IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 October 2023, accepted 8 November 2023, date of publication 20 November 2023,
date of current version 29 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3334623

RESEARCH ARTICLE

When Should Recommenders Account for Low QoS?

MATEUS SCHULZ NOGUEIRA[1], (Member, IEEE), JOÃO ISMAEL DAMASCENO PINHEIRO[1], FELIPE ASSIS[1], DANIEL SADOC MENASCHÉ[1], (Member, IEEE), PAVLOS SERMPEZIS[2], AND THRASYVOULOS SPYROPOULOS[3]

[1]Institute of Computing, Federal University of Rio de Janeiro, Rio de Janeiro 27941-590, Brazil
[2]School of Informatics, Aristotle University of Thessaloniki, Biology Building Aristotle University Campus, 54124 Thesssaloniki, Greece
[3]Department of Electrical and Computer Engineering, University of Crete, Voutes University Campus, Heraklion, 70013 Crete, Greece

Corresponding author: Mateus Schulz Nogueira (msznogueira@gmail.com)

This work was supported in part by the Greece and European Union through the Operational Program Competitiveness, Entrepreneurship and Innovation under the Call RESEARCH-CREATE-INNOVATE under Project T2EDK-04937 and Project T2EDK-03898; in part by the European High-Performance Computing Joint Undertaking under Grant 951732; and in part by the Foundation Coordination for the Improvement of Higher Education Personnel (CAPES), Council for Scientific and Technological Development (CNPq), and Rio de Janeiro State Research Support Foundation (FAPERJ) under Grant E-26/203.215/2017, Grant E-26/211.144/2019, and Grant E-26/201.376/2021.

ABSTRACT Content recommendation systems, also known as recommenders, are pervasive and have significant impact on user demands over the Internet. Platforms such as YouTube and Netflix constantly seek to improve their recommendation systems, to provide better quality of experience (QoE) for their users. QoE, in turn, depends on a multitude of factors, including the quality of recommendation (QoR), e.g., based on users histories and content categories, and the quality of service (QoS), e.g., measured by network delay and throughput. Even though QoS is key in a best-effort Internet, existing recommendation systems overlook it, resulting in recommendations that are suboptimal in terms of QoE. In this study, our goal is to devise a QoS-aware, QoE-friendly, content recommendation system and indicate its feasibility in the wild. For this purpose, we conducted an experiment with real users driven by the following question: When should recommenders account for low QoS? Each user is requested to evaluate pairs of videos, that vary in their contents and QoS levels. We experimentally determined category-dependent thresholds that determine the sensitivity of users with respect to QoS and QoR. Given the collected insights on QoS-aware recommendations, we considered our second research question: Can recommenders compensate for low QoS? We conducted experiments over the Internet, relying on YouTube API and network measurements tools, and report our findings on *(i)* the characterization of QoS and *(ii)* the compensation for low QoS. Our measurements suggest that content far from the trends tends to be far from the user. We quantified the extent to which unpopular content tends to be served with a lower QoS and established a methodology to determine the relationship between content popularity and its physical proximity to users. Then, we verified that making requests a bit trendier can hit much closer content.

INDEX TERMS Network measurements, quality of experience, quality of service, recommenders.

I. INTRODUCTION

Recommendation systems drive a significant fraction of the demand for content on the Internet. More than 50% of YouTube user requests come from their recommendations [1]. In Netflix, this number builds up to 80% [2]. The main goal of a recommendation system is to satisfy users by suggesting contents they will enjoy. However, the user satisfaction (or, quality of experience – QoE) depends on a multitude of factors. This makes the design of content recommendation systems challenging, attracting significant research interest to account for the multiple, sometimes conflicting, factors that affect QoE [3], [4], [5], [6], [7], [8].

The relevance of a recommended item to a user (or, quality of recommendation - QoR) is one among the multiple factors that impact QoE. The quality in which the content is delivered

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

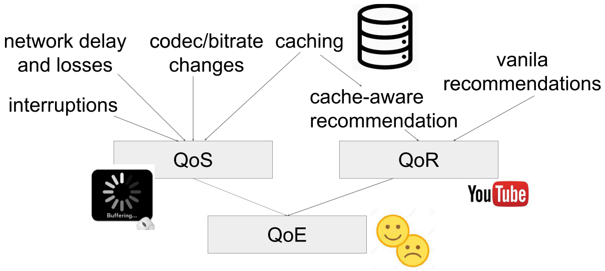


FIGURE 1. Quality of service (QoS) and quality of recommendation (QoR) impact quality of experience (QoE). Our experiments aim at figuring their interplay, with implications to similarity caching, service provisioning and content recommenders.

to the users (or, quality of service - QoS) also plays an important role in QoE. In particular, for video services, which account for the majority of Internet traffic [9], QoS impairments (e.g., buffering, startup delay and low video quality) are key aspects affecting QoE. While recommenders such as those used by Netflix influence a significant portion of users' demands, caches such as those deployed by Akamai serve a vast amount of content to end users. Caches reduce the load on custodians, decrease the latency for users, and benefit the network infrastructure by reducing traffic over bottlenecks.

Given the benefits of caching, significant effort has been invested in improving cache performance. In particular, similarity caching [10], [11], [12] and cost-aware caching, including QoS-aware [13], [14] and utility-driven caching [15], are some of the various recent developments in this domain [16], [17], [18]. Such advances, in turn, suggest novel opportunities but also pose new challenges in the realm of content distribution.

The basic idea behind similarity caching and cost-aware caching consists of determining both the similarity between contents and the cost to serve and/or retrieve a content, and then making decisions about which content to store and/or serve based on such assessments. Clearly, the multiple dimensions involved in the problem are intertwined, and the optimal decisions are non trivial. In particular, a user consuming a content not stored in a local cache may experience low QoS, and may prefer to rely on a content recommender to find a title that suits its expectations in terms of both content and QoS, motivating our main research question: *Can a recommender compensate for low QoS?*

Quality of Service (QoS). QoS metrics include network delay and losses, as well as application layer metrics such as video bitrate. Some of these elements are intertwined. For instance, the bitrate may change to accommodate a high loss ratio. QoS metrics are usually governed by Service Level Agreements (SLAs) and are one of the major focuses of network administrators, who monitor QoS to eventually perform a system upgrade or reconfiguration. For the purposes of this study, caching is a key element impacting QoS. As mentioned above, by bringing content closer to users, caches can reduce delay and losses.

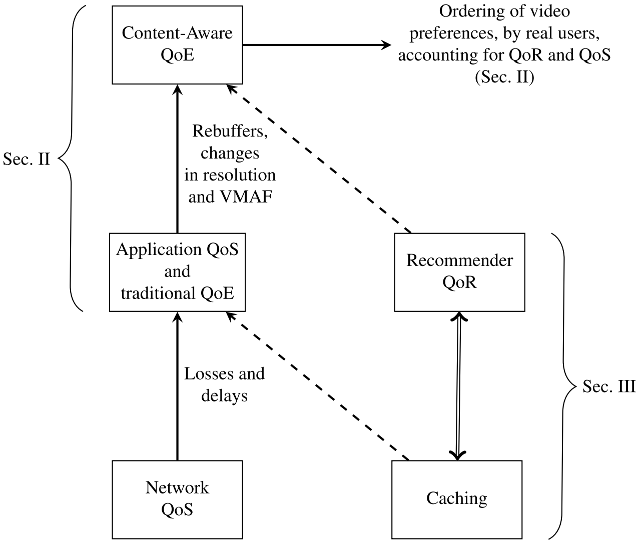

Content-Aware QoE
Ordering of video preferences, by real users, accounting for QoR and QoS (Sec. II)
Sec. II
Rebuffers, changes in resolution and VMAF
Application QoS and traditional QoE
Recommender QoR
Losses and delays
Sec. III
Network QoS
Caching

FIGURE 2. Relation between network, QoS, QoR and QoE.

Quality of Recommendation (QoR). QoR refers to the quality of recommendations issued by an automated recommendation system. Users provide feedback about recommendations explicitly, e.g., by indicating the number of stars deserved by a certain video, or implicitly, e.g., by abandoning a video session before it ends.

Quality of Experience (QoE). QoE is a broad concept that encompasses any aspect that may impact the user experience. It is commonly defined as the QoS perceived by users. In this work, QoE encompasses all the QoS and QoR aspects discussed above (see Figure 1). It may also encompass additional contextual elements, such as the time of the day and the device used by the user to watch a video.

In Figure 2 we further relate the QoS, QoR and QoE. To accommodate the fact that, in previous works, QoE did not account for content recommendations, in that figure we distinguish between application-level metrics, such as rebuffers and changes in resolution, and content-aware QoE, which accounts for recommenders. As mentioned above, given our focus on the relationship between network, caching and recommenders, in the remainder of this work we refer to content-aware QoE simply as QoE.

A. GOALS

Our work is driven by two main questions:

- When should recommenders account for QoS?
- Can recommenders compensate for low QoS?

We provide the following (partial) answer to our first question: recommenders should account for QoS primarily when the difference between interest in two videos (measured by the number of stars to the low QoS video minus the number of stars to high QoS video) varies between 0 and 2. As shown in Section II, in those cases, users may change their preferences due to QoS-impairments. When the difference between interests is large, users typically stick to their choices. When the difference between interests is negative, QoS does not play a significant role. Considering the scenarios wherein recommenders should account for QoS, the second question relates to the extent to which it is feasible to compensate for low QoS: Can recommenders compensate for low QoS? We indicate that most contents have at least one other content that is close to them, with respect to distances in the recommender graph, and that can be served with a higher QoS, e.g., because it is cached. Such contents are natural candidates to replace each other, leveraging the recommender graph, allowing recommenders to compensate for low QoS (Section III).

B. INSIGHT

The design of recommendation systems, whose ultimate goal is to deliver a high QoE, improves if both QoR and QoS are considered by the recommendation algorithms. In fact, QoS-aware recommendation systems have multiple benefits: (*a*) end-users will enjoy a higher QoE, (*b*) content providers will gain from the increased retention rate of their (more satisfied) customers, and (*c*) network administrators can optimize their network design (e.g., through novel caching policies), as shown in some recent studies [18]. Moreover, current architectural trends make joint QoR and QoS recommendations feasible. This is evident in the practices of content providers such as Netflix, Amazon, and YouTube, which currently deploy and/or control caching devices in content distribution networks (e.g., Netflix OpenConnect, Amazon CloudFront, Google Global Cache and Akamai CDN). Thus, content providers have knowledge of the QoS according to which content can be delivered.

C. GAP IN PRIOR ART

While there is vast literature on recommendation systems based on content/user characteristics (QoR) [1], [2], and on the interplay between QoS and QoE [19], to the best of our knowledge the relationship between QoS, QoR and QoE has not been previously investigated (see Figure 1). To build better recommendation systems and content delivery mechanisms, we must understand this interplay. For example: To what extent do QoS impairments affect users' choices with respect to the content they want to consume? How does the content category impact the sensitivity of users with respect to QoS and QoR? Which of the two contents, A and B, with $\mathrm{QoR}_\mathrm{A} > \mathrm{QoR}_\mathrm{B}$ and $\mathrm{QoS}_\mathrm{A} < \mathrm{QoS}_\mathrm{B}$, should be preferred by the recommendation system? Which will lead to a higher user QoE? Currently, we lack answers to these questions.

D. CHALLENGES AND CONTRIBUTIONS

The most pressing challenge in understanding how QoS and QoR impact QoE is conducting experiments with real users. Even though there are some previous works that considered the interplay between caching and recommendation systems [16], [17], [18], [20], [21], [22], none of these works accounted for a reality check against experiments with real users. Running such experiments involves motivating users to participate, designing a robust experimental setup and adjusting the experiments according to preliminary input. In this study, we addressed these challenges by devising controlled online experiments. To the best of our knowledge, we provide the first dataset collected from real users about preferences for video pairs, accounting for QoS impairments.[1] Motivated by this gap, we make the following contributions (see also Table 1):

1) **Experiments with real users.** We built an experimental web platform and collected data from real users regarding their interests and sensitivity to QoS impairments while watching a set of video pairs (one with low QoS and supposedly high QoR, and the other with high QoS and supposedly low QoR). For each pair of videos presented, we collected overall feedback about preferences, taking into account the nature of the content and QoS, as well as feedback about each of the two dimensions of interest: QoS and QoR.
2) **Investigation of the relationship between QoR and QoS.** We report insights on the relationship between recommendation and QoS. The *interest* in a video is characterized by the number of stars the user assigns to the video content, as is typically done on platforms such as Netflix. We then defined the *difference of interests* as interest for video A, served with low QoS, minus interest for video B, served with high QoS. According to our findings, if the difference of interests, measured on a scale ranging from -5 to +5, is between 0 and 2, a significant fraction of users prefer to watch the high QoS video even though they would have preferred the content of the low QoS video, i.e., *if the difference of interests is in the range [0,2], QoS affects users choices with respect to the content they wish to consume.*
3) **Design of QoS-aware recommendation system, given the QoS-QoR-QoE interplay thresholds found.** We conducted an initial data-driven study on the design of future QoS-aware content recommendation systems. The proposed logistic regression based classifier, to classify videos according to the user's preferred choices, accounting for QoS and QoR, reached an accuracy of 77.6%.
4) **Characterization: far from the trends, far from the user.** We quantified the extent to which unpopular content tends to be served with a lower QoS. In particular, we establish a methodology to determine the relationship between content popularity and its physical proximity to users by combining sampling of the recommendation graph and traceroutes in the physical network. The proposed method allows us to determine how popular a content has to be to be closer to the user, and it is instrumental for tuning recommenders.

[1] Our results can be reproduced and extended by the scientific community. URLs, scripts and *datasets* are available at https://tinyurl.com/qosqoeqor

TABLE 1. Overall set of questions and answers addressed in the study.

Question	QoS	QoR	Answer
When should recommenders compensate for low QoS? (Section II)	Controlled through impairments (rebufferings and bitrate changes)	Obtained from explicit feedback from users (number of stars towards each content)	When difference between QoR across alternatives is small (e.g., less than or equal to two, see Figure 4) but the difference in QoS is high (e.g., due to impairments)
Can recommenders compensate for low QoS? (Section III)	Obtained from active measurements in the network, issued towards content servers (delay and number of hops towards servers)	Obtained from recommender graph (path length in recommender graph)	Yes, in some cases being sufficient to simply reorder the contents presented to users (see Figure 12).

5) **Compensation: A bit trendier, much closer.** Favoring slightly trendier content while issuing recommendations (i.e., allowing a *content distance* between the requested content and the served content) can significantly increase the proximity of contents to users (i.e., decreasing *network distance*), positively impacting QoS. In particular, our results suggest conditions under which a recommender can compensate for low QoS, at zero costs for the network admin.

As indicated in Table 1, it is worth noting that the notions of content distance in the recommender graph and content distance in the network are instantiated in a similar but slightly different fashion when conducting the experiments to answer our two main questions. The content distance from the recommender perspective was measured using the difference between the number of stars towards each content while answering our first main question, and through the distance in the recommender graph while answering our second main question. Content distance in the network is characterized by network impairments while answering the first question, and by network delay and number of hops towards servers while answering the second question. Our key takeaways are summarized as follows: recommenders should account for QoS for contents that are close to each other (as measured by the difference between the number of stars issued towards the two) and they can compensate for low QoS (due to large distances in the network, e.g., as measured by network delay) by being replaced by alternative contents (whose distance is small in the recommender graph).

Figure 2 illustrates the connections between our research questions outlined in Table 1. In Section II, our primary focus centers on the concepts of application QoS at the level of visual stimuli, metrics pertaining to traditional QoE, and content-aware QoE, as shown in the upper part of Figure 2. Conversely, in Section III, our emphasis shifts towards network parameters and recommenders, as indicated in the lower part of Figure 2, which refers to the relationship between network delays, caching and recommenders.

Overall, our work bridges network-level QoS, application-level QoS, including rebufferings and resolution changes, and content-aware QoE, through the unified pipeline shown in Figure 2. Prior studies [23], [24], [25], [26] explored part of this pipeline, focusing on how network parameters can be leveraged to predict QoS disruptions at the application level, and how the latter can be used to produce application-level QoE metrics, such as the Video Multimethod Assessment Fusion (VMAF) score. As elaborated in Appendix A, we used the VMAF score to directly gauge the impact of video impairments on QoE (Section II) and to indirectly determine the influence of network conditions on QoE (Section III).

E. PAPER OUTLINE

Section II tackles contributions 1, 2 and 3, answering the question 'when should recommenders account for QoS?' while Section III tackles contributions 4 and 5, answering the question 'can recommenders compensate for low QoS?' Section IV presents related work, and Section V concludes the study.

II. WHEN SHOULD RECOMMENDERS ACCOUNT FOR QoS?

In this section, we describe and report the results of our experiment to quantify the sensitivity of users with respect to QoR and QoS. We developed an online platform and invited users to visit it and participate in our experiments. The data collection phase lasted 10 months and the dataset comprises 1,002 entries and 376 participants. Each user provided feedback for, typically, 3 pairs of videos.

Overview: We conducted experiments with real users to assess their preferences with respect to video pairs. Each user was exposed to a series of video pairs. Each pair of videos was selected such that one of the videos was hypothesized to be more interesting than the other (hypothesis checked in Section II-D). The video that was assumed to be more interesting, however, was served with a lower QoS. To this end, we purposely added QoS impairments to that video. The disturbance types and frequencies were inspired by [19], and consisted of rebufferings and bitrate changes (for details on the disturbance frequencies and durations, see Appendix A). Then, we experimentally validated our hypothesis with respect to users' interests and empirically discovered the thresholds governing the tolerance of users with respect to QoS impairments while choosing which video to watch. In summary, the two products of our methodology are (*i*) the collected dataset and (*ii*) a systematic strategy for quantitatively answering the questions posed in the proposed research agenda.

A. DESCRIPTION OF THE EXPERIMENT

The user watches 3 pairs of videos; the 3 pairs presented to each user are randomly selected from a set of 7 video pairs corresponding to different categories. Although the categories of both videos in the pair are the same (e.g., sports or music), the videos were chosen so that one of them is, on average, more interesting than the other. In addition, the supposedly more interesting video has a low quality of service, whereas the less interesting video has high QoS. Our goal is to analyze how much the user is willing to compromise the quality of service in order to watch a more interesting content, i.e., to quantify the tradeoff between quality of service and quality of the content (for details on QoS impairments and the demographics of users see Appendices A and B).

1) VIDEO WITH LOW QoS (VIDEO A)

Despite having a lower QoS, it has content that is more interesting for most users. For instance, in the sports category, videos of extreme sports generally have more views than those of less radical sports do.

2) VIDEO WITH HIGH QoS (VIDEO B)

Has no QoS impairments at all, but the content is generally considered less interesting. For instance, in the category of comedy, a classical joke is usually not considered as interesting as a brand new one.

All selected videos lasted, roughly, 30 seconds so as to make the experiment more attractive to a broader audience.

3) ETHICS AND PRIVACY

The participants were volunteers invited personally or through mailing lists to participate in our measurement campaign. The experiment lasts for approximately 8 minutes, and participants were free to leave the experiment at any point. In addition, participants were allowed to choose whether they wanted to identify themselves (through their email). We will share our results with those that identified themselves.

4) TECHNICAL CONSIDERATIONS

Our experiments were conducted on the Internet. Participants accessed the experiment through a website, in which they were guided over video pairs and corresponding questionnaires. Note that the QoS impairments were added to the videos in an offline fashion, prior to those videos being presented to the users. The instability of the Internet connectivity of users could, in principle, cause additional QoS impairments. To avoid this, we developed a software module that downloaded the entire video files in a given session to a local cache in the participant's machine before the experiment started. By doing so, we were able to control the QoS impairments to which each user was exposed. A detailed list of the QoS impairments added to each video is available together with our datasets. For the purposes of this study, the impact of those impairments was fully captured through the replies provided by users regarding how those impairments impacted their QoS.

B. TERMINOLOGY AND FEATURES

Next, we introduce basic terminology.

1) VIDEO PAIRS

There were seven video pairs in our experiment. Each pair comprised two videos of the same category/topic.

2) SESSION

Users typically watch three pairs of videos, and rate each pair producing three samples that are recorded in our dataset.

3) SAMPLE

A record of the answers of a user rating a pair of videos, i.e., interests values for the two videos in the pair along with an annoyance value for the low-QoS video, the preferred video and a textual field justifying the choice for the preferred video.

4) CLASSIFIER

A recommendation system encompassing decision rules to classify samples according to our target variable, namely, which video in a given pair a user prefers to consume.

5) CLASSIFIER PERFORMANCE METRICS

Metrics used to assess and compare the performance of different classifiers, e.g., F1 score or accuracy [27].

6) ACCURACY

One of our key performance metrics, which is equal to the fraction of correctly classified samples in our dataset. Accuracy assessment involves a resampling procedure, such as cross-validation.

Next, we list the features collected in our experiment.

After watching a video pair, the user was asked to complete a form with the following fields:

7) INTEREST IN LOW QoS VIDEO (IntLowQoS)

Interest in the content of the low QoS video (integer values from 0 to 5).

8) INTEREST IN HIGH QoS VIDEO (IntHighQoS)

Interest in the content of the high QoS video (integer values from 0 to 5).

9) ANNOYANCE

A measure of the user's discomfort with respect to interruptions in one of the videos. Annoyance reflects the user's sensitivity to video quality glitches, in the video that contained disturbances (integer values from 0 to 5).

TABLE 2. Statistics on video pair preferences, showing the 95% confidence intervals of interests expressed over videos and annoyance due to QoS impairments.

Description	Number of Observations	Prefer High QoS	Interest Low QoS		Interest High QoS		Hypothesis test (p-value)	Annoyance	
			Mean	Standard deviation	Mean	Standard deviation		Mean	Standard deviation
All	1002	561	3.31 ± 0.09	1.42	2.77 ± 0.09	1.51	$< 10e{-}173$	3.56 ± 0.08	1.33
1) Soccer	132	75	3.27 ± 0.28	1.64	2.50 ± 0.28	1.66	2.65e-122	3.83 ± 0.20	1.19
2) Comedy	118	78	3.26 ± 0.24	1.34	2.97 ± 0.27	1.50	4.29e-043	3.52 ± 0.25	1.42
3) Pets	135	64	2.93 ± 0.25	1.52	2.13 ± 0.25	1.51	4.72e-139	3.32 ± 0.24	1.42
4) Sports	158	84	3.16 ± 0.21	1.35	2.49 ± 0.21	1.35	1.51e-164	3.53 ± 0.19	1.24
5) Animals	135	95	3.13 ± 0.24	1.38	3.08 ± 0.25	1.49	5.00e-004	3.58 ± 0.23	1.39
6) Science	141	55	3.94 ± 0.21	1.25	3.05 ± 0.23	1.43	4.06e-173	3.33 ± 0.22	1.33
7) Music	163	98	3.56 ± 0.19	1.22	3.18 ± 0.21	1.40	4.18e-107	3.80 ± 0.20	1.30

10) PREFERENCE

Preference for one of the two videos. It takes a binary value, either lowQoS or highQoS. This is the target variable. It indicates which video a user preferred, accounting for interests and sensitivity to QoS impairments.

11) RATIONALE AND COMMENTS

After each session, the user is invited to explain, in text, why he/she has chosen a video. The user can also add additional information regarding the experiment.

We derived an additional feature based on sentiment analysis of the provided text comments: polarity score.

12) POLARITY SCORE (SENTIMENT)

A continuous metric generated by the VADER sentiment analysis tool [28] executed on top of the textual feedback (rationale and comments), taking values from -1 to $+1$ (details deferred to Section II-F).

Among our findings, we discovered that, together with other features, the polarity score significantly improved classification accuracy (Section II-F).

C. EXPERIMENTAL RESULTS

In this section we report basic experimental results. Our goals are to (*i*) test whether our assumption about the videos hypothesized as being less interesting, holds in the considered population; (*ii*) verify whether users were able to distinguish between QoS and the nature of the content and (*iii*) assess the role of QoS in user choices.

We begin by considering goal (*i*), and compute the means and confidence intervals of users' interests, corresponding to the low and high QoS videos. Our aim is to check whether our assumption regarding videos with high QoS, hypothesized as being less interesting, holds in practice. Let $I^{(\mathrm{L})}$ and $I^{(\mathrm{H})}$ be the interests in low and high QoS videos, respectively. Let $\overline{I}^{(\mathrm{L})}$ and $\overline{I}^{(\mathrm{H})}$ be the corresponding sample means.

From our experiments, we learned that $\overline{I}^{(\mathrm{L})} = 3.31$ and $\overline{I}^{(\mathrm{H})} = 2.77$. In addition, the 95% confidence intervals are given by

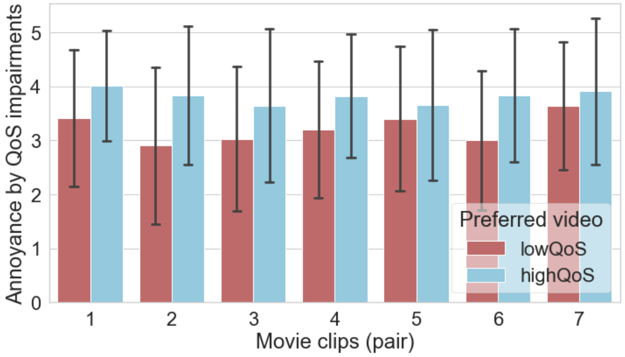


FIGURE 3. Annoyance per video pair.

$$I^{(\mathrm{L})} \in (3.22, 3.39), \quad I^{(\mathrm{H})} \in (2.67, 2.86). \tag{1}$$

Clearly, the interest expressed over low QoS videos is greater than that over high QoS videos, and the confidence intervals do not overlap, suggesting that the difference is statistically significant. To further validate this point, we executed a one-sided statistical hypothesis test, in which the null hypothesis H_0 corresponds to the mean interest over low QoS videos being smaller than or equal to the mean interest over high QoS videos, and the alternative hypothesis H_1 corresponds to the mean interest over low QoS videos being greater than the mean interest over high QoS videos. We were able to reject the null hypothesis, with a p-value less than 0.05, indicating that the positive difference between interests over low QoS and high QoS videos is statistically significant. The above conclusions still hold for all the considered categories, with a notable exception being the category of animals, where the confidence intervals overlap. However, the statistical one-sided hypothesis test is still able to reject the null hypothesis (see column p-value in Table 2).

Table 2 reports the different video pair categories, together with the number of samples collected for each category. It also indicates the number of users that preferred each of

TABLE 3. Correlation matrix.

	IntLowQoS	IntHighQoS	Annoyance	Sentiment
IntLowQoS	1.000	0.393	0.084	0.043
IntHighQoS	0.393	1.000	0.192	-0.019
Annoyance	0.084	0.192	1.000	-0.160
Sentiment	0.043	-0.019	-0.160	1.000

the two videos in each session (those results are discussed in Section II-E). Table 2 shows that the samples are uniformly distributed across categories, and confirms the above observations regarding the difference between interests across categories, as further discussed next.

D. HYPOTHESIS AND CONSISTENCY CHECKS

1) HYPOTHESIS CHECK

Table 2 reports the mean and standard deviations of Annoyance and interest levels in low and high QoS videos (IntLowQoS and IntHighQoS). It also reports the 95% confidence intervals for the two latter metrics. Clearly, the average interest for the low QoS video is consistently higher than that for the high QoS video (thus validating our hypothesis). The greatest difference occurred for the science category (category 6) and the smallest for documentaries on animals (category 5). As discussed above, the latter category is the single category for which there is an overlap between the confidence intervals corresponding to interests in low and high QoS videos.

Figure 3 shows how Annoyance varied for different video pairs, with error bars corresponding to standard deviations. Figure 3 together with Table 2 indicate that users who preferred the high QoS video typically were more sensitive to QoS impairments, implying that QoS played a fundamental role in their preference for the high QoS video. The results reported in Figure 3 and Table 2 serve to confirm that users in general faced a tradeoff between QoR and QoS while choosing their preferred video.

2) CONSISTENCY CHECK

Table 3 presents the correlation matrix for a selected set of features. The matrix indicates that most participants knew how to answer the questionnaires properly. This conclusion is supported by the low correlation observed between Annoyance and interest in the low QoS video (IntLowQoS). It suggests that participants were capable of isolating the influence of QoS when assessing their interest for contents, and that they were able to jointly account for QoS and QoR when determining which videos they preferred to consume.

For each pair of features, we performed a hypothesis test in which the null hypothesis corresponded to no correlation between the features. In Table 3, we highlight the entries for which the p-value is less than 0.0001, indicating that the correlation is statistically significant. The three pairs with correlated features are: (*i*) IntLowQoS and IntHighQoS, (*ii*) Annoyance and IntHighQoS, and (*iii*) Annoyance and Sentiment. The first pair, in particular, confirms our expectations regarding the behavior of users, providing further evidence that the experiment is inherently consistent.

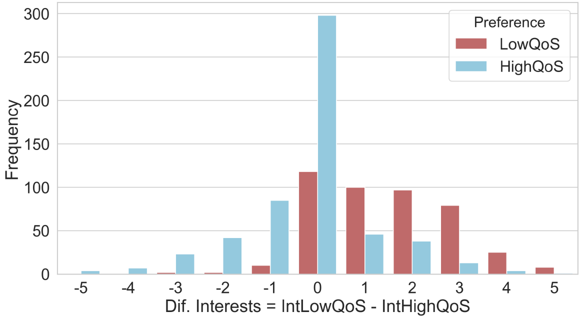


FIGURE 4. Video preferences as a function of interests and QoS: when the difference between interests is smaller than 2, users trade between QoS and QoR and may prefer a less interesting video with higher QoS.

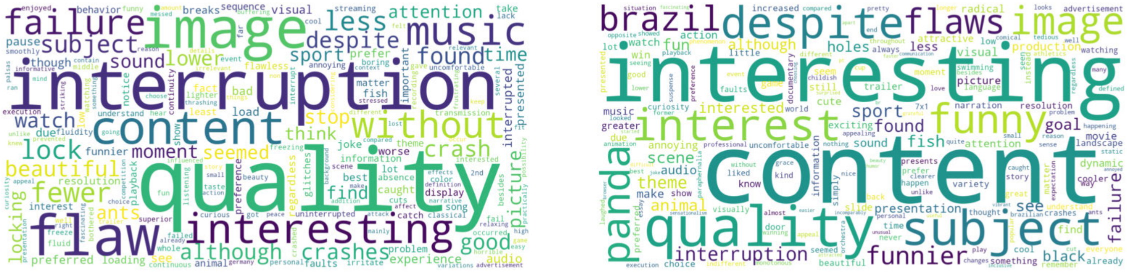
E. QoS VS. QoR: WHAT (AND WHEN) IS IMPORTANT?

Figure 4 summarizes the role played by QoS and the nature of the content in user choices. The x-axis indicates the difference between interests towards low and high QoS videos, IntLowQoS - IntHighQoS (ranging from -5 to 5). Each bar indicates the number of samples in which users preferred each of the two videos in each video pair. As a sanity check, when users prefer the content of the high QoS video (i.e., negative values in the x-axis), it is almost always the case that they choose that video as their preferred one (90% of cases).

Note that some users found that, in certain video pairs, we reduced the quality of the video that was deemed less interesting for them. Hence, it is not always the high QoR video whose QoS is reduced. In the particular instances of video pairs wherein low QoR videos had their QoS reduced, users did not encounter a dilemma in their selections. In such cases, the majority of users consistently chose the video with the higher QoS and QoR as their preferred option, although a few exceptions are discussed in Appendix C.

1) WHEN THE DIFFERENCE BETWEEN INTERESTS IS SMALL OR ZERO, QoS IS IMPORTANT

In particular, when the difference between interests is 0, the high QoS video is chosen in 71.9% of cases. The remaining 28.1% of the answers can be explained by the granularity that the users could declare their interests, e.g., a user slightly more interested in the low QoS video might have equally rated her interest in both videos. In situations where the difference between interests is 1 or 2 (on the x-axis), the low QoS video is still the most preferred choice. However, for a significant fraction of samples (30%) users chose the high QoS video, even when it did not correspond to their favorite contents. Those scenarios clearly involved a tradeoff, wherein users favored QoS over QoR. One of our aims in the remainder of this paper is to characterize and



(a) Prefer high QoS video

(b) Prefer low QoS video

FIGURE 5. Word clouds for comments by users that prefer (a) high and (b) low QoS videos. The former focus on interruptions, and the latter on content.

automatically identify those scenarios to improve the design of recommendation systems.

2) THE ROLE OF QoS IS CATEGORY DEPENDENT

Returning to Table 2, among the 1002 samples, in 561 (i.e., 55.99%) users preferred the high QoS video. As previously pointed out, the average interest towards low QoS videos was higher than that towards high QoS across all the 7 video pairs considered (see Table 2). However, in 5 out of the 7 video pairs (namely, football, comedy, extreme sports, animals documentary and music) most users preferred the high QoS video clips (Table 2) despite expressing more interest towards the low QoS video. The other 2 video pairs (pets and scientific documentary), in which the majority preferred the video with interruptions, were precisely those with the largest difference of interests between the videos of the pair. In these situations, the average difference between interests was greater than 0.8 stars. Additionally, these two video pairs were the ones where users reported the lowest absolute values of Annoyance, suggesting that when watching videos about pets and science users are concomitantly (*i*) more determined with respect to their interests and (*ii*) less sensitive to QoS impairments.

F. TEXT MINING AND SENTIMENT ANALYSIS

By analyzing the textual information provided by users, we can gain insights into how they formulate their decision-making process, with respect to their overall satisfaction with a video. One approach to analyzing the text is to count the occurrences of specific words or terms to determine what the users deem most important. To this end, Figures 5(a) and 5(b) present word clouds for comments made by users who preferred high and low QoS videos, respectively. In the word cloud representing the high QoS choice (Figure 5(a)) there is a much more significant presence of words related to QoS such as: quality, image, interruption, failure and flaw. It is also important to highlight the presence of the preposition ‘‘without’’ which is almost always associated with the word ‘‘failure’’ to indicate ‘‘without failure’’.

In the word cloud referring to instances where the low QoS video was preferred (Figure 5(b)), ‘‘interesting’’ and ‘‘content’’ were the most prevalent words. Words related to QoS, such as ‘‘quality’’, are still present, but the presence of the word ‘‘despite’’ is noteworthy as it generally denotes contrast. In many instances, users explained that they chose the low QoS video *despite* its low quality. In summary, word clouds are instrumental to understand the participants video-choice process, and corroborate the fact that QoS and QoR play a fundamental role in determining QoE.

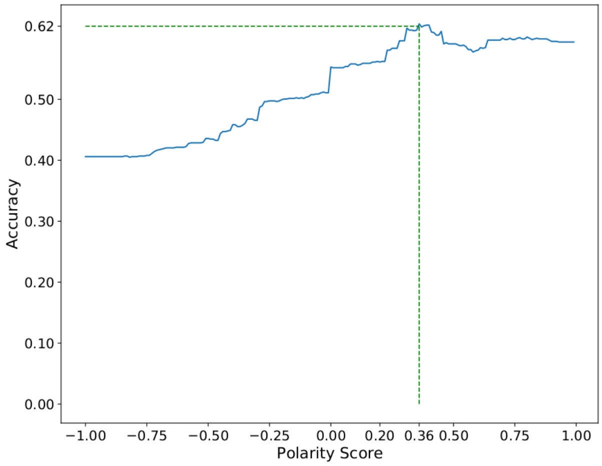


FIGURE 6. Accuracy of classification of users that prefer high QoS and less interesting videos as opposed to low QoS and more interesting videos, as a function of polarity score threshold.

Next, we explore the relationship between sentiment analysis and video recommendations. In particular, we focus on one of our experimental findings that relates (*a*) sentiments expressed by users in their remarks about the experiments, with (*b*) their choice of preferred videos.

Using off-the-shelf sentiment analysis algorithms, we found a correlation between the sentiments expressed by users and their choice of preferred videos. In particular, we found that users with more positive sentiments tended to be more inclined towards preferring videos with lower QoS (they are more resistant and lenient towards QoS impairments). In contrast, users who choose the high QoS video, usually complain about the lack of quality in the low QoS one. Thus their explanations contained more negative words.

Valence Aware Dictionary and Sentiment Reasoner (VADER) provides a lexicon and a rule-based sentiment analysis tool. Given a textual input, VADER returns a polarity score, which measures how positive or negative is a given piece of text [28]. The polarity score ranges between -1 and 1. We empirically found the optimal threshold of 0.36 to classify our data. Using this threshold, we establish the following rule: the estimated preference is categorized towards the "high QoS" video if the polarity score is less than or equal to a threshold of 0.36, and towards the "low QoS" video otherwise.

We empirically observed that negative comments refer to complaints about QoS that lead to the choice of high QoS videos despite their less interesting content. Figure 6 shows the accuracy of the classification of users who prefer high QoS and less interesting videos as opposed to low QoS and more interesting videos, as a function of polarity score threshold T. If the polarity score of a post is below T, we classify the post as pertaining to a user who favors QoS. Otherwise, the post is classified as favoring QoR. Indeed, an interesting finding is that using the polarity score metric and the above simple rule, with $T = 0.36$, namely, that if polarity is less than or equal to 0.36, the estimated preference is high QoS. We achieve an accuracy of 62.3% (see Figure 6) and a precision for low and high QoS videos of 59.1% and 64.1%, respectively. Similar results were obtained while training a decision tree to distinguish between the two classes, with the entropy of the classes as the criterion to set the threshold. This is a promising result for the ability to predict the expected preference of users, with respect to QoS and QoR, e.g., from past users' comments; we plan to further investigate this relationship between natural language processing, users moods, and QoS-sensitivity as part of future research.

G. MODELING USER CHOICES

Next, our goal is to derive decision rules for predicting users' preferences. For that purpose, we consider supervised learning approaches [29]. First, we rely on a classical statistical model, namely, Logistic Regression. Linear Discriminant, Quadratic Discriminant, and K-nearest neighbors are discussed in Appendix C. In Appendix C we also propose a heuristic to remove outliers and proceed with machine learning approaches and a ten fold cross validation. While the first class of models is simpler and amenable to interpretation, the latter leads to higher accuracy. In Appendix D we also balance between interpretability and accuracy through decision trees.

Figure 7 shows a scatter plot of our dataset accounting for two features: DiffInterests and Annoyance. Recall that DiffInterests refers to the difference between interests expressed for videos served with low and high QoS, DiffInterests = IntLowQoS - IntHighQoS. All the features considered in our model have a discrete nature. Therefore, for a clearer visualization with scatter plots, we added a low variance white noise to each point coordinate. We empirically found that adding independent random Gaussian noise with zero mean and standard deviation of 0.2 suffices for our visualization purposes.

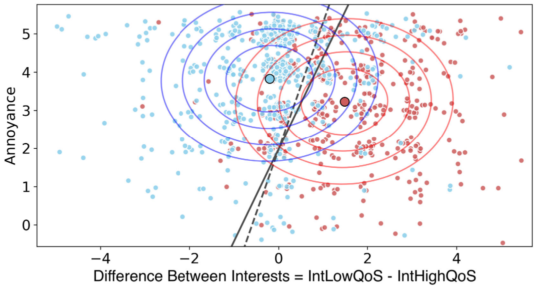


FIGURE 7. Scatter plot of Annoyance against DiffInterests.

In Figure 7, each point corresponds to a given sample. Points in blue and red correspond to samples in which users preferred the high QoS and low QoS videos, respectively. Figure 7 indicates that it is possible to split the plane in two regions. In the extreme right region (content of low QoS video strongly preferred against content of high QoS video, with DiffInterests ≥ 2) most users preferred to watch the low QoS video. When the difference between interests varies between 0 and 2, $0 \leq$ DiffInterests ≤ 2, even though the content of the low QoS video is preferred on average, a significant fraction of users experience higher QoE while watching the high QoS video.

1) LOGISTIC REGRESSION

Motivated by Figure 7, we consider a simple logistic regression model with DiffInterests and Annoyance as candidate features. To formally assess the relevance of these two features, we considered a more complex model wherein all the features were present. We then performed the corresponding hypothesis tests with the null hypothesis being that DiffInterests and Annoyance do not play a significant role in the outcome variable. For both features, we were able to reject the null hypothesis, with p-values smaller than 10^{-8}. We also ran the same tests for the other features, and we were not able to reject the null hypothesis in those other cases. The fact that Annoyance and DiffInterests are both relevant features bodes well with Figure 7, which indicates that these two features indeed play important roles in the classification of the samples.

The dataset was randomly divided into training (752 observations) and testing (250 observations). Using only 75 percent of the data to fit the model, the estimated coefficients of the line separating the two regions were quite close to those obtained when using the complete data set (dotted line in Figure 7). The accuracy of the test set was comparable to that obtained using the resubstitution method (i.e., using the whole data set for both training and testing). In addition, when considering only the training set both features were still considered significant, with p-values less than 10^{-6} for both DiffInterests and Annoyance. The accuracy, precision, recall and F1-score equal 0.776, 0.7727, 0.7748 and 0.7716, respectively. We also experimented with LDA (solid line in Figure 7), producing similar results of 0.768, 0.7625, 0.7619 and 0.7632, respectively.

III. CAN RECOMMENDERS COMPENSATE FOR QoS?

In this section, we report results on Internet measurements indicating the feasibility of characterizing the QoS at which different items can be served. Given such characterization, and information about the content recommendation graph, we present findings to support conditions under which we have an affirmative answer to our main question, namely: Can recommenders compensate for low QoS?

The proposed methodology involves both network and recommender measurements. Network measurements are used to collect information regarding delays incurred to consume different videos. We refer to the approach used to assess those delays as content-aware pings and traceroutes, as it involves first identifying the host that will serve the desired content, and then issuing a `ping` and a `traceroute` towards that host to assess the corresponding delay. Then, we combine information collected from such measurements together with measurements of the recommendation graph. By sampling the YouTube recommendation graph we learn which contents are close to each other. Then, we bridge the gap between *closeness* with respect to the nature of the content (*content distance*) and with respect to host proximity (*network distance*) to draw our main findings. Our key contributions are summarized as follows.

A. CHARACTERIZATION: FAR FROM THE TRENDS, FAR FROM THE USER

We quantified the extent to which unpopular content tends to be served with a lower QoS. In particular, we establish a methodology to determine the relationship between content popularity and its physical proximity to users by combining sampling of the recommendation graph and traceroutes in the physical network. The proposed method allows us to determine how popular a content has to be to be closer to the user, and it is instrumental for tuning recommenders.

B. COMPENSATION: A BIT TRENDIER, MUCH CLOSER

Favoring slightly trendier content while issuing recommendations (i.e., allowing a *content distance* between the requested content and the served content) can significantly increase the proximity of contents to users (i.e., decreasing *network distance*), positively impacting QoS. In particular, our results suggest conditions under which a recommender can compensate for low QoS, at zero costs for the network admin.

Next, we introduce the measurement methodology. Then, Sections III-F and III-H report a characterization of delays towards contents and the extent to which recommenders can compensate for those delays, respectively.

C. METHODOLOGY

1) GOALS

Popular and trending videos are known to be cached close to users. In YouTube, in particular, a list of trending contents is presented to users on their home page. Beyond such remarkably trending contents, which other contents are cached closer to users? How do different features, such as the number of views, impact closeness? And how do users profiles, e.g., reflected through their vantage points, impact delays towards different contents?

TABLE 4. Datasets descriptors.

Country	Videos (samples)	Cache Hit Rate	Unique URLs
BR	872	38%	167
CA	969	44%	211
FR	3360	71%	202
HK	1758	86%	203
IN	617	78%	162
US	1859	69%	281

2) OVERVIEW

We used YouTube API to access the recommendation system and generate recommendation graphs for each trending content by performing a Breadth-First Search (BFS) through the network of recommendations [22]. Then, we emulate a request towards each of the videos, and determine the corresponding media server URLs.

Let $\mathcal{T}$ be the set of trending videos considered as our seeds for the BFS. We let $|\mathcal{T}| = 50$, i.e., we consider the top-50 most popular videos in each of the considered regions, and start a BFS from each of those. The BFSs are executed up to a depth of five hops in the recommendation graph and account for the first three videos in each of the observed recommendation lists.

3) IS A VIDEO CACHED CLOSE TO THE REQUESTER?

We measured network level features, using ping and traceroute towards video servers, and grouped the videos into two clusters based on those metrics. Let C denote an indicator variable, equal to 1 if our measurements suggest that the video is cached close to our measurement vantage point, and 0 otherwise. Then, we computed the correlations between C and metrics related to the recommenders, that are introduced in the sequel.

4) DATASETS

A summary of the dataset descriptors is presented in Table 4. Our dataset collected from vantage points in Rio de Janeiro, for instance, has 872 video samples and 167 unique hosts serving those videos, out of which 38% are marked as low delay hosts. Content served by low delay hosts is assumed to be cached close to users, i.e., $C = 1$.

D. CONTENT DISTANCES BY RECOMMENDERS

We represent YouTube videos and their relations through a recommendation graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Thus, each video is represented by a node, and each recommendation is represented by an edge, where $(i, j) \in \mathcal{E}$ if j appears in the recommendation list of i.

We consider an abstraction of a surfer that navigates across the nodes of the recommendation graph. The surfer begins at the seed of the graph. The shortest path between each video and the seed is referred to as the video *depth*. The position of the video in its corresponding recommendation list is referred to as the video *width*. Note that whereas the depth is oblivious to the ordering at which videos appear in the recommendation lists, width is sensitive to such ordering.

1) RECOMMENDER MEASUREMENTS

Next, we consider the surfer sample path. Each time the surfer visits a node, it produces a sample. The sample comprises the identifier of the visited video and its view count, together with its width and depth. Such measures are collected passively, without interfering with the network state.

TABLE 5. Description of collected features per sample.

feature	description
Recommender and recommendation graph features	
I, video identifier	YouTube video unique identifier
V, view counts	number of views, since video creation
W, width	position of video recommendation list
D, depth	number of hops to seed of recommendation graph
QoS and network related features	
H, host	host serving video I
L, path length	length of path (number of hops) to H
P, ping values	vector of 10 delay observations collected with ping
$\mathbb{E}(P)$, mean delay	average of ping values
C, cached close?	equals 1 if $\mathbb{E}(P)$ is “small”, 0 otherwise

E. HOST DISTANCES IN THE NETWORK

We developed a software module to identify YouTube media servers associated with each video in the recommendation graph. Our module is similar to Wireshark web developer tools [30]. It is important to note that the YouTube hosts returned by our program are dependent on the network used when collecting data. In this study, we focus on data collected in Rio de Janeiro in an ISP's network. After identifying hosts associated with each sample, we performed traceroutes and recorded the corresponding delays and path lengths.

1) FEATURE SUMMARY

A sample is a tuple containing (a) video identifier; (b) view counts; (c) width; (d) depth; (e) host; (f) delay observations; (g) path length. Features are summarized in Table 5.

F. EXPERIMENTAL FINDINGS

Next, we report our experimental findings. Our experimental goal was to characterize the relationship between the recommendation graph and the QoS metrics.

1) BRIDGING RECOMMENDER AND NETWORK

Next, we report the correlations between recommender and network metrics, bridging the collected measurements.

Let V and A denote the video view counts and age, respectively, and let N be an indicator variable equal to 1 if the video is in the native language of the region wherein our vantage point was located, and 0 otherwise. We observed correlations between C and the above three variables of 0.18, 0.12 and 0.13, respectively. Such correlations indicate the extent to which more popular content is cached closer to our vantage points. In particular, the correlation between video language and caching suggests that the recommendation of videos in native languages may favor users' and higher QoS.

Let W and D denote the video width and depth, respectively.

We observed negative correlations of -0.19 and -0.23 between C and the above two metrics, respectively. These correlations measure the tendency for videos nearer to the recommender graph's root to also be closer to users within the cache network.

2) A FEW HOSTS SERVE MOST OF THE CONTENT

We found that different video links have a significant overlap in their URLs, referring to machines stored in the same datacenter. To illustrate this, we considered the hosts observed from our vantage points in Rio Janeiro. We observed four common prefixes when issuing requests to top-trending Brazilian contents from a vantage point in Rio de Janeiro: *b8u*, *8p8* and *bg0*. A fourth prefix, *q4f*, appeared in less than 5% of our requests, and was associated with high delays, on the order of 150 ms (six times higher compared to the others).

- *b8u* and *8p8* correspond to low delay values, as observed from pings issued from Rio de Janeiro. Delays are around 21 ms and the corresponding standard deviations equal 0.28 ms and 0.42 ms;
- *bg0* and *q4f* correspond to larger delays, around 28 ms and 150 ms respectively. The corresponding standard deviations were 11 ms and 0.45 ms.

In what follows, we refer to the aforementioned prefixes simply as *hosts*, noting that they may subsume multiple machines in the same domain. As hosts are distinguished into two groups based on delays, we refer to those hosts as high delay hosts and low delay hosts depending on the group they fit. The corresponding videos served by those hosts are marked as videos served by caches closer and farther away from users, with $C = 1$ and $C = 0$, respectively.

3) A BIRDS EYE VIEW ACROSS COUNTRIES

Next, we present a birds eye view of our measurements. We ran measurement campaigns across six countries. Each country corresponds to one vantage point, and to its own set of trendy contents.

Figure 8 shows the relationship between the distance in the recommender graph against standardized delays, obtained using the MinMaxScaler [31], so that standardized delay values fall in the range between 0 and 1. In all the considered countries, moving deeper into the recommender graph corresponds to higher delays. One of our aims in the remainder of this study is to further investigate and quantify this relationship, discussing its implications for the design of recommenders.

Figure 9, in turn, shows the path lengths for hosts serving contents in the six considered countries. It shows that hosts are always up to thirteen hops from our vantage points, and that the difference between the closest to the furthest host is of at least two hops.

Due to space limitations, in what follows we focus our detailed analysis on the Brazilian measurements, noting that all our results extend to the six considered countries.

4) ACCOUNTING FOR MULTIPLE VANTAGE POINTS

Next, our goal is to further quantify the extent to which unpopular content is impacted by slower-to-respond hosts. To this end, we considered top trending Brazilian and French videos. Then, we ping hosts storing those contents from different vantage points, leveraging the measurement infrastructure provided by SpeedChecker. We collected at least 10 measurements from each host, and report median values.

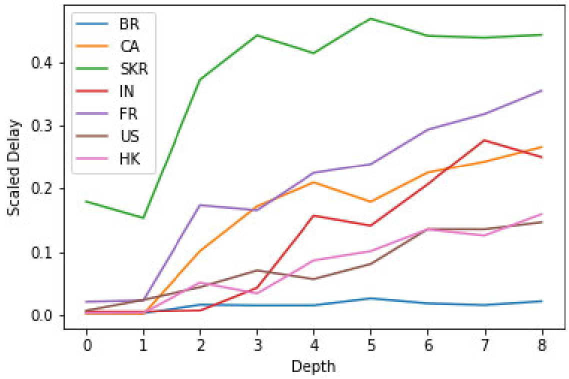


FIGURE 8. Scaled delays versus recommender depth. Scaled delays are computed using MinMaxScaler [31].

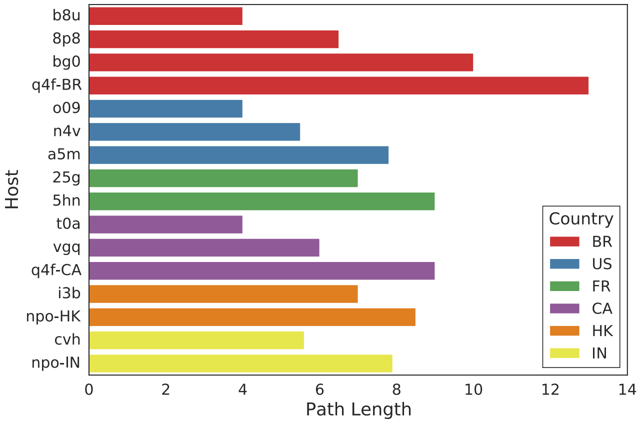


FIGURE 9. Path lengths towards different hosts.

Table 7 illustrates our results. To produce Table 7, we considered the Brazlian and French top-trending videos and ping their closest hosts, from vantage points in Brazil, Greece, USA and France. We grouped samples according to whether they were associated with slower-to-respond or fast-to-respond hosts, using the methodology described in the previous section.

Clearly, the lowest delays for the top trending contents are obtained in the regions where the contents are trending. As the difference in delays can be of an order of magnitude, those results illustrate that recommenders can have a significant impact on the transit costs for service providers.[2]

G. SAMPLING OF RECOMMENDATION GRAPHS

Next, we describe the two strategies considered in this study to sample the recommendation graph.

1) BREADTH FIRST SEARCH (BFS)

We set the seeds of our sampling strategy at the most popular videos recommended by YouTube in a given region, and exhaustively traverse the recommendations in a BFS fashion. Each popular video generates its own rooted tree, where the root (seed) of the tree corresponds to a popular video.

2) RANDOM WALK SAMPLING

We choose popular videos as seeds, and start random walks from those. Each random walk corresponds to a sample path in the recommendation graph, from the seed up to the bottom most level. After reaching the bottom level, another seed is selected from the top-50 most popular videos, and the process is repeated. By the end of the process, we obtain a collection of paths from the selected seeds (popular videos) to the bottom ones (unpopular videos). We do not repeat the same seed twice, with the aim of obtaining a representative set of videos in each level.

3) CHALLENGES IN SAMPLING OF RECOMMENDATION GRAPHS

Recommendation algorithms are strategic products, at the core of content providers such as YouTube and Netflix. Therefore, it is not surprising that the sampling of recommendation graphs by third parties is constrained and must be exercised under a sampling budget. Due to privacy concerns, even in large companies such as Google, the separate content recommendation and service provisioning teams may face challenges in the exchange of data.

In what follows, we address such challenges involved in the sampling of the YouTube recommendation graph, while inquiring the YouTube API. Given the quota on the number of API inquiries per time unit, we discovered a tradeoff between the BFS and random walk sampling approaches. The random

[2] It has been reported in related work that delays and transit costs negatively correlate with experienced QoS [30], [32].

walk approach produces a dataset that is more balanced with respect to the number of videos in each level. Nonetheless, the YouTube quota is reached earlier when compared against exhaustive BFS sampling. To further clarify that point, note that at every request for R related videos, the value of R does not impact the amount of quota spent, as quota are sensitive only to the number of requests but not to the size of the returned list. Parameter R is allowed to vary between 1 and 50 related videos. Therefore, while random walks generate only one sample for each request for related videos, as they randomly pick a video from the list, the exhaustive BFS approach leverages all the videos on the related lists, and allows the collection of larger datasets. Table 6 summarizes the advantages and disadvantages of each of the considered approaches. All results reported in the sequel were obtained using BFS, except for Figures 10(b) and 11, that were obtained using random walks.

H. EVALUATION

Next, we leverage the QoS characterization introduced in the previous section to establish conditions under which content recommennders can compensate for low QoS. First, we indicate that a decision tree can capture delays based on data collected from the recommendation graph. Then, we illustrate a simple mechanism for recommenders to leverage knowledge about delays to compensate for low QoS.

TABLE 6. Sampling approaches.

	BFS	random walk
number of samples (limited by YouTube quota)	more samples	less samples
number of samples per level	unbalanced (grows exponentially with level)	balanced (roughly constant per level)

TABLE 7. Median ping values (times in milliseconds).

Origin city (SpeedChecker vantage point)	Target host with top-trending contents			
	Brazilian contents		French contents	
	bg0 Sao Paulo	b8u Rio de Janeiro	fr1 5hn 4g5 Chartres I	fr2 25g Chartres II
Sao Paulo	17	21	223	223
Rio de Janeiro	21	16	219	219
Athens	295	302	94	94
New York	333	331	334	403
Chartres	223	219	17	8

1) FROM CONTENT RECOMMENDATION GRAPHS TO NETWORK METRICS

In this section we aim to answer the following question: Is it feasible to assess delays (QoS) solely based on recommender features? An affirmative answer to such question can significantly simplify the design of QoS-aware recommenders as, in this case, the distance of contents in the recommender graph can be taken as a proxy to QoS metrics.

Figure 10 shows a decision tree (DT) that illustrates the feasibility of classifying contents as a function of the corresponding QoS at which they are served, solely based on their view counts, published date and on their position in the recommendation graph. The training and test sets were setup in a way such that the two target classes ($C = 1$ and $C = 0$) were balanced, i.e., 50% of the samples correspond to each class.

Each node in the tree corresponds to a decision. The root node corresponds to the decision that entails largest discriminatory power, and consists of classifying videos based on their depth in the recommender graph. As shown in Figure 10(a), if depth is less than 4, a fraction of 70% of the videos are served with high QoS (low ping values, $C = 0$). Alternatively, if the view count is larger than 2,42 million views, the content is classified as served with high QoS (low ping values and $C = 1$). Finally, if the age of the video is larger than one month, it is assumed not to be cached (flash crowds towards recent content are not captured in this simple DT).

Despite its simplicity, the presented DT already corresponds to an accuracy, precision and recall of 0.72, 0.63 and 0.88, respectively.

We repeated the same experiment, considering random walks as opposed to BFS to sample the recommendation graph. After this change, accuracy, precision and recall were 0.77, 0.68 and 0.83, respectively, and the obtained tree is presented in Figure 10(b). Note, in particular, that the depth threshold in the recommendation graph, below which a content is inferred to be cached, decreases when considering random walks. This is because under random walks all top-50 most popular videos are included in our dataset, increasing the relative number of samples with depth of 1 in the dataset. Such depth threshold of 2 is in agreement with Figure 11(a), also obtained with random walks and further discussed next.

2) FROM NETWORK METRICS TO NOVEL CONTENT RECOMMENDATIONS

Next, we consider the extent to which QoS-aware recommenders can compensate for low QoS. Figure 11(a) shows the relationships between recommender features (width, depth and view counts) and QoS (ping values). Each point corresponds to a content. The red (resp., blue) points correspond to contents served by high ping (resp., low ping) hosts. Note that “close” to a red point we typically have multiple blue points, corresponding to low ping values. Points which are “close” to each other in that figure are near each other in the recommendation graph, suggesting that the recommender can compensate for low QoS.

To further illustrate the feasibility of determining which contents can be served with high QoS, Figure 11(b) shows how view counts and depth impact QoS. Figs. 11(a) and 11(b) show a transition at depth 2, suggesting that contents in levels 1 and 2 can be assumed to have low ping values. A recommender can leverage that phase transition to tune the order of recommendations, compensating for low QoS.

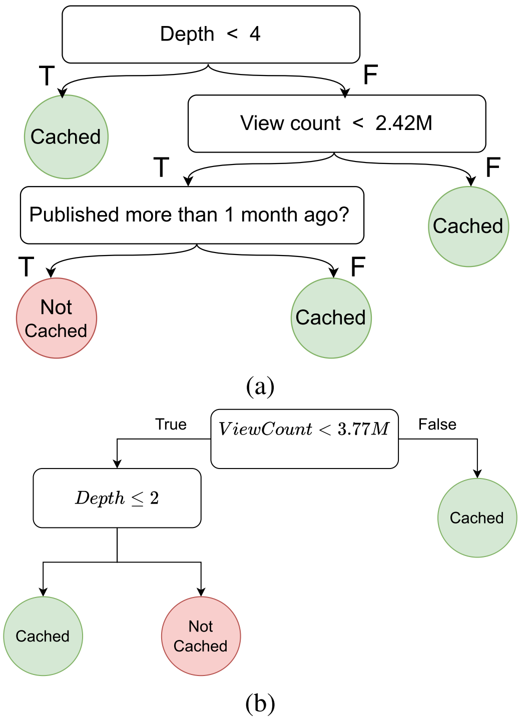


FIGURE 10. Decision tree: inferring QoS-related metrics from content-related attributes: (a) samples collected using breadth first search and (b) samples collected using random walk. Accuracy, precision and recall for cases (a) and (b) equal (0.72, 0.63, 0.83) and (0.77, 0.68, 0.83), respectively.

3) RECOMMENDERS AGAINST LOW QoS

To answer our main research question, we conducted experiments simulating scenarios in which a user randomly starts selecting one of the most trending videos, followed by videos from subsequent recommendation lists. These lists were presented in 2 ways: *(i)* the original order, i.e., like they would be in YouTube and *(ii)* according to an algorithm that prioritizes cached videos, the Cache-Aware & BFS-related Recommendations (CABaRet) [22]. CABaRet recommendations replace some non-cached videos with cached counterparts, and order videos in a way that cached videos are preferably presented at the top of the recommendation lists.

To determine whether a video is cached or not, we leveraged our measurements as described in the previous section. In particular, we use the inferred indicator C to determine whether a video is cached, and to eventually increment hit counts. While the authors of the CABaRet algorithm assumed, shrewdly, that the top 50 trending videos were cached, our measurements allowed the application of CABaRet algorithm with more information regarding the network conditions of the media servers.

Thus, we compared the cache-hit ratio (CHR) produced by YouTube's lists of recommendations against the lists generated by CABaRet. To this end, we varied the mechanism through which users select a video from a recommendation list, considering two alternatives: uniform and Zipf. The uniform distribution assumes that users select videos uniformly at random, whereas the Zipf distribution captures a preference towards videos ranked in top positions [33]. We also vary the two main CABaRet parameters: maximum depth ($\widetilde{D}$) and maximum width ($\widetilde{W}$). Larger values correspond to broader searches for cached contents in the recommendation graph, providing more flexibility for recommenders to compensate for QoS. In particular, when $\widetilde{D} = 1$ CABaRet only reorders the recommendations, whereas for $\widetilde{D} = 2$ it also replaces some non-cached videos with cached alternatives.

Figure 12 shows that CABaRet easily achieved a higher CHR than YouTube baseline. When the request workload is uniform, CABaRet requires larger $\widetilde{D}$ and $\widetilde{W}$ to show its benefits. This is because both replacements and reorderings of recommendations affect CHR under the Zipf workload, whereas the uniform workload is insensitive to reorderings.

Summary: Combining recommender and network measurements, we learned that recommendation reorderings are sufficient to increase CHR from 0.64 to 0.89 under a Zipf workload. Diminishing returns are gained by allowing recommendations to be replaced, in addition to reorderings. In particular, allowing for replacements of videos that are at most two hops away in the recommendation graph suffices to reach a CHR of 0.98.

IV. RELATED WORK

A. QoS AND QoE PARAMETERS

There is a vast literature on the interplay between QoS and QoE [25], [34]. The network-level QoS metrics considered in this study were delays and path lengths. Application-level QoS metrics include rebufferings and changes in resolution, whereas examples of QoE metrics include Mean Opinion Scores (MOS) and VMAF. The abandonment rate and fraction of watched video are two additional QoE metrics that implicitly characterize user experience.

In this work, we acknowledge previous studies on mapping between network-level QoS, application-level QoS and QoE. We contribute to this ecosystem by adding a new element, namely, recommender systems that are influenced by QoS and impact QoE. In what follows, we briefly overview some of the previous works that mapped network-level QoS, application-level QoS and QoE.

1) FROM NETWORK-LEVEL QoS TO APPLICATION-LEVEL QoS AND QoE

In [35] the authors proposed one of the first mappings between network-level QoS and QoE. They relied on random neural networks (RNNs) [36] for that matter. Among the conclusions, they indicate that packet losses significantly impact bitrates, which in turn impact QoE, and show that an RNN can capture such relations. In this work, we also account for the impact of bitrate changes on QoE, indicating that users may prefer to consume a video with fewer bitrate changes even if it is of less interest than its counterpart (Section II).

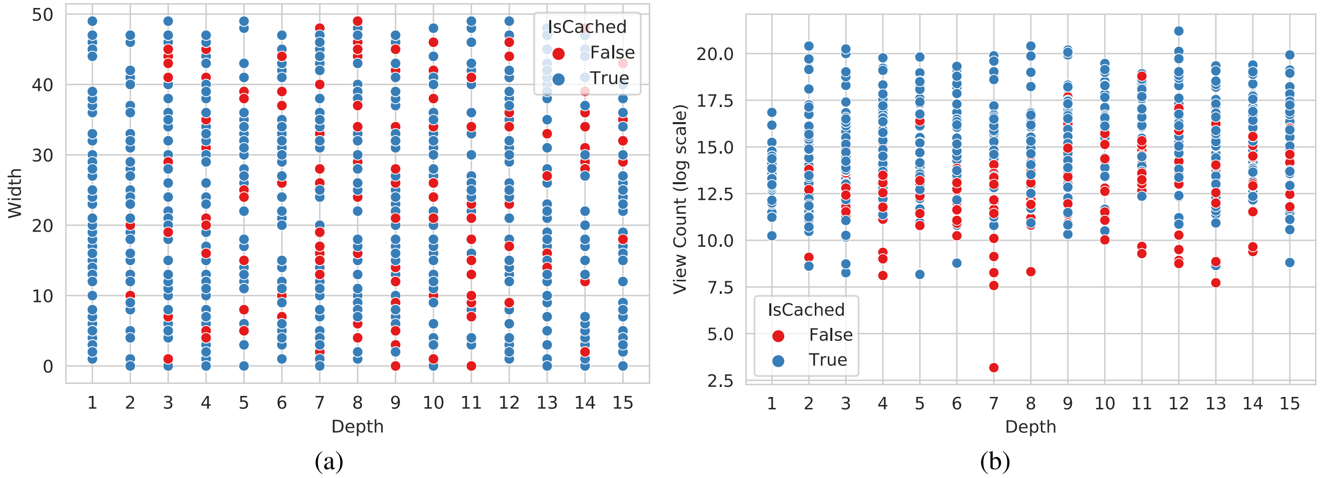


FIGURE 11. Relationships between recommender features (width, depth and log of view counts) and QoS (ping values). Red and blue points correspond to high and low ping values. (*a*) "close" to an red point we typically have multiple blue points; (*b*) there is a clear phase transition when depth grows beyond 2.

When traffic is encrypted, there are additional challenges to relate network-level QoS metrics and application-level QoS. In [24] the authors used passive network measurements from encrypted traffic to estimate application-level QoS metrics such as average bitrate, re-buferring ratio and startup time. For this purpose, they relied mainly on packet sizes and headers, assuming that the latter is not encrypted. In [23], the authors also relied on encrypted traffic information to predict application-level QoS, using machine learning for this goal. In addition to objective metrics, they also inferred subjective QoE metrics, such as the MOS, following the ITU P.1203 model [37]. In this study, we use VMAF as a subjective metric to assess QoE and contribute with experiments with real users to infer the experienced QoE as influenced by recommenders.

Traditional web caching and similarity-caching [10], [11], [38] can significantly impact network-level QoS. Indeed, research on similarity caching bodes well with the idea that the interplay between network-level QoS, application-level QoS and QoE can benefit the network and its users. In this study, we focus on a unified perspective towards QoS and QoR on user experience, which we refer to as content-aware QoE. The tuning of parameters of similarity-caches is, in essence, also related to QoS, QoR and QoE, utilizing recommendation systems to enhance QoS at the expense of diminished QoR. Nevertheless, previous studies on similarity caching have concentrated on the network performance aspect, neglecting the exploration of whether users are receptive to consuming content with lower QoR in exchange for higher QoS. To the best of our knowledge none of these studies carried out experiments with real users on the relationship between QoS, QoR and QoE, and we take a first step in this direction.

2) FROM APPLICATION-LEVEL QoS TO QoE

Many studies have analyzed the impact of application-level QoS on user experience at the application level. Some studies considered implicit metrics related to user engagement, such as retention and abandonment rates, and others considered explicit metrics, such as MOS and VMAF, also known as Video Quality Assessment (VQA) metrics. VQA metrics can act as a proxy for QoE.

The impact of QoS on user retention was considered in [19], [39], and [40]. In [39] it was shown that viewers start abandoning a video if it takes more than 2 seconds to start. In [19], the authors reported that users typically admit up to two rebufferings while watching a YouTube video, and that dropout rates rapidly grow after the second rebuffering event. These results motivated us to add up to two rebufferings per video (and never more than four). Additional details on the QoS impairments and demographics of users considered in our study can be found in Appendices A and B.

In [41] and [42] the authors considered user engagement as their target QoE metric. In particular, in [42] the authors accounted for users' interests and QoS factors to build an engagement/QoE predictive model. Our approach is aligned with this integrated perspective to predict QoE. However, while the authors of [42] inferred users' interests using Collaborative Filtering (CF), we conducted real experiments to directly assess users interests through their explicit feedback.

B. CONTEXT-AWARE AND MULTI-CRITERIA RECOMMENDERS

According to [43, §1.2], the taxonomy of recommender systems encompasses domain, purpose and context. The *context* is the *environment* in which the consumer receives a recommendation [44]. In a store, for instance, the recommendation of items that are out of stock is frustrating in a system employed to discover items to purchase.

Content recommenders are sensitive to the user context [45], [46]. The user context is typically understood as the day of the week, the place where the user is consuming its contents, or even the device. In this study, we considered a novel dimension of context, namely, QoS. We have shown that it is feasible to characterize and quantify QoS per content, and that a recommender can benefit from such a characterization. In particular, our work is complementary to the state of the art in recommendation systems [47], as our insights can be coupled to existing recommenders [48], [49], [50], [51], [52].

In this study, we indicate that QoS impairments are a key element impacting the recommendation system context, and provide a systematic way to quantify its effect on QoE. *Virtual* QoS impairments may be due to content that is out of cache, and in that sense are similar to impairments due *physical* items being out of stock [41].

C. RECOMMENDERS COMPENSATING FOR QoS

There is a large body of work on recommenders and QoS, and on their interplay [22], [53]. However, to the best of our knowledge there has been no measurements to assess the extent at which recommenders can compensate for low QoS.[3]

It is well known that proxy caches are a cheap and effective solution to counter bottlenecks in networked systems [56]. In particular, they are the first choice before considering a costly upgrade of the network infrastructure. In this study, we have experimentally shown that content recommenders can catalyze the benefits of caching, further increasing its potential at virtually zero costs. In [30] and [57] the authors investigated statistical properties of the paths towards contents in YouTube and Netflix. In this study, we build on such measurements, and consider how a content recommender can leverage those to benefit the users.

There has been a recent surge in interest in the relationship between cache networks and recommendation systems [22], [58], [59], accounting for its impact on similarity caching [11], hit rates at femto caches [20] and network coding [60]. In [21], for instance, the authors considered the problem of maximizing the cache hit rate, subject to a maximum distortion in the demand caused by a biased recommendation system. All previous efforts considered *synthetic data* to parameterize and evaluate the proposed solutions. We envision that the dataset presented in this paper, together with the derived insights, will serve as basis for a data-driven evaluation of those early works and a reality check of assumptions.

Optimal allocations in cache networks typically require knowing the cost-to-go, i.e., the cost incurred by a miss up until finding the content at the closest cache [61]. The methodology considered in this study to characterize QoS at the content level can be used for those purposes, as well as to parametrize content recommenders. The interplay between QoS and recommenders involves both human-related and network-related elements. CABaRet [22] comprises a conceptual framework to capture these factors in an integrated fashion. In this study, we report proof-of-concept network measurements that are complementary to CABaRet.

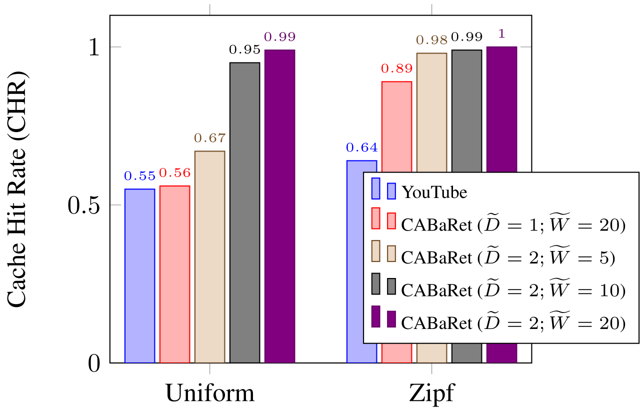


FIGURE 12. Cache Hit Rate (CHR) varying workloads and recommenders.

V. CONCLUSION AND FUTURE WORK

Content recommenders play a fundamental role in the way content is consumed on the Internet. Nonetheless, they are typically devised without accounting for one of the most critical aspects of the Internet infrastructure, namely, its best effort nature.

In this study, we reported insights from experiments with real subjects to assess their sensitivity to recommendations adjusted based on QoS-related features. In particular, we quantified how QoS and QoR impact QoE and users' choice towards their favorite videos. Our experimental results determine how users' preferences depend on different factors, such as the content category and network QoS. Thus, they allow us to find thresholds for the design of QoS-aware content recommendation systems.

We observed that the recommenders have significant flexibility with respect to the available contents to be offered and their corresponding QoS levels. We discovered that contents located at depths of one or two in the recommender graph are highly likely to be cached near users, suggesting a simple heuristic to favor QoS while satisfying users interests with an increased overall QoE. Accounting for QoS, in turn, recommenders have the potential to benefit users (who experience higher levels of QoE), system administrators (who reduce their service costs, e.g., serving more cached content and reducing backhaul traffic) and content providers (who gain increased retention rates for their contents).

We envision that our work opens up a number of interesting avenues for future research, in the intersections between similarity caching, QoE and QoS. In particular, there are limitations in the evaluation of our experiments that present opportunities for future research. First, we did not account for potential correlation among samples produced by the same user, and we envision the use of General Linear Mixed

[3] A preliminary version of this part of our work appeared as a poster at the Internet Measurement Conference (IMC) 2021 [54] and at Globecom 2022 [55].

Models for that purpose. Second, such correlations may be due to *a priori* interests of users towards certain categories, which can be explicitly taken into account in future work.

APPENDIX A
QoS IMPAIRMENTS

In our experiments with real users we consider 7 video pairs, labeled from 1 to 7. Additionally, we include a video pair labeled 0, where both videos contain the same content but differ only in terms of QoS impairments. In each video pair, one of the videos contained video impairments purposely introduced to assess the impact of QoS on QoE. The impairments were designed to replicate real-life scenarios by synchronizing instances of rebuffering and alterations in resolution. Changes in resolution in video demand are typically preceded or expected to occur alongside instances of rebuffering (also known as stalling) either as a result of ABR mechanisms or prompted by a user-initiated resolution change [41].

Next, we discuss the impact of reducing resolution and of rebufferings on user experience.

A. CHANGES IN RESOLUTION

VMAF, which stands for Video Multimethod Assessment Fusion, is a video quality assessment metric developed by Netflix [62]. Its primary objective is to predict the perceived visual quality of a video by comparing it to the original reference video. This metric holds widespread popularity in the video streaming sector for evaluating video encoding and compression performance.

By leveraging VMAF, we can efficiently compute the impact of poor resolution intervals on the videos within our experiment, in addition with the users' feedbacks. Figures 13 and 14 show the boxplot of VMAF values and VMAF per session over time, respectively, for five out of the eight sessions, wherein VMAF was non-constant.

1) COMPUTING VMAF

VMAF can only be used for comparing videos with the same duration. This limitation arises because VMAF scores are computed and compared on a per-frame basis. However, due to video editing that involved changes in resolution and rebufferings, the original and perturbed videos may have different frame counts. In order to match the number of frames in the original videos and in the videos perturbed with rebufferings, we proceeded as follows: First, we compare the elements of each video-pair using the original videos downloaded from YouTube at different resolutions. Second, in the intervals where the video was presented at its baseline resolution, we report in Figure 14 the original VMAF score of 100. Third, in the intervals where we intentionally reduced the resolution, we report in Figure 14 the average VMAF score of the low-resolution video, considering that VMAF tends to be concentrated around its mean. Supporting this observation, Figure 13 presents a boxplot of VMAFs during different phases of video streaming, indicating that variations in VMAF over time, given constant resolutions, are typically on the order of 10 units.

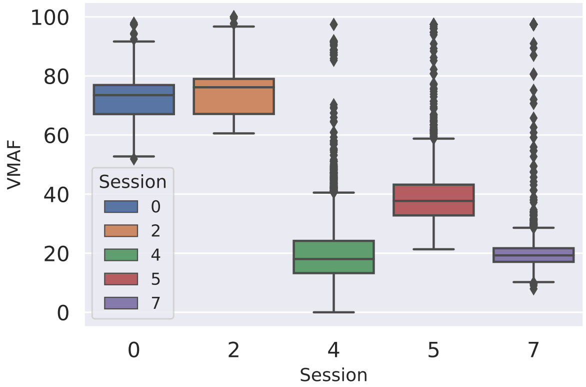


FIGURE 13. Boxplots of VMAF over different sessions.

2) ASSESSING AND COMPARING VMAF IN OUR EXPERIMENTS

All high QoS videos were presented with a resolution of 720p. The resolution of the low QoS videos, in contrast, is different in each video-pair. In sessions 1, 3 and 6, the resolution of the low QoS videos did not change throughout the sessions, and was set to 240p at session 1, and 360p at sessions 3 and 6. Session 1, with the lowest resolution, corresponds to a mean Annoyance of 3.83. The mean Annoyance, for sessions 3 and 6, was 3.32 and 3.33, respectively (see Table 2).

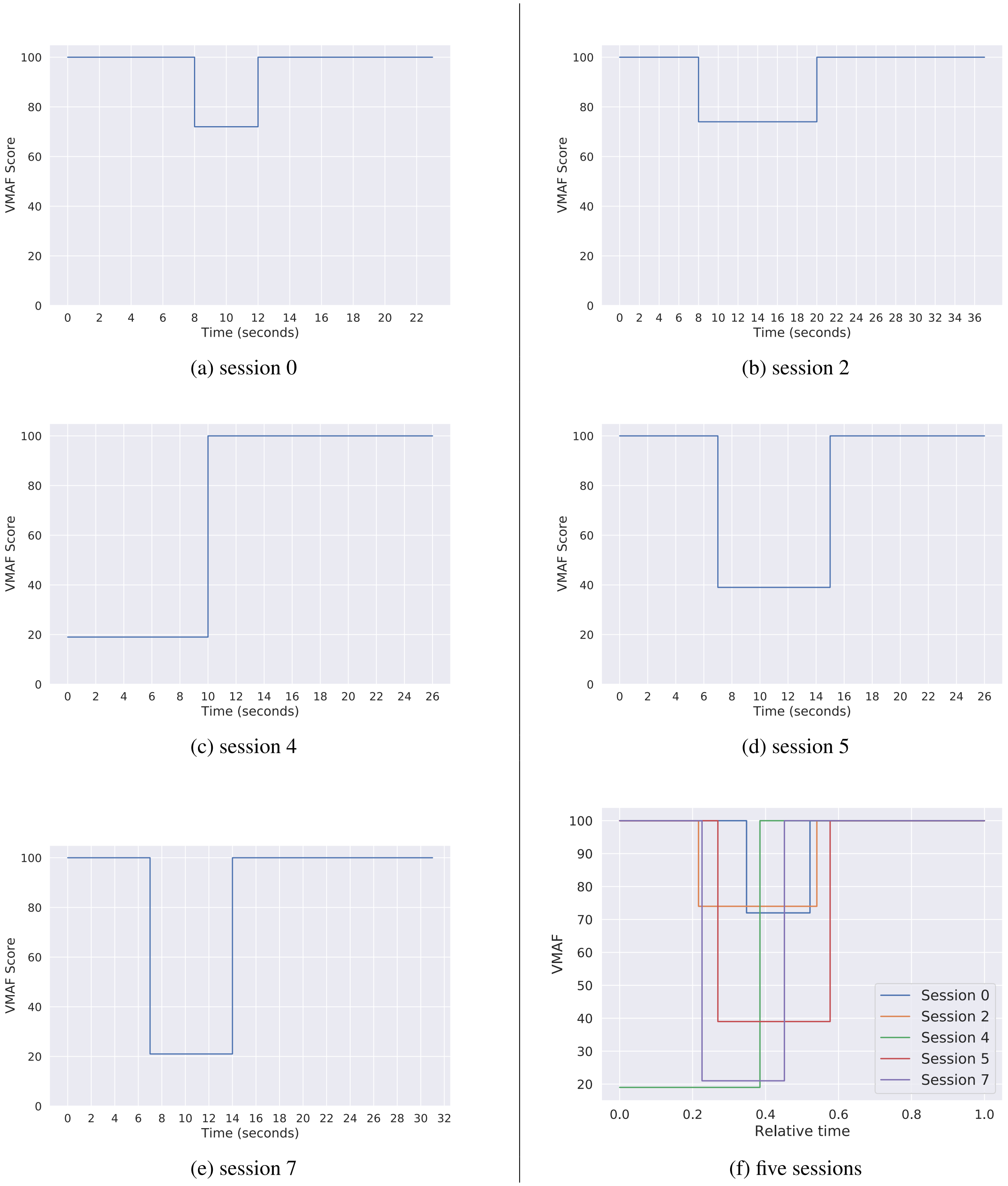
In sessions 0, 2, 4, 5 and 7, the baseline resolution was set to 720p. However, in those sessions, the resolution is reduced at certain intervals to values as low as 144p. Figure 14 reports, for each session, the intervals during which a reduction in resolution occurred. When the reduction was most extreme, to 144p, VMAF reached its lowest value, of 20, in sessions 4 and 7.

Note that users experienced a higher baseline resolution in the sessions reported in Figure 14 compared to the constant resolution sessions 1, 3 and 6. Nonetheless, the variations in resolution over time, e.g., due to bit rate changes caused by network impairments, resulted in comparable or even higher levels of Annoyance for the sessions depicted in Figure 14 (see Table 2). This aligns with findings in [63], where authors report that more fluctuations in video quality lead to decreased user engagement.

B. REBUFFERINGS

Table 8 presents details about the rebufferings introduced in each of the 7 videos with QoS impairments.

There are a number of studies indicating how users react to impairments. In [41], for instance, the authors show that sessions with two impairments tend to get abandoned earlier, after the occurrence of the second impairment, than sessions with only one impairment. For this reason, the authors of [41] focus on sessions with up to two impairments. This is in agreement with our video sessions, wherein all sessions have



(a) session 0

(b) session 2

(c) session 4

(d) session 5

(e) session 7

(f) five sessions

FIGURE 14. VMAF as a function of time for low QoS videos.

two impairments (a notable exception being session 6, with three impairments).

With respect to the duration of the stalls, in [64] the authors indicate that 85% of stalls last less than 5 seconds. According to [65], the number of stalls is more relevant than their duration when assessing QoE, measured through the mean opinion score (MOS). The impairments considered in this work are in agreement with such observations, noting that all the considered stalls last less than 5 seconds.

TABLE 8. Rebufferings on video sessions (Low QoS videos of the pairs).

Session	Description	Total Duration	1st Rebuffering		2nd Rebuffering		3rd Rebuffering		4th Rebuffering	
		(seconds)	Interruption (instant)	Duration (seconds)	Interruption (instant)	Duration (seconds)	Interruption (instant)	Duration (seconds)	Interruption (instant)	Duration (seconds)
0	Sports; 100m Sprint Olympics Record	23	0	2	6	2	13	2	16	3
1	Soccer; Final 2002 World Cup	32	8	3	20.9	3				
2	Comedy; YouTube Channel: "Porta dos Fundos"	38	4	4	20	3.5				
3	Animals; Panda playing	26	5	3	17.9	3.5				
4	Radical sports; Biking, skydiving, etc	27	7	3	17	2				
5	Animal documentary; Jaguar vs crocodile	27	5	2	13	3				
6	Science documentary; Black holes	36	4	3	26.5	2	29.5	2		
7	Music; Antonio Vivaldi Spring Concerto	32	4	3	12.3	3				

APPENDIX B DEMOGRAPHICS

To preserve participants' privacy, we anonymized the dataset, and in what follows we present some aggregate statistics pertaining the collected dataset, comprised mostly of graduate and undergrad Computer Science students. Roughly 82% of the participants were male. A significant portion of the participants (36%) reported that they access, on a daily basis, exclusively content services that require subscription, whereas a small fraction (14%) accesses only free content services. Half of the participants (50%) access both types of contents. To access video content, 65% of participants reported that they use cable services, 9% 4G and 91% WiFi (including public access points).

APPENDIX C ADDITIONAL MODELS TO CHARACTERIZE USER BEHAVIOR

A. LINEAR AND QUADRATIC DISCRIMINANT ANALYSIS

Next, we consider discriminant analysis. Whereas logistic regression models the *target* variable as a Bernoulli trial, discriminant analysis leverages the nature of the *input* variables. Discriminant analysis assumes that the joint conditional distribution of the input, given the target class label, is given by a multivariate normal distribution with same (resp., distinct) covariance matrices across classes when considering LDA (resp., QDA) [29]. Figure 7 indicates the best binormal distributions which approximate our input variables. The two binormal distributions have similar covariance matrices (indicated with ellipses), hence LDA and QDA yield the exact same accuracy. Nonetheless, as the normal approximation is not very accurate, logistic regression still yields slightly superior results.

In Figure 7, the solid separating line was obtained by training LDA. Both separating lines, i.e., the one obtained with logistic regression (dotted line) and the one with LDA (solid line), are effective in producing splitting rules that entail few classification errors, and indicate the expressive power of linear models for our purposes.

B. KNN AND PERFORMANCE COMPARISON

So far we considered parametric classifiers. Next, we evaluate the K-nearest neighbors (KNN) classifier, a non-parametric classifier which yields separating curves that don't admit a closed-form expression [29]. We adopt the Euclidean distance, and vary K between 1 and 50 (reporting in Table 9 results for $K = 1, 10, 30$).

Table 9 compares the performance of the classical statistical methods considered so far. It indicates that the metrics assessed through the validation set vary slightly among methods. In particular, the accuracy varies between 0.768 and 0.788. We also find that 30-NN performed consistently as the best. Regarding KNN, when the number of neighbors (K) is small, we get a very wiggly frontier which causes overfitting; on the other hand, for larger values of K ($K = 50$) the model simplifies and the frontier produces the same accuracy as the LDA line.

C. REMOVING OUTLIERS

Next, we report heuristics to remove outliers. We define an outlier as a sample that is inconsistent with a basic rule implied by the domain under study. Outliers in our dataset may be due to misunderstanding of the posed questions or due to the granularity of the scales considered (e.g., users have to rate interests over videos in a scale between 0 and 5 stars).

In particular, we consider the following simple consistency check. If a user assigns the same number of stars to two videos in the same video pair (DiffInterests=0) and reports Annoyance by the QoS impairments in the low QoS video (Annoyance >3), the user must choose the high QoS video as his preferred consumption choice. Clearly, we assume that in this situation there is no tradeoff between quality of service and quality of content.

In summary, if the following three rules are satisfied simultaneously, the sample is considered an outlier: (*i*) DiffInterests $= 0$; (*ii*) Annoyance > 3; (*iii*) PreferredVideo $=$ lowQoS. We identified 59 samples that concomitantly matched the three rules above. Those entries were classified as outliers and were removed from the analysis that follows.

1) ADDITIONAL REMARKS

Note that we require criterion (*ii*) to classify a sample as an outlier. A careful analysis of the instances wherein (*i*) and (*iii*) are satisfied (and (*ii*) is not) shows that users preferred the low QoS video in those cases because the scale of interests (ranging among integers from 0 to 5) was too coarse. In the textual justifications for their choices, those users typically referred to subtle preferences towards the content of the low QoS video (e.g., using words such as *funnier* and *arose my curiosity*) even though they indicated the same interest

TABLE 9. Performance of classification models.

Model	Accuracy	F1-Score	Recall	Precision
Basic models				
Logistic Regression	0.776	0.7716	0.7748	0.7727
LDA and QDA	0.768	0.7632	0.7619	0.7625
1-Nearest Neighbor	0.768	0.7632	0.7654	0.7641
10-Nearest Neighbor	0.768	0.7639	0.7678	0.7651
30-Nearest Neighbor	0.788	0.7843	0.7888	0.7855
Additional models (trained with outliers removed)				
Nearest Neighbors	0.8336	0.8282	0.8282	0.8301
Linear SVM	0.8122	0.8004	0.7957	0.8126
RBF SVM	0.8398	0.8335	0.8332	0.836
Random Forest	0.8557	0.8523	0.8588	0.8532
Neural Network	0.8547	0.8506	0.8558	0.8523
AdaBoost	0.8409	0.8325	0.8295	0.8406
Naive Bayes	0.8112	0.7997	0.7958	0.8103
QDA	0.8101	0.7998	0.797	0.8077

towards the two videos in the numerical scale. The *polarity score* metric captures those subtleties (Section II-F), and we keep those instances in our analysis.

D. PERFORMANCE EVALUATION OF MACHINE LEARNING METHODS

Next, we report results on the performance of classifiers after removing outliers (bottom part of Table 9). We consider a number of different machine learning models, which are typically more complex than the simpler statistical models discussed so far.

Table 9 shows our results. For each classifier, we report its accuracy, precision, recall and f1 scores. The definitions of those standard performance metrics can be found in [27]. In order to validate the results, we adopted ten fold cross validation. The parameterization considered for each of the classification models (e.g., number of layers in the neural network model) was the best achievable under 10 fold cross validation (details omitted due to space constraints).

The neural network and the random forest yield best accuracy. Although those more complex machine learning solutions produce higher accuracy, they are usually not amenable to interpretation. In the following section, we balance between interpretability and accuracy through the use of decision trees.

APPENDIX D
TOWARDS QoS-AWARE RECOMMENDERS

Next, we use decision trees to design QoS-aware recommendation systems amenable to interpretation. We begin considering the whole dataset, and progressively consider special instances, first removing outliers and then focusing on the music category.

To assess the impact of the multiple features on user choices, we train a C4.5 decision tree model to predict the preference of users towards videos. Figure 15 shows a pruned version of the trained decision tree (Fig. 15). Each node of the tree, except for the leaves, contains a binary decision rule used to split the dataset. In addition, each node also contains the number of samples at that stage wherein users chose the high and low QoS videos (when the latter is greater, the node is marked in red).

The root node of the tree splits the dataset using the difference in interests (DiffInterests) between the two videos. The fact that this feature appears at the top of the tree means that it has significant classification relevance. If DiffInterests $\leq$ 0, users tend to choose the high QoS video. Otherwise, the low QoS tends to be preferred. In both cases, Annoyance also appears as a splitting feature at lower levels of the tree, indicating the role of QoS in the choices.

Following the path among nodes #0 and #1 we note that if DiffInterests < 0 it is extremely likely that the chosen video is the high QoS. In addition, the path across node #0, #8 and #12 indicates that PolarityScore (our sentiment analysis metric) is fairly useful in separating the data when DiffInterests > 0, (nodes #0, #8 and #12). Annoyance is used to further split the samples when PolarityScore is low (see Section II-F).

The decision trees also serve to support the hypothesis that more positive comments indicate a preference towards low QoS videos. The path across nodes #0, #1, #2 and #4 in Fig. 15(b) bodes well with the discussion presented in Section II-F. A threshold of 0.272 (close to 0.35 chosen in Section II-F) is selected in order to split the data so that that samples with PolarityScore < 0.272 are classified in the class wherein users prefer the high QoS video.

1) FOCUSING ON MUSIC

To illustrate how decision trees may vary across categories, we focus on music, which is by far the most watched category on YouTube. According to Table 2, users were less tolerant to QoS impairments in the music category when contrasted against the other considered categories. Accordingly, training a decision tree from samples from the music subset (video pair 7), "Annoyance" appears at the first level of the classification process and participates in the separation of 70% of the samples.

2) BEYOND DECISION TREES

Random forests are the natural extension of decision trees. Using random forests, we obtained our highest levels of performance (see Table 9). Using random forests, we can also estimate the importance of the different features for classification purposes. The feature importance is the mean importance averaged across all trees in the forest. Using such approach, difference between interests (DiffInterests) and Annoyance are the two most important features. Together, they capture QoR and QoS aspects that impact user choices, respectively, indicating that the two play an important role on QoE.

3) PRACTICAL ENGINEERING IMPLICATIONS

In this section, we presented a systematic way to quantitatively assess parameters to be used by QoS-aware content recommendation systems. Some of our results are intuitive,

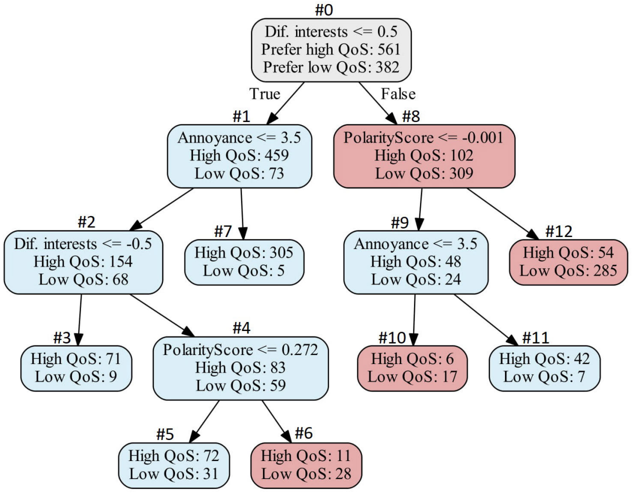


FIGURE 15. Decision trees allow us to interpret user choices.

e.g., some users will trade between QoS and QoR when deciding which content to consume. Nonetheless, the value of our analysis stands for a reality check of intuitions against real data and, more importantly, quantitatively assessing the extent at which different factors impact QoE. This quantitative assessment is instrumental and key for the design of novel recommendation systems. Although we applied our analysis over the entire population of participants, we envision that it can be also adopted in a user-oriented framework, wherein annoyance and interests are tracked. The sensitivity of each user with respect to QoS and QoR across different categories can then be *learned* and used for personalized recommendation.

REFERENCES

[1] R. Zhou, S. Khemmarat, and L. Gao, "The impact of YouTube recommendation system on video views," in *Proc. 10th ACM SIGCOMM Conf. Internet Meas.*, Melbourne, VIC, Australia, M. Allman, Ed., Nov. 2010, pp. 404–410, doi: 10.1145/1879141.1879193.

[2] C. A. Gomez-Uribe and N. Hunt, "The Netflix recommender system: Algorithms, business value, and innovation," *ACM Trans. Manage. Inf. Syst.*, vol. 6, no. 4, pp. 1–19, Jan. 2016, doi: 10.1145/2843948.

[3] RecSys. (2020). *Cars Workshop: Context-Aware Recommender Systems Workshop*. [Online]. Available: https://cars-workshop.com/cars-workshops

[4] T. Jambor and J. Wang, "Optimizing multiple objectives in collaborative filtering," in *Proc. 4th ACM Conf. Recommender Syst.*, Sep. 2010, pp. 55–62.

[5] K. Christakopoulou, J. Kawale, and A. Banerjee, "Recommendation with capacity constraints," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 1439–1448.

[6] K. Christakopoulou, "Towards recommendation systems with real-world constraints," Ph.D. thesis, Univ. Minnesota, Minneapolis, MN, USA, 2018. [Online]. Available: https://hdl.handle.net/11299/201062

[7] J. Park and K. Nam, "Group recommender system for store product placement," *Data Mining Knowl. Discovery*, vol. 33, no. 1, pp. 204–229, Jan. 2019.

[8] M. T. Ribeiro, A. Lacerda, A. Veloso, and N. Ziviani, "Pareto-efficient hybridization for multi-objective recommender systems," in *Proc. 6th ACM Conf. Recommender Syst.*, Sep. 2012, pp. 19–26.

[9] *Visual Networking Index: Forecast and Trends, 2017–2022*, Cisco, San Jose, CA, USA, 2018.

[10] Y. B. Mazziane, S. Alouf, G. Neglia, and D. S. Menasche, "Computing the hit rate of similarity caching," in *Proc. IEEE Global Commun. Conf.*, Dec. 2022, pp. 141–146.

[11] G. Neglia, M. Garetto, and E. Leonardi, "Similarity caching: Theory and algorithms," *IEEE/ACM Trans. Netw.*, vol. 30, no. 2, pp. 475–486, Apr. 2022, doi: 10.1109/TNET.2021.3126368.

[12] J. Zhou, O. Simeone, X. Zhang, and W. Wang, "Adaptive offline and online similarity-based caching," *IEEE Netw. Lett.*, vol. 2, no. 4, pp. 175–179, Dec. 2020.

[13] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "QoS-aware web service recommendation by collaborative filtering," *IEEE Trans. Services Comput.*, vol. 4, no. 2, pp. 140–152, Apr. 2011.

[14] S. Kastanakis, P. Sermpezis, V. Kotronis, D. Menasché, and T. Spyropoulos, "Network-aware recommendations in the wild: Methodology, realistic evaluations, experiments," *IEEE Trans. Mobile Comput.*, vol. 21, no. 7, pp. 2466–2479, Jul. 2022, doi: 10.1109/TMC.2020.3042606.

[15] M. Dehghan, L. Massoulié, D. Towsley, D. S. Menasché, and Y. C. Tay, "A utility optimization approach to network cache design," *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 1013–1027, Jun. 2019.

[16] P. Sermpezis, T. Spyropoulos, L. Vigneri, and T. Giannakas, "Femto-caching with soft cache hits: Improving performance with related content recommendation," in *Proc. IEEE Global Commun. Conf.*, Singapore, Dec. 2017, pp. 1–7, doi: 10.1109/GLOCOM.2017.8254035.

[17] T. Giannakas, P. Sermpezis, and T. Spyropoulos, "Show me the cache: Optimizing cache-friendly recommendations for sequential content access," in *Proc. IEEE 19th Int. Symp. World Wireless, Mobile Multimedia. (WoWMoM)*. Chania, Greece: IEEE Computer Society, Jun. 2018, pp. 14–22, doi: 10.1109/WoWMoM.2018.8449731.

[18] P. Sermpezis, T. Giannakas, T. Spyropoulos, and L. Vigneri, "Soft cache hits: Improving performance through recommendation and delivery of related content," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1300–1313, Jun. 2018, doi: 10.1109/JSAC.2018.2844983.

[19] H. Nam, K.-H. Kim, and H. Schulzrinne, "QoE matters more than QoS: Why people stop watching cat videos," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (IEEE INFOCOM)*, San Francisco, CA, USA, Apr. 2016, pp. 1–9, doi: 10.1109/INFOCOM.2016.7524426.

[20] L. E. Chatzieleftheriou, G. Darzanos, M. Karaliopoulos, and I. Koutsopoulos, "Joint user association, content caching and recommendations in wireless edge networks," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 46, no. 3, pp. 12–17, Jan. 2019.

[21] L. E. Chatzieleftheriou, M. Karaliopoulos, and I. Koutsopoulos, "Caching-aware recommendations: Nudging user preferences towards better caching performance," in *Proc. IEEE Conf. Comput. Commun.*, Atlanta, GA, USA, May 2017, pp. 1–9, doi: 10.1109/INFOCOM.2017.8057031.

[22] S. Kastanakis, P. Sermpezis, V. Kotronis, and X. Dimitropoulos, "CABaRet: Leveraging recommendation systems for mobile edge caching," in *Proc. Workshop Mobile Edge Commun.*, Budapest, Hungary, Aug. 2018, pp. 19–24, doi: 10.1145/3229556.3229563.

[23] M. J. Khokhar, T. Ehlinger, and C. Barakat, "From network traffic measurements to QoE for internet video," in *Proc. IFIP Netw. Conf. (IFIP Networking)*, May 2019, pp. 1–9.

[24] T. Mangla, E. Halepovic, M. Ammar, and E. Zegura, "EMIMIC: Estimating HTTP-based video QoE metrics from encrypted network traffic," in *Proc. Netw. Traffic Meas. Anal. Conf. (TMA)*, Jun. 2018, pp. 1–8.

[25] M. Ghosh, D. C. Singhal, and R. Wayal, "DeSVQ: Deep learning based streaming video QoE estimation," in *Proc. 23rd Int. Conf. Distrib. Comput. Netw.*, Jan. 2022, pp. 19–25.

[26] O. Izima, R. de Fréin, and A. Malik, "A survey of machine learning techniques for video quality prediction from quality of delivery metrics," *Electronics*, vol. 10, no. 22, p. 2851, Nov. 2021.

[27] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.

[28] C. J. Hutto and E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis of social media text," in *Proc. 8th Int. Conf. Weblogs Social Media (ICWSM)*, E. Adar, P. Resnick, M. D. Choudhury, B. Hogan, and A. Oh, Eds. Ann Arbor, MI, USA: AAAI Press, Jun. 2014, pp. 216–225. [Online]. Available: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/view/8109

[29] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, vol. 112. Midtown Manhattan, NY, USA: Springer, 2013.

[30] T. V. Doan, L. Pajevic, V. Bajpai, and J. Ott, "Tracing the path to YouTube: A quantification of path lengths and latencies toward content caches," *IEEE Commun. Mag.*, vol. 57, no. 1, pp. 80–86, Jan. 2019.

[31] (2021). *Scikit Learn*. [Online]. Available: https://scikit-learn.org, sklearn.preprocessing.MinMaxScaler

[32] S. Sundaresan, N. Feamster, R. Teixeira, and N. Magharei, "Measuring and mitigating web performance bottlenecks in broadband access networks," in *Proc. Internet Meas. Conf. (IMC)*, 2013, pp. 213–226.

[33] D. K. Krishnappa, M. Zink, C. Griwodz, and P. Halvorsen, "Cache-centric video recommendation: An approach to improve the efficiency of YouTube caches," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 11, no. 4, pp. 48:1–48:20, Jun. 2015, doi: 10.1145/2716310.

[34] P. Lebreton and K. Yamagishi, "Quitting ratio-based bitrate ladder selection mechanism for adaptive bitrate video streaming," *IEEE Trans. Multimedia*, early access, Jan. 18, 2023, doi: 10.1109/TMM.2023.3237168.

[35] A. P. C. da Silva, M. Varela, E. D. S. E. Silva, R. M. M. Leão, and G. Rubino, "Quality assessment of interactive voice applications," *Comput. Netw.*, vol. 52, no. 6, pp. 1179–1192, Apr. 2008.

[36] E. Gelenbe, "Random neural networks with negative and positive signals and product form solution," *Neural Comput.*, vol. 1, no. 4, pp. 502–510, Dec. 1989.

[37] *Parametric Bitstream-Based Quality Assessment of Progressive Download and Adaptive Audiovisual Streaming Services Over Reliable Transport-Quality Integration Module*, Int. Telecommun. Union, Geneva, Switzerland, 2017.

[38] T. S. Salem, "Online learning for network resource allocation," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 50, no. 3, pp. 20–23, Dec. 2022.

[39] S. S. Krishnan and R. K. Sitaraman, "Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs," *IEEE/ACM Trans. Netw.*, vol. 21, no. 6, pp. 2001–2014, Dec. 2013.

[40] R. K. Mok, E. W. Chan, and R. K. Chang, "Measuring the QoE of HTTP video streaming," in *Proc. Integr. Netw. Manage.*, 2011, pp. 485–492.

[41] M. Plakia, E. Tzamousis, T. Asvestopoulou, G. Pantermakis, N. Filippakis, H. Schulzrinne, Y. Kane-Esrig, and M. Papadopouli, "Should I stay or should I go: Analysis of the impact of application QoS on user engagement in YouTube," *ACM Trans. Model. Perform. Eval. Comput. Syst.*, vol. 5, no. 2, pp. 1–32, Jun. 2020.

[42] X. Tan, Y. Guo, M. A. Orgun, L. Xue, and Y. Chen, "An engagement model based on user interest and QoS in video streaming systems," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–11, Sep. 2018.

[43] K. Falk, *Practical Recommender Systems*. Shelter Island, NY, USA: Manning, 2018.

[44] M. T. Ribeiro, N. Ziviani, E. S. D. Moura, I. Hata, A. Lacerda, and A. Veloso, "Multiobjective Pareto-efficient approaches for recommender systems," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 4, pp. 1–20, Jan. 2015.

[45] N. M. Villegas, C. Sánchez, J. Díaz-Cely, and G. Tamura, "Characterizing context-aware recommender systems: A systematic literature review," *Knowl.-Based Syst.*, vol. 140, pp. 173–200, Jan. 2018.

[46] V. Bajpai, S. Ahsan, J. Schönwälder, and J. Ott, "Measuring YouTube content delivery over IPv6," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 47, no. 5, pp. 2–11, Oct. 2017, doi: 10.1145/3155055.3155057.

[47] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for YouTube recommendations," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 191–198.

[48] T. Agagu and T. Tran, "Context-aware recommendation methods," *Int. J. Intell. Syst. Appl.*, vol. 10, no. 9, pp. 1–12, 2018, doi: 10.5815/ijisa.2018.09.01.

[49] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Boston, MA, USA: Springer, 2011, pp. 217–253, doi: 10.1007/978-0-387-85820-3_7.

[50] L. Baltrunas, B. Ludwig, and F. Ricci, "Matrix factorization techniques for context aware recommendation," in *Proc. 5th ACM Conf. Recommender Syst.*, B. Mobasher, R. D. Burke, D. Jannach, and G. Adomavicius, Eds., Chicago, IL, USA, Oct. 2011, pp. 301–304, doi: 10.1145/2043932.2043988.

[51] T. V. Nguyen, A. Karatzoglou, and L. Baltrunas, "Gaussian process factorization machines for context-aware recommendations," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, S. Geva, A. Trotman, P. Bruza, C. L. A. Clarke, and K. Järvelin, Eds., Gold Coast, QLD, Australia, Jul. 2014, pp. 63–72, doi: 10.1145/2600428.2609623.

[52] R. Pagano, P. Cremonesi, M. Larson, B. Hidasi, D. Tikk, A. Karatzoglou, and M. Quadrana, "The contextual turn: From context-aware to context-driven recommender systems," in *Proc. 10th ACM Conf. Recommender Syst.*, S. Sen, W. Geyer, J. Freyne, and P. Castells, Eds., Boston, MA, USA, Sep. 2016, pp. 249–252, doi: 10.1145/2959100.2959136.

[53] P. Sermpezis, S. Kastanakis, J. I. Pinheiro, F. Assis, M. Nogueira, D. Menasché, and T. Spyropoulos, "Towards QoS-aware recommendations," in *Proc. CARS Workshop ACM Recommender Systems (RecSys)*, 2020, pp. 1–8.

[54] M. Nogueira, D. Menasché, and P. Sermpezis, "Poster: Can recommenders compensate for QoS?" in *Proc. ACM Internet Meas. Conf. (IMC)*, 2021, pp. 1–2. [Online]. Available: https://conferences.sigcomm.org/imc/2021/pdf/QoS.pdf

[55] M. Nogueira, C. Bravo, D. Menasché, T. Spyropoulos, and P. Sermpezis, "Can recommenders compensate for low QoS?" in *Proc. IEEE Global Commun. Conf.*, Dec. 2022, pp. 4185–4190.

[56] S. Traverso, K. Huguenin, I. Trestian, V. Erramilli, N. Laoutaris, and K. Papagiannaki, "Social-aware replication in geo-diverse online systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 2, pp. 584–593, Feb. 2015, doi: 10.1109/TPDS.2014.2312197.

[57] T. V. Doan, V. Bajpai, and S. Crawford, "A longitudinal view of Netflix: Content delivery over IPv6 and content cache deployments," in *Proc. IEEE Conf. Comput. Commun.*, Toronto, ON, Canada, Jul. 2020, pp. 1073–1082, doi: 10.1109/INFOCOM41043.2020.9155367.

[58] D. Zheng, Y. Chen, M. Yin, and B. Jiao, "Cooperative cache-aware recommendation system for multiple internet content providers," *IEEE Wireless Commun. Lett.*, vol. 9, no. 12, pp. 2112–2115, Dec. 2020, doi: 10.1109/LWC.2020.3014266.

[59] Z. Lin and W. Chen, "Joint pushing and recommendation for susceptible users with time-varying connectivity," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6, doi: 10.1109/GLOCOM.2018.8647838.

[60] B. Zhu and W. Chen, "Coded caching with joint content recommendation and user grouping," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6, doi: 10.1109/GLOCOM.2018.8647664.

[61] S. Ioannidis and E. Yeh, "Adaptive caching networks with optimality guarantees," *IEEE/ACM Trans. Netw.*, vol. 26, no. 2, pp. 737–750, Apr. 2018, doi: 10.1109/TNET.2018.2793581.

[62] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, "Toward a practical perceptual video quality metric," *Netflix Tech Blog*, vol. 6, no. 2, p. 2, 2016.

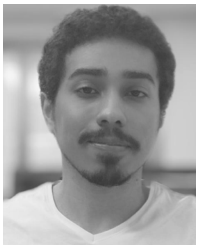
[63] A. Ahmed, Z. Shafiq, H. Bedi, and A. Khakpour, "Suffering from buffering? Detecting QoE impairments in live video streams," in *Proc. IEEE 25th Int. Conf. Netw. Protocols (ICNP)*, Oct. 2017, pp. 1–10.

[64] H. Nam, K.-H. Kim, D. Calin, and H. Schulzrinne, "YouSlow: A performance analysis tool for adaptive bitrate video streaming," in *Proc. ACM Conf. SIGCOMM*, Aug. 2014, pp. 111–112.

[65] R. K. P. Mok, E. W. W. Chan, X. Luo, and R. K. C. Chang, "Inferring the QoE of HTTP video streaming from user-viewing activities," in *Proc. 1st ACM SIGCOMM Workshop Meas. Up Stack*, Aug. 2011, pp. 31–36.

MATEUS SCHULZ NOGUEIRA (Member, IEEE) received the B.S. degree in computer science and the M.Sc. degree in informatics from the Federal University of Rio de Janeiro (UFRJ), in 2019 and 2022, respectively. His research interests include the modeling and analysis of systems, with focus on machine learning, recommendation systems, fairness, and covertness.

JOÃO ISMAEL DAMASCENO PINHEIRO received the master's degree in statistics from Stanford University, and the Ph.D. degree in informatics from the Federal University of Rio de Janeiro. His research interests include the statistical analysis of computer systems, performance evaluation, machine learning, and data analytics. He has coauthored a number of books, including *Estatística Básica: A Arte de Trabalhar com Dados*, that is adopted as textbook in a number of Brazilian universities.



FELIPE ASSIS is currently pursuing the Engineering degree in computer and information engineering with the Federal University of Rio de Janeiro (UFRJ). His research interests include the modeling and analysis of systems, with a focus on performance evaluation and industry-oriented applications.

DANIEL SADOC MENASCHÉ (Member, IEEE) received the B.S. and M.Sc. degrees, in 2003 and 2005, respectively, and the Ph.D. degree in computer science from the University of Massachusetts, Amherst, in 2011. Currently, he is an Associate Professor with the Computer Science Department, Federal University of Rio de Janeiro, Brazil. His research interests include the modeling, analysis, security, and performance evaluation of computer systems. He was a recipient of best paper awards at GLOBECOM 2007, CoNEXT 2009, INFOCOM 2013, and ICGSE 2015. He is an Alumni Affiliated Member of the Brazilian Academy of Sciences.

PAVLOS SERMPEZIS received the B.Sc. and Ph.D. degrees, in 2011 and 2015, respectively, the Diploma degree in electrical and computer engineering from the Aristotle University of Thessaloniki (AUTH), Greece, and the Ph.D. degree in computer science and networks from EURECOM, Sophia Antipolis, France. He was a Postdoctoral Researcher with FORTH, Greece. He is currently a Researcher with the Informatics Department, AUTH. His research interests include the modeling and performance analysis of communication networks, network measurements, and data science.

THRASYVOULOS SPYROPOULOS received the B.Sc. and Ph.D. degrees, in 2000 and 2007, respectively, the Diploma degree in electrical and computer engineering from the National Technical University of Athens, Greece, and the Ph.D. degree in electrical engineering from the University of Southern California. He was a Postdoctoral Researcher with INRIA. Then, he was a Senior Researcher with the Swiss Federal Institute of Technology (ETH) Zürich. He is currently an Assistant Professor with EURECOM, Sophia Antipolis, France. He was a recipient of the Best Paper Award in IEEE SECON 2008 and IEEE WoWMoM 2012. He was a Postdoctoral Researcher with INRIA, France, a Senior Researcher with ETH Zürich, and then a Professor with EURECOM. He is also a Professor with the Technical University of Crete, Greece. He was a recipient of the Best Paper Award in IEEE SECON 2008 and IEEE WoWMoM 2012.