Let me decrypt your beauty: real-time prediction of video resolution and bitrate for encrypted video streaming

Sarah Wassermann, Michael Seufert, Pedro Casas, Li Gang, Kuang Li

Angaben zur Veröffentlichung / Publication details:

Wassermann, Sarah, Michael Seufert, Pedro Casas, Li Gang, and Kuang Li. 2019. "Let me decrypt your beauty: real-time prediction of video resolution and bitrate for encrypted video streaming." In *2019 Network Traffic Measurement and Analysis Conference (TMA), 19-21 June 2019, Paris, France*, edited by Stefano Secci, Isabelle Chrisment, and Marco Fiore, 199–200. Piscataway, NJ: IEEE. https://doi.org/10.23919/tma.2019.8784589.

Let me Decrypt your Beauty: Real-time Prediction of Video Resolution and Bitrate for Encrypted Video Streaming

Sarah Wassermann*, Michael Seufert†, Pedro Casas*, Li Gang‡, Kuang Li‡
*AIT Austrian Institute of Technology, Vienna, Austria
†University of Würzburg, Würzburg, Germany
‡Huawei Technologies, Department of R&D, Shenzhen, P.R. China

***Abstract*—The dynamic adaptation of the video quality induced by HTTP Adaptive Streaming (HAS) technology introduces new Quality of Experience (QoE) metrics beyond re-buffering. In this work we address the problem of real-time QoE monitoring of HAS, focusing on the continuous prediction of video resolution and average video bitrate, for the particular case of YouTube. Through empirical evaluations over a large video dataset, we demonstrate that it is possible to accurately predict the specific video resolution, as well as the average video bitrate, both in real time, and using a time granularity as small as one new prediction every second, not achieved by other proposals in the literature.**

I. Introduction

Video streaming is one of the key applications of the Internet. To satisfy end users and avoid customer churn, Internet Service Providers (ISPs) strive to deliver a high video streaming Quality of Experience (QoE). With HTTP Adaptive Streaming (HAS), degradations such as re-buffering and high initial delays have been partially mitigated, by dynamically adapting the video bitrate to the network conditions. To modify the video bitrate, the visual quality level of the streamed video has to be changed, e.g., in terms of resolution, frame rate, or compression, which introduces an additional impact on QoE. ISPs are therefore highly interested in monitoring solutions able to immediately detect events when the displayed quality level drops, to take appropriate countermeasures. The trend towards end-to-end encryption (e.g., HTTPS), however, has significantly reduced the visibility of network operators on the traffic of their customers, making the monitoring process more challenging and cumbersome.

In [1], [2] we have recently introduced ViCrypt, an on-line machine-learning based system to predict re-buffering events in YouTube encrypted traffic. ViCrypt analyzes ongoing streaming sessions using fine grained time slots of one-second length, computing multiple traffic-level features in a stream-like fashion, using different temporal aggregations of past measurement slots. More precisely, ViCrypt considers features extracted from the most recently ended time slot – *snapshot* features, features aggregating the last t time slots – *trend* features, and features aggregating all past time slots since the start of the streaming session – *progressive* features. At the end of each new time slot, all these features are fed into machine-learning models, which predict the occurrence of re-buffering and permits to reconstruct the full (re-)buffering pattern with low temporal resolution. Here we extended ViCrypt to additionally predict the video resolution and average bitrate in real-time, using the same fine-grained time granularity of one second. *To the best of our knowledge, this is by now the finest time-granularity for real-time prediction of YouTube QoE over encrypted traffic.*

To demonstrate the performance of ViCrypt in these new prediction tasks, we would show how the complete approach computes the aforementioned features in real-time, and how the proposed models provide accurate predictions during the course of an ongoing YouTube video streaming session.

II. Methodology & Evaluation Dataset

ViCrypt relies on 208 features extracted from the current and from past time slots. The usage of fine-grained, one second time slots represents a good trade-off between prediction delay and accuracy. Trend features contain information about the last $t = 3$ time slots, whereas progressive ones capture statistics describing all the traffic encountered during the sessions so far. ViCrypt uses features such as number of total, uplink, and downlink packets, the amount of transferred bytes (total, uplink, downlink), and time-based features, including the time from the start of the slot until the first packet and the burst duration, i.e., the time between the first and last packets of the time slot. The entire feature set is computed in an on-line fashion with constant memory consumption, without the need to store the previous traffic or detailed information about past packets. As such, the approach has minimal memory footprint, and can run in constrained hardware equipment.

To train and evaluate the specific ViCrypt machine learning models for the new proposed predictions targets, we built a dataset composed of about 15,000 YouTube video sessions streamed over a period of several months from June 2018 to February 2019. The dataset includes a total number of 4,672,719 one-second time slots. Ground truth is directly captured at the YouTube player side, using Java-based monitoring tools. We use Selenium browser automation to run Chrome browsing and stream random videos from YouTube. To conceive generalizable models, videos were streamed with highly diverse network characteristics, using home and corporate WiFi networks, as well as LTE mobile networks. Videos sessions cover both QUIC and TCP transport. For each video streaming session, packet sizes and arrival times are collected, as well as DNS lookup responses to obtain a mapping between IP addresses and domain names serving YouTube contents. Finally, also the recently published open

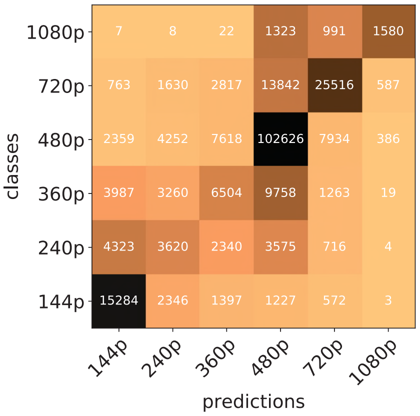

Fig. 1. Confusion matrix for kNN-based video resolution prediction.

dataset [3] was considered, which consists of YouTube video streaming network and application measurements, related to the usage of the native Android YouTube app.

Regarding model training, validation, and testing, 80% of the video sessions – randomly selected, were used for model training and validation, while the remaining 20% of the sessions composed the independent testing set. In YouTube, the video resolution is typically indicated by the amount of vertical pixels, for which standard quality classes exist. The classes contained in the dataset are 144p, 240p, 360p, 480p, 720p, and 1080p. Distribution analysis shows that most of the videos were streamed in 480p (55% of time slots), but also very low resolutions occur (9% 144p, 6% 240p, 10% 360p). At the other end, HD resolution accounts for about 20% of the time slots (18% 720p, 2% 1080p).

The average bitrate was obtained via the YouTube API and represents the average bitrate of the whole video when streamed with a given quality level (itag). Thus, this prediction target is not the momentary bitrate of the current slot, but rather the average bitrate of the quality level that was downloaded in the current slot. Two different models were trained on the datasets, namely random forests with 10 trees (RF10) and k-nearest neighbors with $k = 1$ and $k = 3$.

The first prediction target is the video resolution, which is highly linked to the visual quality of the streamed video, and is therefore a major QoE-relevant metric. The estimation of the video resolution is handled as a **classification** problem. Similar to [6], the considered classes correspond to the typically observed YouTube video resolutions, resulting in a substantially more granular prediction than other approaches so far studied in the literature [4], [5]. The second target to predict is the corresponding average video bitrate, which is very relevant to track for proactive network management. As the bitrate is per-se continuous, the estimation of the average video bitrate of the currently played video resolution is tackled as a **regression** problem.

III. Performance Evaluation

ViCrypt achieves fairly accurate prediction results for both targets. For the video resolution prediction, both RF10 and kNN with $k = 3$ yield promising results. Nevertheless, we noticed that, even though RF10 has a higher accuracy – 71% correct predictions vs. 66% for kNN, the kNN model outputs accurate predictions for a non-negligible number of samples among all the different classes, whereas the correct predictions of RF10 are almost all in the 480p and 144p classes; the confusion matrix for kNN is depicted in Fig. 1. While HD contents such as 720p tend to be underestimated, lower resolution videos tend to be overestimated. The highly imbalanced classes are a problem we are trying to solve to provide more robust models and better prediction results.

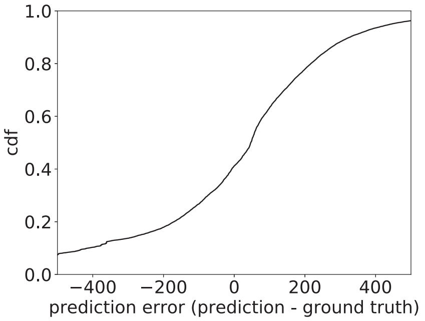

Fig. 2. CDF of differences between actual and RF-predicted average bitrate.

For the bitrate estimation, we predict the average bitrate of the video which was downloaded during the one-second time slot. ViCrypt achieves again encouraging results with both models RF10 and kNN, with $k = 1$ this time. Contrary to the resolution prediction, RF10 outperforms kNN for this task. Indeed, RF10 yields an absolute error of at most 400 kbps for 83% of the time slots, while this holds only for about 74% of the timeslots when using kNN. ViCrypt-RF10 achieves a mean absolute error (MAE) of only 233 kbps, while kNN has a higher MAE of 305 kbps. The CDF for RF10 prediction errors is depicted in Fig. 2. ViCrypt overestimates the actual bitrate for a significant fraction of the time slots with RF10 (59%), while kNN is very balanced between over- and underestimation. This overestimation of RF10 is advantageous from the point of view of the ISP, as overestimating the video bitrate helps them avoid allocating insufficient bandwidth in the context of traffic shaping. This could cause the video to stall, which is a major QoE degradation.

References

[1] M. Seufert, P. Casas, N. Wehner, L. Gang, and K. Li, "Stream-based Machine Learning for Real-time QoE Analysis of Encrypted Video Streaming Traffic," in *QoE-Management 2019*, 2019.

[2] M. Seufert, P. Casas, N. Wehner, L. Gang, and K. Li, "Features that Matter: Feature Selection for On-line Stalling Prediction in Encrypted Video Streaming," in *IEEE INFOCOM Workshops, 2nd International Workshop on Network Intelligence*, Paris, France, 2019.

[3] T. Karagkioules, D. Tsilimantos, S. Valentin, F. Wamser, B. Zeidler, M. Seufert, F. Loh, and P. Tran-Gia, "A Public Dataset for YouTube's Mobile Streaming Client," in *IEEE/IFIP MNM Workshop 2018*, Vienna, Austria, 2018.

[4] I. Orsolic, D. Pevec, M. Suznjevic, and L. Skorin-Kapov, "YouTube QoE Estimation Based on the Analysis of Encrypted Network Traffic Using Machine Learning," in *IEEE QoEMC Workshop 2016*, Washington, DC, USA, 2016.

[5] M. H. Mazhar and M. Z. Shafiq, "Real-time Video Quality of Experience Monitoring for HTTPS and QUIC," in *IEEE INFOCOM*, USA, 2018.

[6] C. Gutterman, K. Guo, S. Arora, X. Wang, L. Wu, E. Katz-Bassett, and G. Zussman, "Requet: Real-time qoe detection for encrypted youtube traffic," in *Proceedings of the 10th ACM Multimedia Systems Conference, MMSys 2019*, 2019.