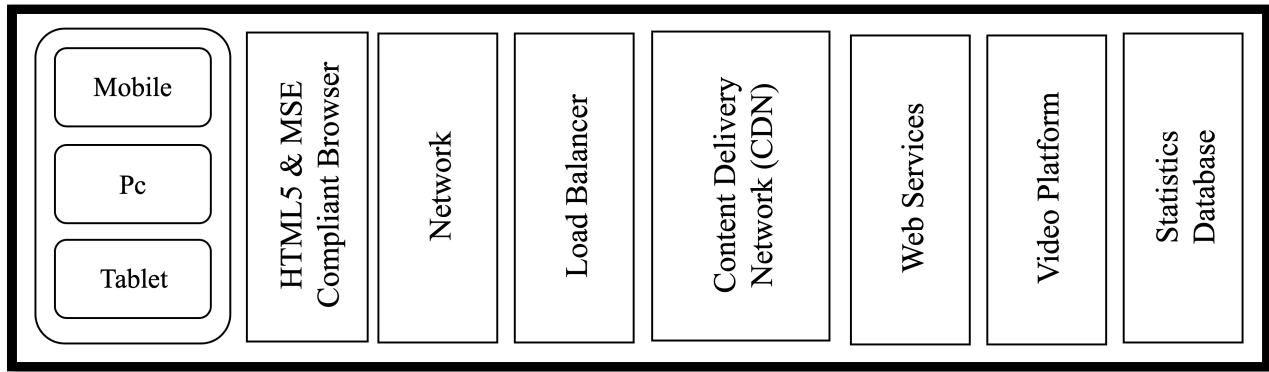


Figure 1. A generalized view of online video platform components

11] are more reluctant to answer a user survey. To overcome these conditions, service providers need a mechanism to estimate what might have gone wrong in the actual workflow by comparing past well-known conditions that was collected from trustworthy observers to improve customer satisfaction.

III. STATE-OF-ART QOE QUESTIONNAIRE IMPLEMENTATIONS

Most of the online services are currently using QoE analysis and consider their product's quality measurements based on the QoE assessments that they receive from the users. Nowadays, it is moderately typical to see a user survey at the end of a Skype call or come across a Facebook satisfaction questionnaire about Facebook's news feed.

Figure 2. QoE Questionnaires for Popular On-line applications (Skype, Facebook and WhatsApp)

Skype uses a survey of one to five stars to grade the overall skype call quality and Facebook asks whether the user is satisfied with news feed content. Popular instant messaging and telephony application “WhatsApp” follows a similar pattern and frequently queries users their service quality with an additional feature of logging personal experience by asking

“Tell us more” as shown in Figure 2. Measuring overall success of a service by considering a five minutes call or social media news feed can be a challenging task as these are comprehensive concepts to be covered by a single value evaluation methodology. The same situation applies for any online video delivery service; it consists of interactions of many different and complex tiers.

Despite the false wide belief, the objective of these methodologies does not intend to understand a single user's perception, but through induction of a real-time modelling, to evaluate the quality perceived by clusters of users distributed on different geographical regions. According to this information, service providers might take action and reconsider their resource management mechanisms on different layers of service including cloud, network, load balancing, routing and CDN. Ultimately, this will enhance the overall success of the online service.

IV. RELATED WORK

Over-the-top content (OTT) technologies bring more content to the users more than ever. Still, a higher QoE might mean more than the context of the content [10]. In this section, both academic and research works on the impact of QoE on OTT will be discussed.

M. Knoll et al have provided a Mean Opinion Score (MOS) model for OTT services [3], where x stands for number of stalls and t for time since last stall and a for the memory parameter (which was set as 0.14) given with Equation 1.

$$MOS = e^{-\frac{x}{5} + 1.5 - a e^{\sqrt{t}}} \quad (1)$$

This equation provides a ground understanding for a single user's perception and mainly relates it to the number of stalls during the watch session. However, the nature of the model cannot reflect a time varying understanding of the experience and obviously it reflects a standalone, single user centric perception. In ITU-T P.1203.3 recommendation [5], a media session quality score is formulated based on number of stalls, total stall duration, buffering duration, media length and compression quality as given with Equation 2.

$$SI = e^{-\frac{numStalls}{s1}} \cdot e^{-\frac{(totalbuf)}{T}} \cdot e^{-\frac{(bufdur)}{T}} \quad (2)$$

These equations; both 1 & 2 reflects single users QoE while

correlating them to video metrics. The environment that is reflected is only the user and the single consumer device that has been used. The medium that is used to transmit the video data has not been taken into consideration. From a service provider's perspective, modelling a single user's perception would not induce a valid model for the delivery system. In this work, primary task is to bring a methodology that will relate video metrics to end-to-end system parameters.

C. Li et al has presented a QoE driven mobile edge caching methodology [40] where for user $u \in U$ using server $s \in S$, ΔT is the time fraction within a video file that is required to be buffered as given with Equation 3. The initial startup delay constraint requires that the waiting time interval between submitting a request and the actual video playback must not exceed the maximum tolerable waiting time of that user, which is denoted as d_u^s .

$$d_u^s = \frac{R_{f,m} \cdot \Delta T}{c(s,u)}, \forall u \in U, \forall s \in S \quad (3)$$

$R_{f,m}$ refers to the bitrate of the video file f for the transcoding rate m . The download link transmission rate of the wireless link from server s to use u is denoted by $c(s,u)$. This model provides a good understanding of initial delay and resolution considering the impact on user's QoE. Yet, it lacks the ability to consider stall duration and total number of stalls that happens through the watching experience. This paper provides a better understanding of user QoE regarding a wide variety of video metrics including total stall duration, number of stalls, initial buffering and resolution at the same time through machine learning modelling. L. Zhou has published [41] a QoE oriented analytical delay indicator for video streaming systems based on a fluid framework model. Fluid dynamics can very well simulate the watching experience due to the fact that video streaming is expected to be similar to "a flowing experience" which circumvents holdups and interruptions. However, the author pointed out in the conclusion part in their own words, "a more practical user response should be considered". As a comparison to [41], our work provides a practical, applicable, easy to integrate scientific methodology for any OTT delivery platform.

F. Wamser et al [4] have provided an extensive collection of objective models for network operators to better understand the OTT traffic in their networks, to predict the playback behaviour of the video player, and to reflect how efficient in delivering OTT videos to their customers. In this model, network measurements including bandwidth capacity, download duration of a block, request duration of a block have been considered. Although network parameters are considered, measurements are only taken within user domain and a conclusion on QoE of the whole service is not possible.

In a recent whitepaper from Cloudstreet [10], a connected city scenario is described where there are many users with different quality and service expectations that are trying to access OTT services. The company introduced a solution where cloud bandwidth-auctioning algorithm that makes intelligent determinations of priority in real time and has effectively provisioned assured QoS/QoE. Gomez et al [11]

presented an Android application, which is able to evaluate and analyse the perceived QoE for YouTube service in wireless terminals. Their application carries out measurements of objective Quality of Service (QoS) parameters, which are then mapped onto subjective QoE (in terms of MOS) by means of a utility function.

The research works [3, 4, 11] have defined and analysed QoE from a content generation and segment size point of view while providing a relation to picture quality only. Nevertheless, QoE definition of this paper will be in correlation with [4, 10] and the intention will be to analyse the concept from service provider's perspective where models are real-time and targeted for clusters of users instead of the single user. Rather than only a measure for picture quality, QoE has been used as a quantity which can be a measure for the whole end-to-end service perception.

V. CONTRIBUTIONS

This paper is based on an experimental QoE platform [46]. The main intention of this work is bringing a methodology to measure the QoE of an online video system and determine QoE capacity from the service provider of view. To achieve this, an online platform has been developed to measure single user QoE with following properties:

1. A video service is implemented to provide random movie trailers that can serve multiple users simultaneously.
2. Any user can watch a different content at a time.
3. Users can stop watching a content anytime they desire and continue on another random content.
4. The resources are randomly reconfigured on platform which changes throughput and latency of the service that corresponds to changes in stalling and buffering behaviour of user video experience.
5. Video metrics (Active watch duration, number of stalls, total stall duration, initial buffering duration), online video platform resource parameters (goodput, latency) and subjective QoE information ($QoE_{overall}$, QoE_{stalls} , $QoE_{initial}$) are collected for each session.

A replicate of the platform is available through Amazon Web Services (AWS) EC2 and accessible via: www.utkubulkan.co.uk/qoe.html. QoE database for the online video delivery platform is available for public access at "www.utkubulkan.co.uk/qoedatabase.php".

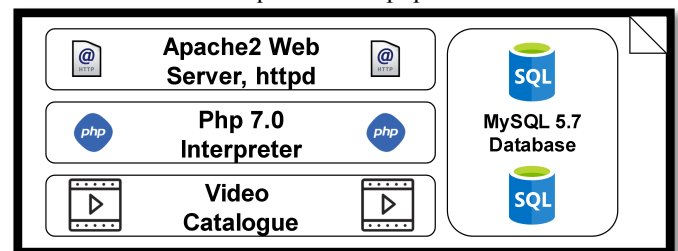


Figure 3. VM Instance application layout

The Virtual Machine (VM) instance runs a collection of applications necessary for online streaming; apache II web server, PHP7.0 interpreter, MySQL5.7 database and a catalogue of video content as presented with Figure 3.

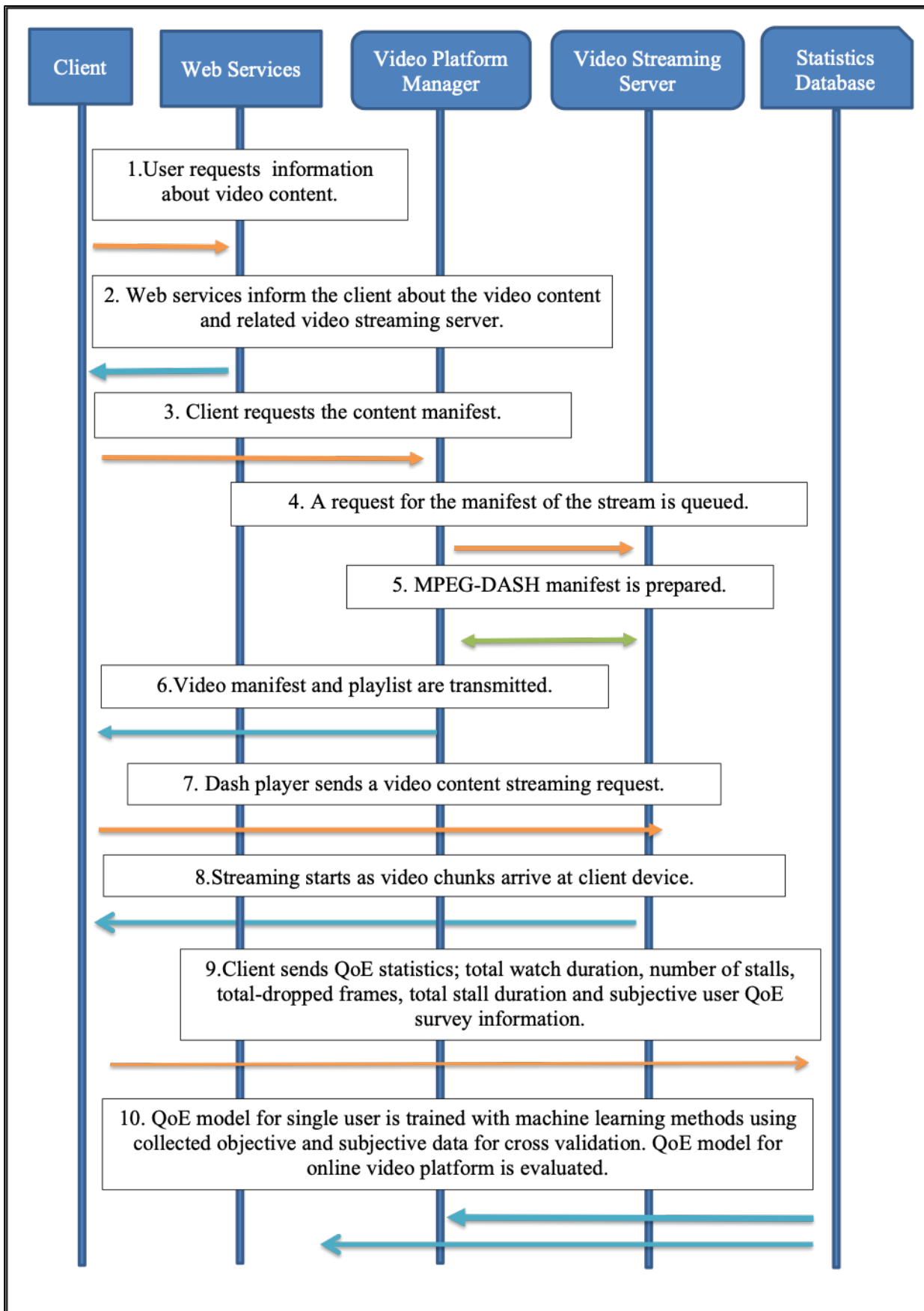


Figure 4. Online Video Platform Workflow

The subjective QoE survey that has been used in the online video platform has been presented in Figure 5. The subjects are queried for their opinions about overall, stall and initial loading time of the watch session.

Can you please answer Quality of Experience questions ?

Overall experience of the video viewing :
 Very bad Bad Moderate Good Very good

How do you feel about "continuity" and "initial loading duration" of the video?
 How do you rate the continuity of the video playback ?
 Very bad Bad Moderate Good Very good

How do you rate the initial loading time?
 Very bad Bad Moderate Good Very good

Submit QoE Survey

Figure 5. Subjective User Survey

The inputs and outputs are used to train, cross validate and test three different machine learning models: ANN, KNN and SVM to predict QoE for a single user. Finally, single user's QoE is used for evaluation of Q_s , online video platform's QoE value. Q_s and the relationship of Q_s with network parameters including goodput and latency will be evaluated. This will provide a fundamental understanding for QoE and end-to-end delivery requirements.

VI. QOE ECOSYSTEM IMPLEMENTATION

The proposed QoE ecosystem consists of five main components; Client, Web services, Video Platform Manager, Video Streaming Service and QoE Database. The workflow diagram that is illustrated in Figure 4 shows the interactions between these components and their influences on calculating QoE.

A. Client

The client can be either a mobile device or personal computer that runs a web browser with Moving Picture Experts Group & Dynamic Adaptive Streaming (MPEG-DASH) over Hypertext Transfer Protocol (HTTP) content playing capability. Dynamic Adaptive Streaming over HTTP (DASH), also known as MPEG-DASH [44], is an adaptive bitrate streaming technique that enables high quality streaming of media content over the Internet delivered from conventional HTTP web servers. MPEG-DASH works by breaking the content into a sequence of small HTTP-based file segments, each segment containing a short interval of playback time of content that is potentially many hours in duration. MPEG-DASH is the first adaptive bit-rate HTTP-based streaming solution that is an international standard.

1) MPEG-DASH Player

Current browsers are not able to handle MPEG-DASH streams by default. They need a JavaScript library such as "dash.js" [19, 20] or "Google's Shaka" [21] player, to understand, parse and feed the chunks of video to the html5 video player. Without loss of generality, Google's Shaka player has been used and "video.js" [26] libraries for MPEG-DASH manifest parsing and stream injection to browser's player.

2) Browser Support

Mobile or PC client must use a browser with HTML5 capability and Media Source Extensions (MSE) [18] to support MPEG-DASH players for playing content that are available in streaming platform. Major browsers that support MSE should be or greater than following versions, Firefox 42, Google Chrome 33, Microsoft Internet Explorer version 11, Safari version 8 [18].

3) Video Metric Collection

02:17

Show Statistics Hide Statistics Increase Quality

Cookie: 6833

Content Name: http://192.168.0.19/video/mpd/swepvii-tlr2_h1080p.mp

Estimated Bandwidth: 7272932

Bitrate: 3976833

Available Bitrates: 625472-180p, 875686-360p, 1121987-480p, 20

Stalls: 2

Total Stall duration : 3

Latency for playback : 0.07

Total Video Frames: 3355

Figure 6. Video Player Application

A player application has been developed using JavaScript & PHP that runs on the client and gathers statistics for the video metrics. These metrics can also be monitored via enabling statistics debug mode. An example screenshot of statistics that video player application shows in debug mode is demonstrated in Figure 6.

B. Web Services

According to the sequence diagram in Figure 4, user requests information about the video services. An HTTP conversation is initiated from client to web server. The web server will reply with the location of the MPD (media presentation description) manifest for the MPEG-DASH content. A CDN consists of many different devices and hence IP addresses which requires access to many different computers and domains. For this reason, Cross Domain Origin Policy (CORS) [12, 13] has been configured to avoid access inconsistencies.

C. Video Platform Manager

1) Transcoding and MPEG-DASH Manifests

Manifest files can be created in many different ways [14, 15, 16, 17]. In this work, FFmpeg has been used to transcode the content on the on-line video platform [14] and Mp4Box [15] has been used for DASH manifests. Major platform suppliers which provide this capability as SaaS include Wowza

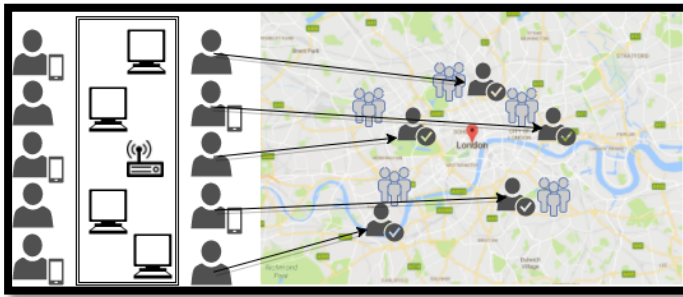


Figure 7. Lab testing (left) and implementation into real life scenario

Streaming Server [16], thePlatform [22], Kaltura [17] etc. These platforms work as a companion company for actual content suppliers such as FOX or ESPN and provide solutions to play their content on all screens and devices.

2) Video Catalogue

The video catalogue consists of 10 different film trailers in the following genres; science fiction, drama, comedy, documentary and action. The duration for each trailer ranges from 2 to 3 minutes. Trailers tend to be short, attention grabbing while provide an exemplification of the entire film. Additionally, availability to download for public use makes them appropriate candidates to be used in a scientific research environment. In these experiments, the catalogue has been transcoded into 5 different resolutions -180p, 360p, 480p, 720p, 1080p with h264 encoding using libx264 with main profile using adaptive bitrate [33]. All these industry standard resolutions are explicitly defined in MPEG-DASH MPD manifest following a similar method such as Youtube and Vimeo use to support adaptive bitrate content streaming.

D. Video Streaming Server

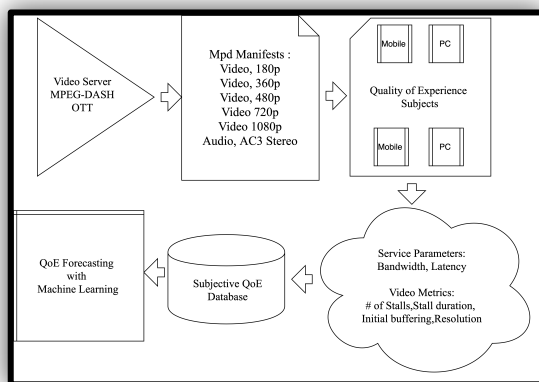


Figure 8. A diagram of proposed QoE Forecasting Methodology [46]

The Linux-based Mpeg-DASH streaming server provides the content to the clients. It interacts with Netem network [23] that manages network throughput limitation and introduces delay in order to simulate real life scenarios like mobile or PC applications working via wireless-mobile networks [24]. Changes in network conditions will force the DASH players in client devices to switch to a more suitable bitrate.

E. QoE Database and Video Metrics Collection

After the video ends, the user is queried with subjective metrics related with user experience. The following subjective QoE values are collected: $QoE_{overall}$ (overall customer satisfaction from whole experience), QoE_{stalls} (level of stalls in user's perspective), $QoE_{initial}$ (initial buffering time in user's perception). Figure 5 shows the subjective user survey that is queried to user at the end of each watched content. The database stores the following information for each content; trailer name, watch duration, initial buffering duration, bitrate, number of stalls and total stall duration.

VII. EXPERIMENT METHODOLOGY

In recent years, QoE aware service experimentation has diverged into two mainstream techniques, traditional lab testing and crowdsourcing. L. Anekeuh et al. [29] have discussed in their recent paper how crowdsourcing is being preferred to lab testing. Without a doubt, crowdsourcing has emerged as a cheaper and quicker alternative to traditional lab testing where people are recruited from different geographical regions with ubiquitous Internet access to perform low cost subjective video quality tests [32]. However, crowdsourcing introduces uncertainty in network/connectivity parameters and reliability of the observers, which might introduce a bias considering the tendency of people to criticize more than praise. T. Hoffeld et al has stated [42] that in general, every crowdsourcing task suffers from bad quality results. Though, even if the task is designed effectively, subjects might still submit unreliable and misleading survey results [31]. Establishing a trustworthy cluster of subjects (either paid or voluntarily registered) across distributed geographical locations that access the service via different network operators will establish a good understanding of the QoE of the service. Decisively, to keep full control on network monitoring capabilities and user consistency, lab-testing methodology is preferred as shown in Figure 7.

The focus of this paper is "QoE of an online video system" rather than a single user's perception. From the perspective of a service provider, metrics of a reliable user's opinion will be the basic building block to train the model for the QoE of the video delivery platform. However, the ultimate goal is not to measure and act upon the satisfaction of each particular customer but the whole service from the providers point of view. Obviously, measurement of a real-time performance of an online service requires QoE to be modelled as a function of time, considering the number of requests at an instant, and their impact on service throughput and latency.

A. Test Subjects and Equipment

Subjects who have participated in the experiment are both undergraduate and postgraduate students in the London South Bank University. A total of 30 users have participated in the testing evaluation. Testers have used 10 different consumer devices including a variety of mobile phones; Samsung S3, S4, Note 3, Sony Xperia XZ which have resolution of 1920x1080, HTC 10 (2560x1440) and personal computers; Dell Latitude e6410 (1280x800), Macbook (2560x1600), HP Elitebook8460 (1366x768), Probook 430 (1366x768) where Firefox or Safari browsers are used depending on OS. All

devices in our lab testing are connected to a wireless router TP-Link TD-W8961N with 300Mbps throughput. Our video services and web services run in Ubuntu 16.04 OS as virtual machines via VirtualBox on a HP-Elite Book with 8GB of RAM and i5 Intel processor.

The left part of Fig. 7 denotes the lab testing methodology and relationship to the real-life online video system scenario. The subjects are representing the recruited observers on the map from cluster of users that are receiving the online video service. Each trustworthy user will reflect the QoE of the system for a particular network operator-CDN and controlled lab environment guarantees valid network metrics monitoring.

B. Information about Movie Trailers

The purpose of a movie trailer is to provide an overview of the context of the motion picture using selected shots throughout the film that is being advertised [45]. A trailer has to achieve that in less than 2 minutes and 30 seconds, the maximum length allowed by the Motion Picture Association of America (MPAA). Each studio or distributor is allowed to exceed this time limit once a year, if they feel it is necessary for a particular film.

C. Test Methodology

During the experiments, subjects request content via their MPEG-DASH players and video platform provides random movie trailers with durations ranging from 2:18 seconds to 2:54s; Theory of Everything (2:29s), Thor II (2:27s), Star Wars 7 (2:18s), Saving Mr. Banks (2:54s), Back In Time (2:32s), James Bond Spectre (2:31), The Intern (2:30s), Independence Day Resurgence (2:31s).

The content is streamed in an asynchronous manner and the content is unicasted to each client where the test has been performed for all participants concurrently as shown in Figure 8. Each subject might watch a different content at a time. Users can either start watching or stop or also even exit in the middle of a session whenever they desire. At runtime, server's goodput and latency are reassigned randomly and this may cause some users to stall and wait for the service to become available again.

In computer networks, goodput is the application-level throughput corresponding to the number of useful information bits delivered by the network to a certain destination per unit of time [43]. This capability simulates the service of an actual online video broadcasting system where the load and the number of requests on the system vary in time.

At any moment, if the subject desires to quit watching (the reason might be anything; number of stalls, stall duration, even unwillingness), she/he simply presses "Exit Watching"

and proceeds with QoE survey. Each time a video is watched and QoE survey is submitted by the user, video metrics, active watch duration, service goodput and latency for each session is logged.

TABLE I
LIST OF NOTATIONS

Notation	Meaning
$u \in U$	Single user, element of all users
$v \in V$	Virtual Network Function (VNF)
$m \in M$	Physical machine
$Q_u^{v,m}(t)$	User's QoE from $v \in V$ running on $m \in M$ at time t
W	User's total watch duration for the content
B	Average bitrate of the stream
S_t	Number of stalls
St_{dur}	Time spent during stalls
t_{lat}	Initial content buffering duration in seconds
S_{gp}	Service goodput
S_l	Service latency
$Q_v^m(t)$	QoE for a VNF $v \in V$, running on $m \in M$, at time t
$Q_s(t)$	QoE for entire system at time t
$QoE_{overall}$	overall customer satisfaction from whole experience
P_{CMA}	Central moving average
P_M	m th observed measurable quantity
pcc	Pearson correlation coefficient
$rmse$	Root mean square error
mae	Mean Absolute Error
X_i, Y_i	i th input & output for the error functions pcc & $rmse$
\bar{X}, \bar{Y}	Sample means for input & output for pcc & $rmse$
$Y_{actual_i}, Y_{calculated_i}$	Calculated and experimental function value for mae
$P(x,y)$	Polynomial cubic fit function with two variables
x^k, y^l	Variables of fit function with relevant power indexes
p_{ik}	Coefficients of polynomial function with indexes i,k
S_t	The total number of stalls
St_{dur}	Time spent during stalls
t_{lat}	The amount of time to load the content
$Q_s(S_{gp}, S_l)$	System QoE as a function of system goodput & latency

TABLE II. LAB SESSION DATA COLLECTED VIA ONLINE VIDEO PLATFORM

User ID	Trailer Name	Trailer Time(s)	Trailer Watched(s)	Watch Rate	Playback Latency	Content Bitrate (Mbits)	Stalls	Exit Video	Stall (s) Duration	Service Goodput (Mbit)	Service Latency (ms)	QoE overall	QoE stalls	QoE initial	Session Time (min)
User-1	theintern	150	128.67	0.86	0	4.00	1	1	1.01	0	0	5	5	5	2
User-7	kedı	125	93.52	0.75	4.655	4.00	3	1	3.01	0	0	4	4	4	2
User-6	skyfall	151	88.19	0.58	0.509	4.00	3	1	10.03	0	0	2	2	5	2
User-5	theintern	150	149.17	0.99	0.852	4.00	10	0	26.93	0	0	1	1	5	2
User-2	sw-vii	138	10.42	0.08	2.585	3.00	3	0	15.03	0	0	1	1	4	2
User-8	sw-vii	138	22.32	0.16	1.144	4.00	1	1	0.00	0	0	5	5	5	3
User-6	backintime	152	38.85	0.26	0.502	4.00	1	1	3.01	0	0	5	4	5	3
User-1	thor2	147	119.52	0.81	1.313	4.00	2	1	6.02	43	979	4	4	5	5
User-3	sw-vii	138	137.17	0.99	0.319	4.00	0	0	0.00	43	979	5	5	5	5
User-6	backintime	152	81.75	0.54	8.279	4.00	1	1	1.00	43	979	2	5	2	7
User-2	skyfall	151	93.65	0.62	10.53	3.00	2	1	13.03	46	945	1	2	1	7
User-8	theintern	150	150.00	1.00	1.859	4.00	5	0	18.06	46	945	1	1	5	8
User-6	independenceday2	151	35.14	0.23	8.448	4.00	1	1	6.02	46	945	1	4	2	8
User-7	independenceday2	151	142.49	0.94	14.797	4.00	3	0	10.04	46	945	2	2	1	8
User-1	theintern	150	150.00	1.00	1.907	4.00	5	0	22.14	38	854	1	1	5	10
User-2	theoryofeverything	149	85.06	0.57	4.169	4.00	5	1	24.12	38	854	1	1	4	10
User-5	savingmrbanks	174	29.17	0.17	0.521	3.00	14	0	30.87	38	854	1	1	5	10
User-2	sw-vii	138	24.01	0.17	3.256	4.00	1	1	3.01	38	854	2	4	4	11
User-6	savingmrbanks	174	104.70	0.60	7.329	3.00	6	1	22.10	38	854	2	1	2	11
User-8	sw-vii	138	137.34	1.00	3.648	3.00	2	0	14.03	38	854	1	1	4	11
User-2	kedı	125	15.97	0.13	7.45	3.00	1	1	4.01	38	854	2	4	2	12
User-4	thor2	147	129.43	0.88	4.448	3.00	8	0	26.71	37	617	1	1	4	12
User-7	skyfall	151	150.21	0.99	14.913	4.00	3	0	10.04	37	617	4	2	1	12
User-8	theintern	150	5.88	0.04	3.36	4.00	2	1	2.01	37	617	4	5	4	12
User-4	theintern	150	24.67	0.16	0.702	4.00	1	1	7.07	37	617	4	3	5	12
User-5	thor2	147	46.55	0.32	1.238	4.00	6	0	20.98	37	617	1	1	5	13
User-6	sw-vii	138	138.00	1.00	1.761	3.00	4	1	10.03	37	617	3	2	5	14
User-5	skyfall	151	39.03	0.26	5.246	3.00	4	0	24.96	40	245	2	1	3	15
User-7	thor2	147	129.22	0.88	3.857	4.00	2	0	8.08	40	245	3	3	4	15
User-2	skyfall	151	150.26	1.00	6.336	3.00	3	0	7.02	40	245	3	3	3	16
User-8	skyfall	151	150.30	1.00	6.429	3.00	2	0	6.01	40	245	4	4	3	16
User-2	backintime	152	10.76	0.07	1.906	3.00	0	1	0.00	40	245	5	5	5	16
User-8	theintern	150	76.67	0.51	0.251	3.00	4	1	8.02	38	593	4	3	5	18
User-4	savingmrbanks	174	174.00	1.00	2.881	4.00	4	0	7.06	38	593	3	3	4	18
User-7	skyfall	151	149.97	0.99	3.843	3.00	5	0	8.00	38	593	3	3	4	19
User-6	sw-vii	138	71.25	0.52	0.879	3.00	7	1	20.00	23	762	1	1	5	19
User-8	theintern	150	90.29	0.60	0.884	2.00	1	1	1.00	23	762	5	5	5	19
User-4	backintime	152	16.13	0.11	8.523	2.00	1	1	0.00	23	762	2	5	2	20
User-5	independenceday2	151	5.88	0.04	7.843	3.00	8	0	20.00	23	762	1	1	2	21
User-8	kedı	125	64.16	0.51	8.367	3.00	1	1	4.00	23	762	2	4	2	21
User-6	independenceday2	151	55.62	0.37	7.513	2.00	2	1	5.00	23	762	3	4	2	21
User-6	kedı	125	87.14	0.70	4.034	2.00	2	1	2.00	21	44	5	5	4	23
User-2	thor2	147	129.29	0.88	0.878	3.00	5	0	6.00	21	44	4	4	5	24
User-7	skyfall	151	150.17	0.99	13.644	2.00	2	0	9.00	42	77	1	3	1	24
User-5	backintime	152	152	1.00	4.304	1.00	4	0	16.00	42	77	1	1	4	24
User-8	theoryofeverything	149	149	1.00	1.014	1.00	1	1	0.00	42	77	5	5	5	26
User-6	independenceday2	151	142.61	0.94	4.001	3.00	2	0	3.00	42	77	4	4	4	26
User-2	independenceday2	151	142.57	0.94	2.826	3.00	3	0	4.00	25	994	4	4	4	27
User-4	theoryofeverything	149	149	1.00	1.756	3.00	3	0	25.00	25	994	1	1	5	27
User-7	theoryofeverything	149	148.79	1.00	0.39	2.00	1	0	0.00	25	994	5	5	5	27
User-5	theintern	150	0	0.00	1.198	2.00	5	1	13.00	25	994	3	2	5	28
User-6	sw-vii	138	30.96	0.22	0.367	1.00	2	1	15.00	25	994	1	1	5	28
User-8	skyfall	151	83.4	0.55	0.811	1.00	4	1	20.00	25	994	1	1	5	28
User-8	theintern	150	66.21	0.44	2.049	1.00	4	1	7.00	25	501	3	3	4	30
User-6	skyfall	151	104.48	0.69	6.25	1.00	3	1	15.00	25	501	1	1	3	31
User-1	savingmrbanks	174	174	1.00	3.252	2.00	8	0	19.00	25	501	2	1	4	31
User-7	sw-vii	138	137.17	0.99	11.386	3.00	5	0	22.00	25	501	1	1	1	31
User-9	thor2	147	128.96	0.88	1.119	2.00	9	0	17.00	25	501	1	1	5	31
User-5	theintern	150	0	0.00	3.378	2.00	8	1	20.00	12	628	3	1	4	32
User-4	theintern	150	150	1.00	0	1.00	6	0	21.00	12	628	1	1	5	32
User-8	kedı	125	119.86	0.96	4.928	1.00	3	1	8.00	12	628	2	3	4	32
User-1	skyfall	151	150	0.99	9.16	1.00	7	0	21.00	34	687	1	1	1	34
User-4	kedı	125	114.22	0.91	5.891	1.00	4	1	13.00	34	687	4	2	3	35

D. Derivation of Overall Online Video System QoE

The ultimate aim of this paper is to bring forward a scientific methodology to evaluate QoE for the video delivery system from a single subject oriented QoE. In order to establish this association, a user's QoE has to be defined in terms of video quality metrics. Eq. 4 stands for the abstract representation of the input-output relationship that the machine learning methods SVM, KNN and ANN are based on in Section VIII. For the functions in this section, the definitions of the variables are declared in Table 1 as list of notations.

For a user using the service on virtual machine "v" that is running on physical machine "m", single user's QoE " $Q_u^{v,m}(t)$ " is represented as Equation 4 as a function of W (total watch duration), B (average bitrate of the stream), St (number of stalls), St_{dur} (time spent during stalls), t_{lat} (the amount of time to load the content).

$$Q_u^{v,m}(t) = Q(W, B, St, St_{dur}, t_{lat}) \quad (4)$$

Following that, an online video platform consists of several distributed video servers and CDN nodes. This needs a clarification and classification for the particular user and physical & virtual server that provides the service for that user. The users are getting this service from a Virtual Network Function (VNF) $v \in V$ and this VNF runs on a physical machine $m \in M$ at a moment t. The QoE for v can be defined as Eq. 5:

$$Q_v^m(t) = \sum_{u=1}^u \frac{Q_u^{v,m}(t)}{u} \quad (5)$$

Conclusively, QoE of the service can be reflected as the success of corresponding $\forall v \in V$ that build up the entire system.

$$Q_s(t) = \sum_{v=1}^v \frac{Q_v^m(t)}{v} \quad (6)$$

In order to reflect the local behavior of QoE, Q_v and eventually Q_s are calculated with central moving average (P_{CMA}) [30, 38] that spans through the Q_U dataset as declared in Table II and given with Equations 7 & 8.

$$P_{CMA} = \frac{p_M + p_{M-1} + \dots + p_{M-(n-1)}}{n} \quad (7)$$

$$P_{CMA} = p_{CMA,prev} + \frac{p_{CMA} - p_{CMA-n}}{n} \quad (8)$$

In this context, single v has been used, so Q_s & Q_v to the same entity. As a future work, a load balancer mechanism, support for multiple-CDN and edge cache node support will be implemented for system quality attributes analysis: scalability, resilience, responsiveness and availability.

E. Results and Discussions

Six lab-testing sessions have been conducted with the subjects. The duration of each session was about thirty minutes. All participants have used the limited resources of the online video platform simultaneously. The collected data have been used for modelling with K-nearest Neighbours Algorithm (KNN), Artificial Neural Network (ANN) and Support Vector Machine (SVM) on a MacBook Pro running Matlab R17 with i5 processor and 16 GB RAM.

VIII. SUPERVISED MACHINE LEARNING USING OBJECTIVE METRICS AND SUBJECTIVE SURVEY

A. SVM, support vector machines

Support Vector Machines (SVM) categorizes data by discovering the linear decision boundary (hyperplane) that separates all data points of one class from those of the other class [34]. Once the model parameters are recognized, SVM relies only on a subclass of these training cases, termed support vectors, for future estimations. An increase on the weight c (box-constraint) will cause stricter separation of the classes. However, this may introduce an increase factor in false assumptions on the classification.

B. KNN, k-nearest neighbor classification

The K-Nearest Neighbor (KNN) classification technique categorizes objects regarding the classes of their nearest neighbors [34]. KNN forecasts are based on the supposition that objects near each other should be similar. During the learning phase, the best number of similar observations has been chosen [35]. In order to ensure that models generated using different values of k are not overfitting, a separate training and cross validation test set have been used.

C. ANN, artificial neural networks

Inspired by the human brain, a neural network consists of highly connected network of neurons that relate the inputs to the desired outputs [34]. ANN is quite efficient for modelling highly nonlinear systems and unexpected changes are anticipated in the input data.

D. Training Methodology

For any attempt to use machine learning modelling for simulating the behavior of a function, the methodology of training using the available dataset plays a crucial role for the base understanding of the mathematical endeavor. In this work three different machine learning methods have been employed and this section presents a clarification of the training phase.

In order to train a SVM, there needs to be couple of foundational decisions taken regarding; how to preprocess data and what kernel to use [47]. In this work, cubic kernel model has been used for SVM. Three different values for box-constraint have been taken and the results have been presented in Table III.

For the KNN models that have been used in this work, distance metric has been selected as Euclidean with equal distance weights. The accuracy of three different neighbour settings has been presented in the results table.

The ANN models have been modelled with three different f neurons settings; 8, 10 & 12. The network is trained with Levenberg-Marquant training algorithm modifying the strengths of the connections so that given inputs map to the correct response.

During lab testing, over 400 watched session information have been collected regarding the input & output relations where user's QoE " $Q_u^{v,m}(t)$ " is modelled with parameters; W , B , St , St_{dur} & t_{lat} as given with Eq. 4. The data collected from these experiments are used for training the models and cross validation. A set of these streaming sessions have been presented as Table II. Regarding this dataset, in the following section, confusion matrixes for different machine learning models will be presented where predicted values will be compared against true values. The classes that are given in the confusion matrixes in section VIII.E refers to subjective QoE evaluation labels; 1 - Very Bad, 2 - Bad, 3 - Moderate, 4 - Good and 5 - Very Good.

E. Confusion Matrix

The confusion matrix (also known as the error matrix) provides the scattering of the correct match rates for predicted versus true classes as shown in Figure 9, 10 & 11. True positive rate reflects correct hit levels while false negative rate provides the miss percentage. KNN QoE model has shown the best accuracy rates with a setting of 10 neighbors.

True Class	5	%93	%2	%1		%5	True Positive Rate	False Negative Rate
	4	%1	%87	%1	%3	%7		
	3	%1	%3	%82	%7	%6		
	2	%2	%12	%11	%74	%2		
	1	%20	%12	%6	%1	%60		
	Predicted Class	1	2	3	4	5		

Figure 9. Confusion Matrix for KNN QoE Model. Weighted KNN has been implemented with a setting of 10 neighbors. Distance metric is Euclidean with squared inverse distance weight. The accuracy of the

true positive rates %82.6 have been presented via confusion matrix.

True Class	5	%86	%2	%2		%10	True Positive Rate	False Negative Rate
	4	%4	%72	%3	%6	%15		
	3	%3	%1	%78	%9	%9		
	2	%5	%9	%11	%72	%4		
	1	%10	%4	%2	%1	%83		
	Predicted Class	1	2	3	4	5		

Figure 10. Confusion Matrix for SVM Cubic Kernel Model, with the box-constraint value as $c=3$.

True Class	5	%80	%10			%10	True Positive Rate	False Negative Rate
	4	%4	%65	%4	%10	%17		
	3	%6	%6	%72	%10	%6		
	2	%2	%14	%11	%72	%2		
	1	%7	%10		%2	%80		
	Predicted Class	1	2	3	4	5		

Figure 11. Confusion Matrix for ANN with 10 neurons. The network is trained with Levenberg-Marquant training algorithm.

This may be due to the fact of lazy learning and KNN's capability to distinguish neighbouring class features, when a strict clustering is not possible across the dataset due to bias. Still, performance of SVM & ANN is very close and selecting a methodology for implementing one of these methods should rely on empirical confirmation for each test setup and session.

F. Experiment Dataset and Error Analysis

Regarding calculated and actual qualitative values, the error has been measured with three different methods [37]: Pearson correlation (Eq. 10), root mean square error (Eq. 11) and mean average error (Eq. 12). The definitions of the arguments in these equations are explicitly described in Table 1 as list of notations.

The Pearson correlation measures the linear association between a model's performance and the subjective QoE. It provides a standard scale of -1 to 1 : 1 indicates a total positive correlation, 0 no linear correlation and -1 total negative correlation.

$$pcc = \frac{\sum_{i=1}^N (X_i - \bar{X}) * (Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2} * \sqrt{\sum (Y_i - \bar{Y})^2}} \quad (10)$$

Root mean square error is the square root of the average of squared errors. Despite the general false assumption, it does not reflect the average error. Due to the fact that the square of the error has been used, greater error rates have greater impact on the rmse. A lower value of rmse indicates a better correlation between prediction of the model and actual values.

$$rmse = \sqrt{\frac{1}{N-d} \sum (Y_i - \bar{Y}_i)^2} \quad (11)$$

Mean average error provides a simple analysis of the average difference between prediction and real values. The difference error is proportional to the absolute difference of actual and calculated.

$$mae = \frac{1}{n} \sum_{i=1}^n |Y_{actual_i} - Y_{calculated_i}| \quad (12)$$

TABLE III
QOE ERROR ANALYSIS FOR MACHINE LEARNING METHODS FOR DIFFERENT SETTINGS

ML configuration CV=5	pcc	rmse	mae
Svm (box-constraint c= 1)	0.8973	0.2882	0.489
Svm (box-constraint c= 3)	0.883	0.2141	0.4739
Svm (box-constraint c= 5)	0.9048	0.2224	0.5012
Ann 8 neurons	0.8612	0.2556	0.4912
Ann 10 neurons	0.8998	0.2007	0.4877
Ann 12 neurons	0.8842	0.2217	0.467
Knn (neighbor = 5)	0.8423	0.196	0.4993
Knn (neighbor = 10)	0.91653	0.1924	0.4622
Knn (neighbor = 15)	0.8843	0.2163	0.4593
ML configuration CV=10			
Svm (box-constraint c= 1)	0.8626	0.2536	0.503
Svm (box-constraint c= 3)	0.8788	0.2522	0.4623
Svm (box-constraint c= 5)	0.8955	0.2489	0.4599
Ann 8 neurons	0.8849	0.2245	0.4875
Ann 10 neurons	0.8799	0.2362	0.4636
Ann 12 neurons	0.899	0.2369	0.4663
Knn (neighbor = 5)	0.8363	0.1941	0.4475
Knn (neighbor = 10)	0.8345	0.1963	0.4585
Knn (neighbor = 15)	0.845	0.2045	0.4691
ITU-T P.1203.3	0.9035	0.2135	0.4598
Knoll et al.	0.8761	0.2193	0.4524

G. K-fold Cross Validation

K-fold cross validation (CV) is a model validation technique, which partitions the dataset into equal sized subsets. Single subset is used as validation data while the rest k-1 subsets are used as training data. Spanning through all dataset k-fold times guarantee each data as training data exactly once. Results from the folds can be averaged to produce a single estimation [39]. In this work, three different cross validation training strategies have been conducted with 3, 5 and 10 k-fold values and two of them have been presented.

IX. MODEL PERFORMANCE COMPARISON

Performance of SVM QoE model changes thoroughly with different values of box-constraint configuration in Matlab's fitcecoc functionality [34]. The best results of SVM for Pearson correlation have been achieved with a value of c=3 while rmse provides better results for the c=5 function. Additionally, k-fold cross validation values with k=5 subsets give better values than k=10.

ANN gives good results for real time analysis due to its dynamic programming nature and continuous training capability, which makes it a perfect candidate for a QoE modelling system implementation. Though ANN's real-time capabilities, when compared with other methods, ANN provide the worst performance for QoE modelling for this dataset. With a setting of 10 hidden neurons, ANN provides an estimation performance of pcc \approx 0.89 and rmse \approx 0.2.

KNN, although a lazy learning methodology, shows the best results with the 10 neighbor settings and k-fold=5 (pcc \approx 0.91, rms \approx 0.19, mae \approx 0.4622) compared to all three methods. Commonly, SVM and ANN provide better solutions when compared to KNN for nonlinear variables. However due to the nature of our methodology, bias that might have been caused by user subjective observations might lead SVM to fail distinct classification of input data whereas KNN might have shown better results in mimicking the neighboring classes.

Time-invariant models of ITU-P.1203.3 [5] and Knoll et al [3] have shown parallel behavior and reflected single user. However, from overall perspective, machine learning methods provide a better understanding of the trends of the QoE, in regard to their learning and cross-validation ability directly from the same dataset.

Principally, on such subjective tests where long periods of testing are needed, one of the key facts that must be considered is the exhaustion of test subjects, which may cause unreliable MOS values. In order to avoid such misleading conclusions, precautions such as user's intention to watch particular genre or user's wish for participating in such an experiment at any point is considered during experimentation as discussed by B. Gardlo et al [32].

X. ONLINE VIDEO PLATFORM QOE CAPACITY ESTIMATION

The prime intention of this paper is to measure the QoE vs online video delivery platform capacity parameters. In order to achieve that, single user experience is taken as the elementary unit. After training a model for Q_u , system wide QoE is calculated. The relationship of Q_s vs online video delivery platform goodput and latency has been shown on Figure 12. Equation 13 is a cubic polynomial function in its generalized form [48]. In order to fit Q_s , the arguments goodput and latency have been used in Eq. 13 and their relationship Eq. 14 have been obtained. The coefficients also have been declared in Figure 12.

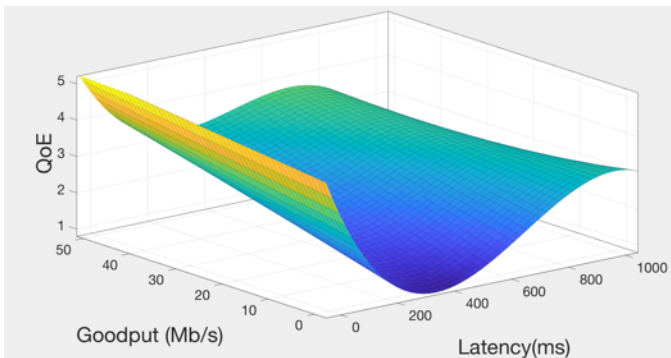


Figure 12. Online Video Platform QoS “ Q_S ” vs Goodput and Latency. The model that is given as Equation 14 relates system QoS to goodput and latency and has following coefficients for the polynomial: $p_{00} = 2.205$, $p_{10} = 1.01$, $p_{01} = 0.6451$, $p_{20} = 0.7613$, $p_{11} = -0.1645$, $p_{30} = -0.8037$, $p_{21} = -0.2947$. Goodness of fit, R -square: 0.07697, RMSE: 0.736.

$$P(x,y) = \sum_{k=0}^3 \sum_{i=0}^1 p_{ij} \cdot x^k \cdot y^i \quad (13)$$

$$Q_S(S_{gp}, S_l) = p_{00} + p_{10} \cdot S_{gp} + p_{01} \cdot S_l + p_{20} \cdot S_{gp}^2 + p_{11} \cdot S_{gp} \cdot S_l + p_{30} \cdot S_{gp}^3 + p_{21} \cdot S_{gp}^2 \cdot S_l \quad (14)$$

Providing a distinct understanding of system QoS information will help any online video delivery platform and service provider to take appropriate action regarding the orchestration of their system resources.

The proof of concept platform that has been discussed in Section VI consists of a single network device and a single virtual machine running on a physical server. Streaming capacity of the online video platform which refers to the obtainable bandwidth that can be served with a tolerable latency while providing an adequate perception quality from this service can be declared as a function of network capabilities, goodput and latency. From the perspective of this paper, Eq. 13 & Eq. 14 present an understanding of QoS in terms of system resource metrics which are modelled using Matlab cubic curve fitting tool [48] with cubic interpolation configuration based on the subjective user experience records and objective experiment statistics. The data that is used as input for the curve fitting tool has been collected through the lab sessions which are available on the publicly accessible database in Section V. The polynomial coefficients of the fit function are declared in information section of Figure 12 and the variables for these equations are defined in list of notations as Table I.

Any QoS degradation that concludes as stalls or initial buffering durations can be prevented by refining existing resources or providing additional capability to system. Real life scenarios rely on many servers, running multiple instances of virtual machines and several network peripherals. When proceeding with this kind of experimentation, empirical validation of the test bed and parallelism to real life scenarios should always be carefully considered.

For a given goodput and minimum latency request, QoS can be estimated with Eq 14. Whenever there is more demand for

content, correspondingly there is a probability for QoS degradation as the load increases. Depending on the advertised service quality such as; basic service (Youtube, Vimeo) or advanced & premium service (Amazon Prime, Netflix, Youtube Premium) the intended and expected QoS levels can be adapted. One important point for the operator is to consider the QoS changes through time and decide when to act against QoS deprivation comparing the delta between two instants during serving period.

This work provides a foundation for scaling strategies of an online video platform. Whenever there is more demand for video which corresponds to relative increase in goodput and latency, Eq. 13 will provide the QoS value in regard with system resources.

XI. CONCLUSION & FUTURE WORKS

The work has provided an evaluation methodology for video delivery system QoS ‘ Q_S ’ through single user Q_U and showed that modelling is possible through objective video metrics and subjective QoS survey analysis. System performance parameters goodput and latency can be associated with user experience, whereas a controlled testing environment is available guaranteeing reliable network performance measurement when network metrics are introduced into numerical prediction analysis.

The methodology that has been proposed in this paper can provide a fundamental understanding on how to act for QoS degradation for online video platforms. This paper can be guideline for any network operator on how to maintain resources; instantiate or terminate VMs responsible for streaming content that will save cloud budgets and deployment costs while considering QoS.

As an extension to our research work, implementation of a load balancer with multi-CDN support is planned while considering cloud computing resource constraints to cover wide variety of needs of future online video trends.

REFERENCES

- [1] Akamai, “Maximizing Audience Engagement: How online video performance impacts viewer behavior”, 10/2016.
- [2] Akamai, “How Akamai Defines And Measures Online Video Quality White Paper, 02/2016.
- [3] T. Knoll et al, “QoS Evaluation and Enforcement Framework for Internet Services”, In Itu Study Period 2013-2016, Chemnitz University of Technology, Germany.
- [4] F. Wamser et al, “Modeling the YouTube stack: From packets to quality of experience”, 2016.
- [5] ITU-T, “P.1203.3, Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport –Quality integration module”, 2016.
- [6] M. Cheon et al, “Evaluation of objective quality metrics for multidimensional video scalability”, Republic of Korea, 2015.
- [7] M. Li et al, “On quality of experience of scalable video adaptation”, Singapore, 2013.
- [8] L. Anegekuh et al, “Content-Based Video Quality Prediction for HEVC Encoded Videos Streamed Over Packet Networks”, 2015 IEEE.
- [9] Y. Zhu et al, “Understanding the role of social context and user factors in video Quality of Experience”, The Netherlands, 2015.
- [10] Cloudstreet, “Mobile OTT Solved – Delivering a flawless mobile OTT Video Experience Whitepaper”, [online] cloudstreet.co, June 2016.

- [11] G. Gómez et al, "YouTube QoE evaluation tool for Android wireless terminals", Malaga, Spain, 2014.
- [12] S. Hsiao et al, "A Secure Proxy-Based Cross-Domain Communication for Web Mashups", Taipei, Taiwan, 2011.
- [13] K. H. Larson, Mozilla Developer Network, "HTTP access control (CORS)", [online], https://developer.mozilla.org/en-US/docs/Web/HTTP/Access_control_CORS. Last accessed on 15 August 2018.
- [14] L. Xiaohua et al, "Design and Implementation of a Real-time Video Stream Analysis System Based on FFMPEG", Beijing, China, 2013.
- [15] J. Ozer, "Encoding and Delivering to Multiple ABR Formats", Streaming Media Industry Sourcebook p168-172, 2016.
- [16] Wowza Technologies, "How Wowza Media Systems Powers Music: Live and On-Demand Video Streaming with TourGigs, Music Choice, and Microsoft Azure", p167.
- [17] S. David, "Kulture Whitepaper, Generation 'I' And The Future Of Tv, Some predictions for 2017", [online], <https://corp.kaltura.com/sites/default/files/generacioni.pdf>. Last accessed on 15 August 2018.
- [18] Mozilla Developer Network, "Media Source Extensions API", [online], https://developer.mozilla.org/en-US/docs/Web/API/Media_Source_Extensions_API. Last accessed on 15 August 2018.
- [19] J. Kornich, "Embedding a MPEG-DASH Adaptive Streaming Video in an HTML5 Application with DASH.js", [online], <https://docs.microsoft.com/en-us/azure/media-services/media-services-embed-mpeg-dash-in-html5>. Last accessed on 15 August 2018.
- [20] R. F. Moyano et al, "A user-centric SDN management architecture for NFV-based residential networks", Spain, 2017.
- [21] Wowza Media Systems, "How to use Google Shaka Player with Wowza Streaming Engine (MPEG-DASH)", [online], <https://www.wowza.com/docs/how-to-use-google-shaka-player-with-wowza-streaming-engine-mpeg-dash>. Last accessed on 15 August 2018.
- [22] Comcast Technology Solutions Whitepaper, "MPX: VIDEO MANAGEMENT SYSTEM", 2016.
- [23] A. Jurgelionis et al, "An Empirical Study of NetEm Network Emulation Functionalities", Norway, 2011.
- [24] E. Ancillotti et al, "Load-aware routing in mesh networks: Models, algorithms and experimentation", Computer Communications, P948-961, Italy, 2010.
- [25] A. Zielesny, "From Curve Fitting to Machine Learning, An Illustrative Guide to Scientific Data Analysis and Computational Intelligence", Springer, 2011.
- [26] S. Heffernan, "Building an HTML5 Video Player" [online], Streaming Media Industry Sourcebook p166-169, 2012.
- [27] H. Nourikhah et al, "Impact of service quality on user satisfaction: Modeling and estimating distribution of quality of experience using Bayesian data analysis", Elsevier, 2016.
- [28] S. Nissen et al, "Implementation of a Fast Artificial Neural Network Library (FANN)", Copenhagen, 2003.
- [29] L. Anekeuh et al, "A Screening Methodology for Crowdsourcing Video QoE Evaluation", Communications QoS, Reliability and Modelling Symposium p1152-1157, Plymouth, 2014.
- [30] A. Raudys et al, "Pareto Optimised Moving Average Smoothing For Futures And Stock Trend Predictions", Insight for Stock Market Modeling and Forecasting p480-483, Lithuania, 2016.
- [31] T. Volk et al, "Crowdsourcing vs. laboratory experiments – QoE evaluation of binaural playback in a teleconference scenario", Computer Networks Elsevier, Germany, 2015.
- [32] B. Gardlo et al, "Crowdsourcing 2.0: Enhancing Execution Speed and Reliability of Web-based QoE Testing", Communication QoS, Reliability and Modeling Symposium, 2014, Austria.
- [33] R. Garrido-Cantos et al, "On the impact of the GOP size in a temporal H.264/AVC-to-SVC transcoder in baseline and main profile", Multimedia Systems, 2013, 19: 163.
- [34] Mathworks, "Applying Supervised Learning", Machine Learning Ebook, Section4.
- [35] G. J. Myatt et al, "Making Sense of Data I : A Practical Guide to Exploratory Data Analysis and Data Mining", p168-170, Wiley, 2014.
- [36] M. Awad et al, "Efficient Learning Machines, Theories, Concepts, and Applications for Engineers and System Designers", p42.
- [37] ITU-T, "Perceptual visual quality measurement techniques for multimedia services over digital cable television networks in the presence of a reduced bandwidth reference", J.246, 08-2008.
- [38] https://en.wikipedia.org/wiki/Moving_average. Last accessed on 15 August 2018.
- [39] [https://en.wikipedia.org/wiki/Cross-validation_\(statistics\)](https://en.wikipedia.org/wiki/Cross-validation_(statistics)). Last accessed on 15 August 2018.
- [40] C. Li et al, "QoE-Driven Mobile Edge Caching Placement for Adaptive Video Streaming", IEEE Transactions on Multimedia, Vol. 20, No. 4, 2018, China.
- [41] L. Zhou, "QoE-Driven Delay Announcement for Cloud Mobile Media", IEEE Transactions On Circuits And Systems For Video Technology, Vol. 27, No. 1, January 2017, China.
- [42] T. Hoßfeld et al, "Quantification of YouTube QoE via Crowdsourcing", IEEE International Symposium on Multimedia, 2011, Germany.
- [43] <https://en.wikipedia.org/wiki/Goodput>. Last accessed on 15 August 2018.
- [44] https://en.wikipedia.org/wiki/Dynamic_Adaptive_Streaming_over_HTTP. Last accessed on 15 August 2018.
- [45] [https://en.wikipedia.org/wiki/Trailer_\(promotion\)](https://en.wikipedia.org/wiki/Trailer_(promotion)). Last accessed on 15 August 2018.
- [46] U. Bulkan et al, "Predicting Quality Of Experience For Online Video Systems Using Machine Learning", 19th IEEE International Workshop on Multimedia Signal Processing MMSp, 2017, UK.
- [47] A. Ben-Hur et al, "A User's Guide to Support Vector Machines", [online content], <http://pymml.sourceforge.net/doc/howto.pdf>, USA.
- [48] <https://uk.mathworks.com/products/curvefitting.html>. Last accessed on 15 August 2018.