HAL
archives-ouvertes.fr

Machine Learning Models for YouTube QoE and User Engagement Prediction in Smartphones

Sarah Wassermann, Nikolas Wehner, Pedro Casas

▶ **To cite this version:**

Sarah Wassermann, Nikolas Wehner, Pedro Casas. Machine Learning Models for YouTube QoE and User Engagement Prediction in Smartphones. Workshop on AI in Networks (WAIN) 2018, Dec 2018, Toulouse, France. hal-01898083

HAL Id: hal-01898083
https://hal.inria.fr/hal-01898083

Submitted on 18 Oct 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Machine Learning Models for YouTube QoE and User Engagement Prediction in Smartphones

Sarah Wassermann
Inria Paris
sarah.wassermann@inria.fr

Nikolas Wehner
University of Würzburg
Austrian Institute of Technology
nikolas.wehner.fl@ait.ac.at

Pedro Casas
Austrian Institute of Technology
pedro.casas@ait.ac.at

ABSTRACT

Measuring and monitoring YouTube Quality of Experience is a challenging task, especially when dealing with cellular networks and smartphone users. Using a large-scale database of crowdsourced YouTube-QoE measurements in smartphones, we conceive multiple machine-learning models to infer different YouTube-QoE-relevant metrics and user-behavior-related metrics from network-level measurements, without requiring root access to the smartphone, video-player embedding, or any other reverse-engineering-like approaches. The dataset includes measurements from more than 360 users worldwide, spanning over the last five years. Our preliminary results suggest that QoE-based monitoring of YouTube mobile can be realized through machine learning models with high accuracy, relying only on network-related features and without accessing any higher-layer metric to perform the estimations.

Categories and Subject Descriptors

I.5.4 [**Computing Methodologies**]: Pattern Recognition—*Applications*

Keywords

Machine Learning; Smartphone Measurements; QoE

The research leading to these results has been partially funded by the Vienna Science and Technology Fund (WWTF) through project ICT15-129, "BigDAMA".

1. INTRODUCTION

Quality of Experience (QoE) is becoming one of the leading concepts for network monitoring and performance evaluation in operational networks. The intensifying competition among cellular-network operators is forcing Internet Service Providers (ISPs) to integrate QoE into the core of their network-monitoring systems. ISPs need to offer high-quality services to reduce the risk of customer churn for quality dissatisfaction in a complex and bandwidth-restrictive context. Within this scenario, video streaming, and in particular YouTube-video streaming, represents the most challenging and relevant use case for QoE-based network monitoring and analysis.

Workshop on AI in Networks (WAIN) 2018 Toulouse, France
Copyright is held by author/owner(s).

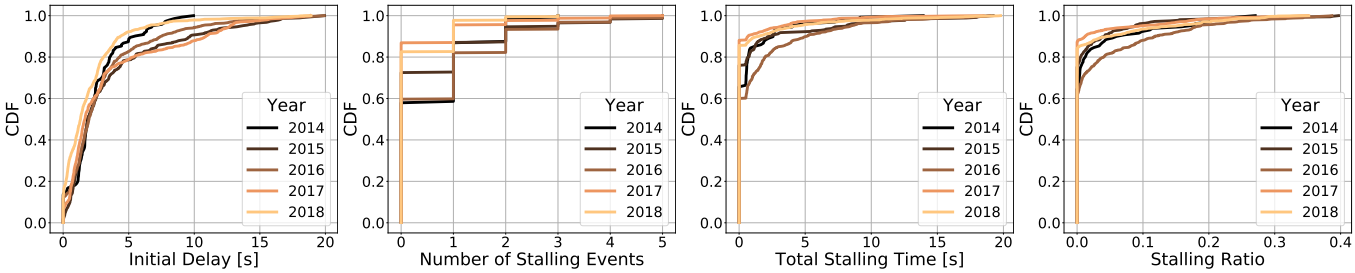
Passively measuring YouTube-QoE-relevant metrics such as stalling, quality changes, and initial delays is a challenging task, especially when dealing with smartphones and cellular networks. On the one hand, if the monitoring is done at the device level, it is not easy to access application-level metrics directly on the YouTube application without having root privileges on the phone or embedding a YouTube player in a different application [3]. On the other hand, if the monitoring is performed at the network level, the prevalence of end-to-end encryption turns previous in-network monitoring approaches inapplicable or highly inaccurate.

In this paper, we present a lightweight approach to predict YouTube-quality metrics based on features extracted from end-user smartphones running Android, by relying on simple metrics directly accessible through the Android APIs, such as the number of incoming and outgoing bytes, the signal strength, and the number of network switches. We rely on a crowdsourced database of more than 3,000 YouTube-video sessions monitored by *YoMoApp* [3], an Android application freely available on the Google Play Store. This app allows users to watch YouTube videos on an embedded YouTube video player while collecting a rich set of measurements such as traffic statistics, signal strength, and video-quality metrics like stalling events and video-quality switches. We conceive multiple machine-learning models to predict different QoE-relevant metrics only based on the Android-APIs-accessible measurements, notably those related to simple network-level features.

The remainder of this paper is organized as follows: Section 2 overviews the related work, focusing on machine-learning models for YouTube-QoE analysis. Section 3 describes the data used to build our models, briefly analyzing the five years of YoMoApp measurements which were collected between 2014 and 2018. Section 4 studies different machine-learning models to predict QoE-relevant metrics as well as end-user QoE and engagement. Finally, Section 5 concludes this work.

2. RELATED WORK

The literature provides an assorted list of tools to measure QoE-based network performance from network-device measurements: some examples include Mobilyzer [10] and Netalyzr [11]. QoE Doctor [12] measures mobile-app QoE, using active measurements at both application and network layers. Similar tools for YouTube monitoring on smartphones are presented in [13] and [14]. In [3,4], we introduced YoMoApp, an Android application to passively monitor YouTube-QoE-related features in mobile devices.



(a) initial delay (b) number of stallings (c) total stalling time (d) stalling ratio

Figure 1: Distribution of different video-quality metrics observed in the YouTube-mobile dataset.

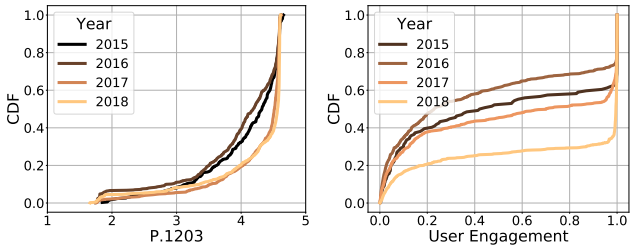
Previous work has also focused on the usage of machine-learning techniques to predict QoE for mobile applications: for example, the authors of [5] and ourselves in [8, 9] propose machine-learning-based approaches to evaluate mobile-app QoE using passive in-network and/or in-device measurements.

Due to the current trend towards end-to-end encryption, Deep-Packet-Inspection-based approaches cannot be applied as-is any longer. This has motivated a recent reappraisal of the YouTube-QoE-monitoring problem, with a surge of papers proposing the application of machine-learning models to infer or predict QoE-relevant metrics from network-level measurements [1, 2, 5–7]. These papers generally rely on packet-level features extracted from network measurements, which can be cumbersome to obtain if performed on the mobile device itself, or providing "limited visibility" (i.e., rather poor prediction performance) when conducted on the network side.

3. YOMOAPP DATA ANALYSIS

The dataset we study consists of more than 3,000 YouTube video sessions collected worldwide over 70 different cellular ISPs and from more than 360 different users, from 2014 until today. These users are scattered throughout multiple regions of the world, with most of them in Europe (Germany, France, Austria, and the UK), but also with some residing in the US. Measurements are collected with our YoMoApp tool. The goal of YoMoApp is to provide a distributed, crowdsourcing-based monitoring platform that gathers user feedback and application-layer Key Performance Indicators (KPIs) of YouTube mobile, which have a high correlation with the QoE undergone by the YouTube-mobile-app users. KPIs such as initial delay, stallings, and quality adaptations are the most relevant QoE-related features measured by YoMoApp. These are passively collected during the playback of a video session from the state and buffer of the video player, as well as from the resolution of the played-out video. Measurements performed on each device are locally logged and periodically exported to a cloud server.

Besides the monitoring of the playback, network and context parameters are also retrieved by YoMoApp. Several device characteristics and their changes, namely screen size, screen orientation, volume, player size, and player mode (normal/full screen), are monitored. Network usage is also logged. The amount of downloaded and uploaded bytes on the device (i.e., for all running applications), on the mobile network, and only for YoMoApp are polled every second. Moreover, changes of operator, RAT, cell ID, signal strength, and GPS position are also collected.



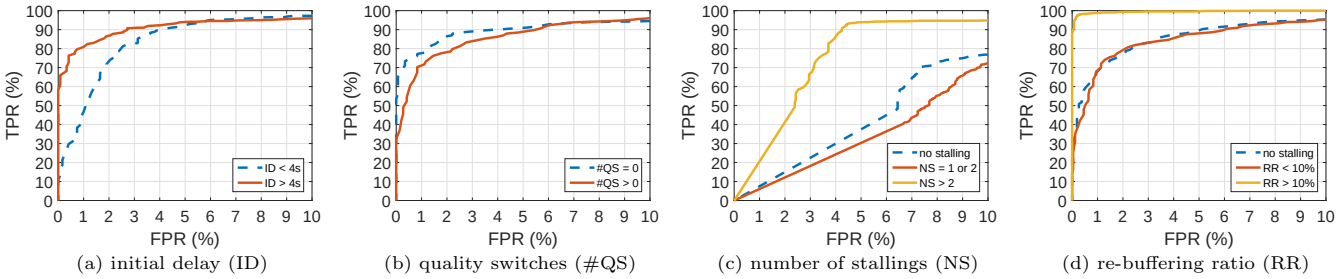
(a) MOS scores (P.1203 model). (b) User engagement.

Figure 2: MOS scores and user engagement.

Figure 1 depicts the distribution of four key QoE-relevant metrics for adaptive video streaming which we consider for the study, split by year: initial playback delay, number of stalling events, total stalling time, and re-buffering/stalling ratio. Besides a characterization of the measurements, we also focus on their properties along time, to better understand the inherent dynamics behind QoE in YouTube on the move.

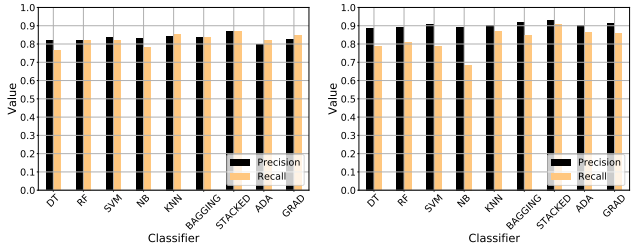
The first interesting observation is that, excluding 2014 and 2015 which had a smaller number of sessions, one can clearly appreciate an improvement over time on all the QoE-relevant metrics, with 2018 sessions showing the smallest initial delays and best performance in terms of number of stalling events. As of 2018, more than 90% of the video sessions experienced an initial playback delay below 5 seconds, and almost 90% of the sessions played smoothly without re-buffering events. In contrast, the initial delay for video sessions in 2016 was below 5 seconds for 80% of the sessions, and only 60% of the 2016 sessions experienced no stallings. When considering highly-QoE-impaired video sessions, we see that more than 12% of the video sessions in 2016 had a re-buffering ratio above 10%, whereas this number reduced to about 5% in 2017 and 2018.

Figure 2 reports the distribution of (a) an estimation of the QoE experienced by our users by applying a standardized QoE model – P.1203 [16] – and (b) user engagement. Recall that QoE is provided in terms of MOS scores, using a 5-level absolute category rating (ACR) scale [15]. User engagement is defined as the fraction of the total video length a user watched, before the video was aborted or ended (in particular, the user engagement is equal to 100% in case she watched the entire video). There has been a clear im-



(a) initial delay (ID) (b) quality switches (#QS) (c) number of stallings (NS) (d) re-buffering ratio (RR)

Figure 3: QoE-metrics prediction performance. ROC curves highlight a high recall for the considered classes.



(a) User engagement. (b) P.1203 MOS.

Figure 4: Prediction of user engagement and P.1203 MOS.

provement in terms of QoE and user engagement in 2017 and 2018, with about 80% of the sessions being rated as very good or excellent (MOS 4 or 5), in contrast to the 40% to 60% reported in the previous years. User engagement has systematically increased over time, and in particular in 2018: more than 60% of the videos were watched completely and only 20% of the users aborted the video at 20% or less of the video playback.

4. QOE METRICS AND USER ENGAGEMENT PREDICTION

Based on the previously described metrics, we now focus on the prediction of QoE-relevant metrics which are normally measured directly by YoMoApp. However, in our particular prediction setting, we assume that we only have access to the network-level measurements, as these are available through the Android APIs. The rationale here is that we would like to monitor and infer YouTube-mobile KPIs such as initial delay, stalling, quality switches, QoE (MOS scores), and user engagement, but without using an application like YoMoApp. These predictors could be applied in a more generic smartphone-based monitoring system, where users would not be forced to run an embedding app such as YoMoApp to measure relevant KPIs, and where such KPIs could actually be forecast for any user watching YouTube videos on her smartphone, independently of the YouTube player being used.

In the context of this work, we focus on the prediction of four QoE-relevant metrics, of the MOS scores (as provided by the P.1203 model), and of the user engagement. For all these prediction tasks, we rely on the network-layer features captured by YoMoApp. Predictors are built using machine-learning models, treating each problem as a classification task, where targets are discretized. The reason why we choose to address our prediction tasks as classification problems is that ISPs are more interested in the overall performance of their services than in exact numbers, i.e. want to mainly know whether a given KPI is above/below a certain threshold or not. The four QoE-relevant targets are as follows: *(i)* whether initial delays (ID) are above or below a pre-defined QoE-relevant threshold – based on the analysis of our dataset and previous work on initial-delay tolerance, we set this value to four seconds; *(ii)* whether a video-quality switch has occurred during the session or not; *(iii)* the number of stalling events (NS), considering three classes – *zero-stalling*, *mild-stalling*: one or two stalling events, and *severe-stalling*: more than two stallings; and *(iv)* the stalling frequency or re-buffering rate (RR), considering again three classes – *stalling-free*; *mild-stalling*: stallings occurred and lasted for less than 10% of the total duration of the video session, and *severe-stalling*: stallings occurred for at least 10% of the whole video session. For the prediction of QoE scores, we use as target a binary discretization of the MOS scores provided by the P.1203 model, and consider a two-classes classification problem, either better or worse than MOS = 4 - i.e., using good quality as threshold. Finally, we turn the prediction of user engagement into a three-classes classification problem, predicting whether a user has watched less than 50% of the video, between 50% and 70%, or more than 70%.

The full feature set encompasses 275 features, including information about the received signal strength, the number of handovers, the number of network switches, and multiple statistics about the incoming and outgoing traffic, aggregated at different time windows lasting 1 second, 5 seconds, 10 seconds, 30 seconds, and 60 seconds. For each metric, we evaluate a 10-tree random-forest model through 10-fold cross-validation. We rely on simple bootstrapping techniques to balance classes for learning purposes.

We use feature-selection techniques to identify the most relevant features for each prediction target. In particular, we rely on a wrapper approach, which ranks features based on their prediction power for a specific prediction model – in this case, a 10-tree random forest. We find that about 30 features out of the 275 are needed to obtain accuracies highly similar to the ones achieved with the full feature set. In particular, features derived from the received and transmitted traffic are very important. Indeed, statistics about the traffic by the application itself and by the device are among the top features, including metrics describing the variation of the throughput across multiple time-window lengths. Metrics related to changes in the signal strength are also among the selected features. Interestingly, the number of handovers

does not seem to play an important role for the QoE-metrics inference in our case.

Figure 3 reports the obtained results for the prediction of the four QoE-relevant KPIs prediction in terms of ROC curves. ROC curves help understand the performance of binary-classification models at all classification thresholds and show the different false positive rates (FPRs) and true positive rates (TPRs). Our results are fairly accurate for the four prediction targets, achieving good classification rates for most of the classes. For example, the initial delay discrimination as well as the quality-switching detection can be done with a false positive rate below 5% for more than 90% of the sessions. Results are even better when predicting the re-buffering ratio, with an almost perfect performance for detecting bad-quality sessions with a high stalling ratio. Inferring the number of stalling events is clearly more challenging than that of the other three targets.

For the prediction of user engagement and MOS scores, we also consider random forests, but additionally evaluate other models such as a single decision tree (DT), SVM, k-nearest neighbors (KNN), and Naïve Bayes (NB). We additionally consider ensemble learning approaches, covering the three basic paradigms available in the ensemble-learning domain: bagging, boosting (AdaBoost (ADA) and gradient boosting (GRAD)), and stacking. Rather than finding the best model to explain the data, ensemble-learning methods build a set of models and combine them, seeking complementarity among models. Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. An ensemble of models also exhibits higher robustness with respect to uncertainties in training data, which is highly beneficial for generalization of results.

Figure 4 summarizes the obtained results in terms of precision and recall for all the tested models, obtained through 10-fold cross-validation. As before, high prediction performance can be achieved for both targets, particularly when using more complex, ensemble-learning-based approaches, like stacked trees (STACKED). Prediction of P.1203 MOS classes and user-engagement discrimination can be realized with an overall accuracy of around 90%.

5. CONCLUSION

In this paper, we have studied the problem of YouTube-mobile-QoE monitoring and analysis through machine-learning models, by relying on a large-scale dataset of QoE measurements passively collected on users' smartphones with the YoMoApp monitoring application. We conceived multiple machine learning models to infer different YouTube-QoE-relevant metrics and user-behavior-related metrics from network level measurements only. We observed an outstanding performance of random-forest models to predict the QoE of the end-users. We have also shown the advantages of ensemble-learning techniques with respect to simpler models in terms of prediction accuracy, and particularly of stacking models, when it comes to the inference of MOS scores and user engagement.

The presented models could enable a broader, non-intrusive, and privacy-preserving approach for large-scale, QoE-based monitoring of YouTube in mobile devices, as they could be directly applied without accessing any higher-layer metric to perform the estimations.

6. REFERENCES

[1] V. Krishnamoorthi et al., "BUFFEST: Predicting Buffer Conditions and Real-Time Requirements of HTTP(S) Adaptive Streaming Clients," in *MMSys*, 2017.

[2] G. Dimopoulos et al., "Measuring Video QoE from Encrypted Traffic," in *IMC*, 2016.

[3] F. Wamser et al., "Understanding YouTube QoE in Cellular Networks with YoMoApp – a QoE Monitoring Tool for YouTube Mobile," in *MobiCom*, 2015.

[4] ——, "YoMoApp: A tool for analyzing QoE of YouTube HTTP adaptive streaming in mobile networks," in *EuCNC*, 2015.

[5] V. Aggarwal et al., "Prometheus: Toward quality-of-experience estimation for mobile apps from passive network measurements," in *HotMobile*, 2014.

[6] I. Orsolic et al., "A machine learning approach to classifying YouTube QoE based on encrypted network traffic," Media Tools and Apps., vol. 76(21), 2017.

[7] M. H. Mazhar et al., "Real-time video quality of experience monitoring for HTTPS and QUIC," in *INFOCOM*, 2018.

[8] P. Casas et al., "Predicting QoE in cellular networks using machine learning and in-smartphone measurements," in *QoMEX*, 2017.

[9] P. Casas et al., "Next to You: Monitoring Quality of Experience in Cellular Networks From the End-Devices," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 181–196, June 2016.

[10] A. Nikravesh et al., "Mobilyzer: An Open Platform for Controllable Mobile Network Measurements," in *MobiSys*, 2015.

[11] C. Kreibich et al., "Netalyzr: Illuminating the edge network," in *IMC*, 2010.

[12] Q. A. Chen et al., "QoE Doctor: Diagnosing Mobile App QoE with Automated UI Control and Cross-layer Analysis," in *IMC*, 2014.

[13] I. Ketykó et al., "QoE Measurement of Mobile YouTube Video Streaming," in *MoViD*, 2010.

[14] G. Gómez et al., "YouTube QoE Evaluation Tool for Android Wireless Terminals," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 164, pp. 1–14, 2014.

[15] International Telecommunication Union, "ITU-T Recommendation P.800: Methods for Subjective Determination of Transmission Quality," 1996.

[16] International Telecommunication Union, "ITU-T Recommendation P.1203: Parametric Bitstream-based Quality Assessment of Progressive Download and Adaptive Audiovisual Streaming Services over Reliable Transport," 2016.