

Mobile Web and App QoE Monitoring for ISPs -from Encrypted Traffic to Speed Index through Machine Learning

Pedro Casas, Sarah Wassermann, Nikolas Wehner, Michael Seufert, Joshua Schüler, Tobias Hossfeld

▶ **To cite this version:**

Pedro Casas, Sarah Wassermann, Nikolas Wehner, Michael Seufert, Joshua Schüler, et al.. Mobile Web and App QoE Monitoring for ISPs -from Encrypted Traffic to Speed Index through Machine Learning. 13th IFIP Wireless and Mobile Networking Conference (WMNC), Oct 2021, Montréal, Canada. hal-03365897v2

HAL Id: hal-03365897
https://hal.science/hal-03365897v2

Submitted on 30 Oct 2021

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.

Mobile Web and App QoE Monitoring for ISPs – from Encrypted Traffic to Speed Index through Machine Learning

Pedro Casas*, Sarah Wassermann*, Nikolas Wehner†
Michael Seufert†, Joshua Schüler†, Tobias Hossfeld†
*AIT Austrian Institute of Technology, †University of Würzburg

***Abstract*—Web browsing is one of the key applications of the Internet. In this paper, we address the problem of mobile Web and App QoE monitoring from the Internet Service Provider (ISP) perspective, relying on in-network, passive measurements. Our study targets the analysis of Web and App QoE in mobile devices, including mobile browsing in smartphones and tablets, as well as mobile apps. As a proxy to Web QoE, we focus on the analysis of the well-known Speed Index (SI) metric. Given the wide adoption of end-to-end encryption, we resort to machine-learning models to infer the SI of individual web page and app loading sessions, using as input only packet level data. Empirical evaluations on a large, multi mobile-device corpus of Web and App QoE measurements for top popular websites and selected apps demonstrate that the proposed solution can properly infer the SI from in-network, encrypted-traffic measurements, relying on learning-based models. Our study also reveals relevant network and web page content characteristics impacting Web QoE in mobile devices, providing a complete overview on the mobile Web and App QoE assessment problem.**

***Index Terms*—Web QoE; Mobile Devices; Apps; Speed Index; Network Monitoring; Machine Learning; Encrypted Traffic.**

I. Introduction

Web browsing is the most important Internet service for the end user; in fact, most services and applications are offered today through the web. The performance of a web service as perceived by the end user can be measured by the corresponding web browsing Quality of Experience, or Web QoE; from a practical perspective, reliably measuring Web QoE is challenging. Different from other specific services, such as video streaming, web browsing is a mix of multimedia contents and embedded services; loading a single web page requires to download tens of contents from different servers and content providers. In this complex process, the network can significantly impact users' experience, forcing ISPs to deploy effective means to monitor their customers' Web QoE.

The literature on Web QoE analysis is rife with objective metrics capturing the performance of web pages, including metrics such as Page Load Time (PLT), Speed Index (SI), Above the Fold Time (AFT), First Input Delay (FID), etc. However, all these metrics require access to the application layer, which is hidden from the eyes of the ISP by the wide deployment of end-to-end network traffic encryption. In [3], [4], authors showed the potential of using Machine Learning (ML) to infer Web QoE metrics from encrypted network traffic, for the specific scenario of desktop web browsing. By using page load controlled experiments, where network data is simultaneously collected with ground-truth Web QoE metrics such as SI, AFT, etc., their study built a labeled dataset and trained supervised ML models to infer these QoE-related metrics from network traffic features, computed on the stream of collected bytes. In [1], we showed that the performance of these ML models built on desktop measurements does not generalize to mobile browsing, resulting in poor Web QoE inference performance when applied to web browsing in smartphones. Finally, in [2] we have recently introduced a novel measurement framework to measure loading times in apps, which can be used to extend the analysis of Web QoE to those mobile apps where perceived waiting times determine the experience of the end-user – e.g., different from video streaming, where re-buffering and video quality are the key to user experience. We would therefore refer to Web and App QoE from now on, meaning the access of remote web contents from browsers or mobile apps.

In this paper we follow a similar approach to [1], [2], focusing the analysis on Web and App QoE in mobile devices exclusively. Web and App QoE is paramount for mobile ISPs, as the lion's share of Internet-access devices is today smartphones, with nearly three quarters of the world population using just their smartphones to access the Internet by 2025 [5]. *Our contributions are as follows*: **(i) Data:** we generate a unique dataset of Web and App QoE measurements in mobile devices, targeting the most popular websites in today's Internet, and a selected set of popular mobile apps. The dataset includes both application-layer Web QoE metrics – such as SI, as well as network traffic traces, for more than 40,000 web and app *loading sessions* (i.e., the loading of a single browser or app page). **(ii) Models:** leveraging these data, we present an extensive benchmark comparing the performance of different ML models to infer the SI of web browsing and app sessions in mobile devices, considering models for web browsing in smartphones, tablets, and for apps. **(iii) Mobile Web QoE insights:** we characterize the contents and properties of the targeted websites, unveiling how relevant network and web page characteristics impact mobile Web QoE.

The remainder of the paper is organized as follows. Sec. II overviews the related work on Web and App QoE monitoring and analysis. Sec. III presents the data generation and overall modeling/inference approach, including a characterization of the generated datasets. In Sec. IV we introduce and evaluate the proposed ML models for mobile Web QoE inference, using as input features derived from the encrypted streams

ISBN 978-3-903176-42-3 © 2021 IFIP

of network traffic. Inference assessment is extended to apps in Sec. IV-C, including the evaluation of a single, consolidated model for multi-device mobile Web and App QoE inference. In Sec. V, we dig deeper into the web measurements to understand the correlations and implications of different web page content and network characteristics on mobile Web QoE. Finally, Sec. VI concludes this paper.

II. Related Work

First Web QoE models described in the literature were based on plain Page Load Times (PLT) [6], [7], and are still broadly used in the practice to infer user satisfaction in web-browsing [8]. However, research in the field demonstrated that PLT is not the most accurate proxy to user perception of web page loading times, as the actual web content visible to the user is usually displayed much earlier, because most web pages often stretch beyond the browser's viewport. Additional in-browser metrics have been devised to better suit the page display on the screen. An approach is the so-called Above the Fold Time (AFT), i.e., the time until the visible portion of a web page has been fully loaded, which has also been tested in traditional Web QoE models [9]. Newer Web QoE metrics have been proposed recently, such as the SI, which takes into account the whole visual progress of the page loading, by processing a video capture of the screen. Besides single metric modeling, ML-based approaches have been presented [10], [11] to model Web QoE from a combination of metrics.

Another direction in the literature proposes to understand how external components influence Web QoE. In [12], [13], the impact of network quality fluctuations and outages on user Web QoE was studied. Other components besides network quality influence Web QoE, linked to the specific web page content – usability [14], aesthetics [15], etc., as well as device type – desktop, smartphone, tablets [1], [16]. Important to our study, these papers show that smartphones and tablets have their own characteristics, not only regarding screen sizes but also in terms of content rendering and web designs.

Most of these papers are based on the analysis of Web QoE in lab, controlled environments. Others directly rely on in-browser metrics as a proxy to Web QoE, conducting large-scale active measurement campaigns. For example, the impact of multiple features such as transport protocols, network connections, visible portion, etc., on PLT and AFT is studied in [17], based on a set of 244 million measurements collected during 6 months for the top-10000 Alexa websites. Other papers also measured the impact of similar features on PLT and SI or AFT in different countries and different types of network [18], including mobile networks [19].

While useful, **most of prior work stays at the application-level**. **This is problematic for ISPs, which have no direct access to in-browser metrics, but only to network traffic**. In recent years, TLS encryption has even narrowed the information that ISPs can collect from the network side, and previous approaches based on DPI and HTTP traffic analysis such as [20] are no longer applicable. Other papers [21], [22] have developed metrics with high correlation to the SI metric,

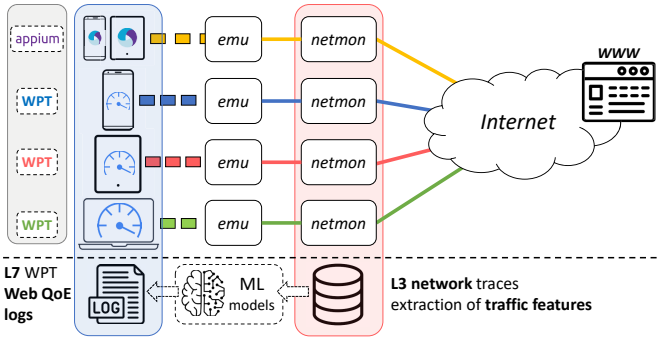


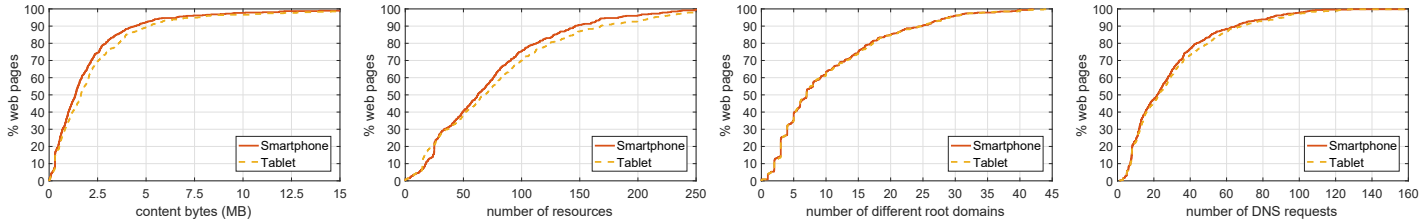
Fig. 1: Measurement platform for model calibration.

such as Byte/Object-Index [21] and Pain-Index [22], which can be computed from packet- and flow-level measurements statistics, thus seamlessly operating with encrypted traffic. Still, such metrics are mostly informative, as they do not provide an absolute estimation of the actual SI value, which is the key for Web QoE analysis. Recent work [3], [4] takes a step further to directly infer the SI metric, using ML techniques mapping network (encrypted) traffic features to SI. In [1], we complemented [3], [4] and showed that the performance of previously proposed ML models built on desktop measurements does not generalize to mobile browsing, resulting in poor Web QoE inference performance when applied to web browsing in smartphones.

In the specific case of mobile Web and App QoE monitoring, there have been multiple papers using ML [23], [24], [25], [26] or simple modeling approaches [27] to map application layer metrics [25], [26] or network QoS metrics [23], [24], [27] into QoE-related metrics. From these, [23], [24] are the closest to our work, but either propose analysis approaches which are no longer applicable due to traffic encryption [24], or do not address the web browsing scenario [23]. Regarding App QoE measurement technology, the task is far from trivial, given all the complexities associated to the instrumentation of QoE measurement in mobile devices e.g., lack of APIs for measuring QoE-relevant metrics, lack of open measurement tools for mobile, a vast heterogeneity of different apps, just to name a few of them; in this direction, platforms presented in [27], [2], [28] provide means to semi-automate App QoE measurement, in a per-app instrumentation basis. Here we use our platform [2] to study the QoE of four popular apps (Amazon, YouTube, Facebook, and BBC News), splitting the analysis of each app loading session into different *user interactions*, including the starting of the app, clicking different in-app pages or links, scrolling through the app screen, etc.

III. Web and App QoE Datasets & Modeling

The proposed solution to the mobile Web and App QoE monitoring problem consists of training supervised ML models to map network traffic features, extracted from the encrypted network web-page traffic, into relevant Web QoE metrics. The approach is data-driven, and thus needs datasets containing both the collected traffic traces – the *input*, and the targeted Web QoE metric – the *ground truth*. To fully control the



(a) Web page bytes. (b) Web page resources. (c) Root domains. (d) DNS requests.

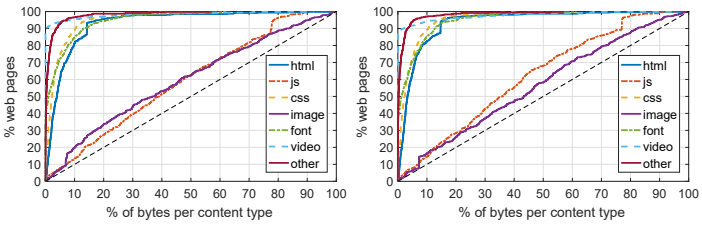
Fig. 2: Characterization of the web pages in terms of size, resources, and requested domains.

generation of such datasets, we conceived a measurement platform and testbed based on multiple private instances of WepPageTest (WPT) [30], a well-known and widely used open-source web performance analysis tool. Different from previous studies [17], [3], [4], [21], [18], [19], which have focused exclusively on desktop browsers and desktop devices (or in some exceptional cases, browser-emulated mobile devices), our measurement testbed consists of three different, non-emulated types of devices, including smartphones, tablets, and desktop (Chrome is used as browser), using WPT agents for Android and Linux. Using WPT measurements, the platform extracts about 90 different KPIs and Web QoE metrics from independent web page loading sessions – such as PLT, SI, AFT, Byte Index [21], Object Index, Time to Interactive (TTI), etc., as well as content characteristics of the visited web pages. From these metrics we selected the SI as target, which is today one of the most accepted metrics reflecting Web QoE. Nevertheless, the methodology applies to any other similar Web-QoE metric. Network traffic is captured at an intermediate monitoring device and stored as .pcap traces, which are post-processed to extract the input features to the models. In this paper we focus exclusively on the data generated in mobile devices, including smartphones and tablets.

The testbed is extended to also run controlled tests on apps through Appium (https://appium.io), a popular open source test automation framework for automating native, hybrid and mobile web apps. Using Appium, we built a test automation framework which allows the analysis of independent user interactions with a particular app, producing for each of these actions a separate traffic capture trace and a screen capture video, the latter used to automatically extract the corresponding SI metric through frames' analysis (see https://github.com/WPO-Foundation/visualmetrics).

Fig. 1 shows the components of the measurement platform. Devices are connected to the open Internet through independent network emulators, which allows for controllable network access performance configurations in terms of bandwidth, latency, packet loss, etc. This allows for heterogeneity in the generated measurements. Configurations used in the study include access downlink bandwidth up to 10 Mbps, packet loss rates up to 10%, and RTTs up to 100ms.

Web browsing measurements target the top 500 websites according to Alexa top sites list. The same web pages are visited multiple times for each device type, under the same access network setups. We do not consider the effect of



(a) Smartphone. (b) Tablet.

Fig. 3: Shares of web page contents (bytes).

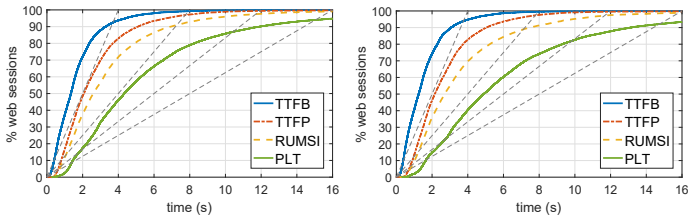
caching, thus all tests correspond to a first-view loading session. In the case of web browsing, we collect the so-called RUM SpeedIndex (RUMSI) metric [31], which is a passive approximation to the SI, computed from the analysis of web page resource timings. RUMSI provides the same information as SI, but does not require screen capture at the end device.

Given that app measurement requires instrumentation for each specific app, we selected four popular apps for the study, including Amazon, YouTube, Facebook, and BBC News. The main idea of this selection is to test different app technologies – e.g., Facebook and YouTube are both native apps, BBC News is a web app, and Amazon is a hybrid app, with different levels of interactivity in terms of user actions. We tested different kinds of interaction, including app startup (i.e., start the app, and wait for the main page to load), page scrolling, search, and menu items/links clicking; in all cases, the SI is measured from the time of the action execution (0% visual progress), till the completion of the resulting change in the screen (100% visual progress). For simplicity, the app startup is tested for all four apps, and the rest of the user interactions is only tested for Amazon and BBC News. The resulting dataset consists of more than 40,000 web page and app loading sessions.

Finally, for the sake of simplicity and to keep the scope of the study, we assume that the monitoring system takes as input network traffic from a single web or app session. In the case of concurrent web sessions, we have conceived a classification methodology similar to the one used in [22] to disentangle them; the traffic filtering and classification is nevertheless out of the scope of this paper.

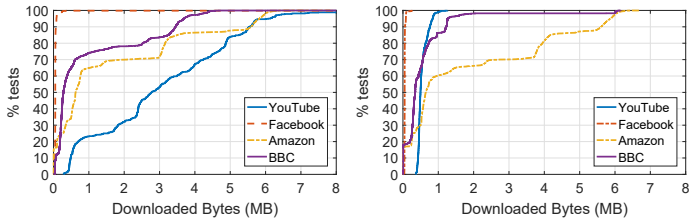
A. Web Data Characterization

The list of top 500 Alexa web pages is very assorted in terms of contents, and as we show next, there is a mild yet visible impact in terms of web page characteristics and timing performance regarding the type of mobile device being used.



(a) Smartphone. (b) Tablet.

Fig. 4: Time performance of web loading sessions.



(a) Smartphone. (b) Tablet.

Fig. 5: Downloaded bytes per app, multiple actions.

As expected, Fig. 2(a) shows that web pages in tablet devices are slightly bigger than those in smartphones, with an average page size of 2.4MB and 2.1MB, respectively. Figs. 2(b) and 2(c) further evidence the richness and complexity of the web pages in terms of number of resources (images, java-script, html, etc.) and external (root) domains, with more than 30% of the web pages consisting of more than 100 resources, and about 40% of the web pages contacting more than 10 different domains. Fig. 2(d) shows that the number of different DNS requests observed for a single page loading session can be as high as 160, and for more than 50% of the web pages, it takes more than 20 DNS requests to fetch the content. Again, tablet web pages consist of a slightly higher number of resources, requiring a higher number of requests.

Fig. 3 depicts the shares of web page bytes per content type, and per device. Content is split in *html*, *java-script*, *CSS*, *image*, *font*, *video*, and *other* content types. Images and java-script contents make the most of the bytes, with 50% of the pages having a share of either images or java-script above 40%. Tablet pages have a higher share of image contents, whereas smartphone pages have more java-script. Video contents are very limited, with a small share below 30% and only present in about 10% of the pages.

Finally, in terms of performance, Fig. 4 depicts the distribution of three relevant Web QoE metrics for the different web page loading sessions, discriminated by device type. These include: the Time to First Paint (TTFP), which accounts for the time at which the first object is painted on the browser, the RUMSI, and the PLT. The Time to First Byte (TTFB) is also added, as a timing reference. TTFB values are almost identical, as they do not reflect the performance of the content rendering, but rather the performance on the network/server side. TTFP values are also very similar, but RUMSI values are higher for tablet, and PLT are significantly higher, which is explained by the fact that web pages have more contents/resources to load in tablet (cf. Fig.2). Also interesting to note is how PLT significantly overestimates the perceived loading time of the contents, represented by the (RUM)SI metric.

B. App Data Characterization

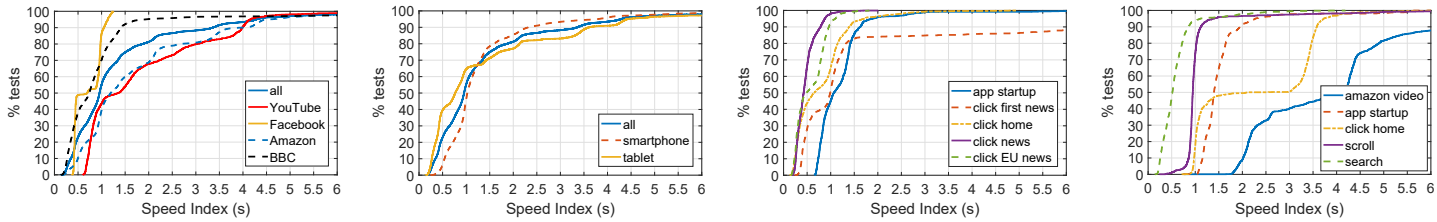
To show the heterogeneity of samples generated by the app tests, Fig. 5 presents the distribution of downloaded bytes per app and device type, for all the tested user actions. The first interesting observation is that the volume of traffic generated by the Facebook app at startup is negligible, which corresponds to the fact that no user account is associated to this app, and that the app cache is cleared at the beginning of each new startup test. In addition, while the YouTube and BBC News apps generate significantly higher traffic volumes in smartphone, the Amazon app produces slightly more traffic in the tablet device, pointing to significant differences in app behavior depending on the mobile device type.

In terms of loading performance, Fig. 6 reports the SI (a) per app, (b) per device type, and (c-d) per user action for BBC News and Amazon apps, respectively. Firstly, considering the RUMSI values obtained for the web browsing test (cf. Fig. 4), obtained app SI values are significantly lower, which adds to the heterogeneity of the overall study. YouTube and Amazon SI values are markedly higher, which is coherent with the higher downloaded volumes of these apps. According to Fig. 6(b), variations between devices are noticeable, especially for lower SI values. Finally, regarding SI for specific user interactions, Figs. 4(c-d) show bi-modal distributions for most of the interactions – corresponding to the different device types, as well as much higher SI variations for the Amazon app, suggesting that identification of particular user interactions might be relevant for the analysis.

C. Targets and Input Features

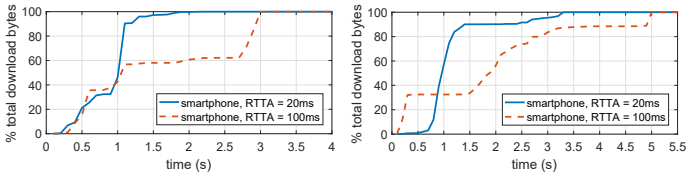
We treat the inference of the (RUM)SI metric as a regression problem. To define input features, we follow the rationale behind the computation of the SI metric itself, which considers the whole progress of the page loading. We define the Cumulative Bytes Downloaded features $CBD(i)_{\Delta T}$, as the (normalized) cumulative number of bytes downloaded from the first collected byte at time t_0 (TTFB) up to time $t = t_0 + i \times \Delta T$, with $i = 1, \ldots, m$. The CBD features track the download progress of the page bytes, using a time resolution ΔT. Fig. 7 depicts examples of CBD features for different network configurations, using $m = 100$ and $\Delta T = 100$ms. Pages loading faster have a CBD *loading curve* rising sharper and arriving to full download earlier.

We take $m = 100$ samples, and consider three different resolutions to compute features, using ΔT = 50ms, 100ms, and 500ms, for a total of 300 CBD features. Using different resolutions helps in capturing different phenomena in the traffic downloading progress, and allows to track different page load durations, in this case up to 5, 10, and 50 seconds, respectively. We consider $n = 11$ additional input features, related to the complete page loading session; these include: full session duration (first to last packet), downlink/uplink session duration (first to last packet in downlink/uplink direction), total



(a) SI per app. (b) SI per device. (c) SI per user action (BBC). (d) SI per user action (Amazon).

Fig. 6: Time performance of app loading sessions, considering specific app, device, and user action.



(a) https://zoom.us (b) https://cnn.com

Fig. 7: Examples of CBD features, using ΔT = 100ms, and different Access RTT (RTTA) network setups.

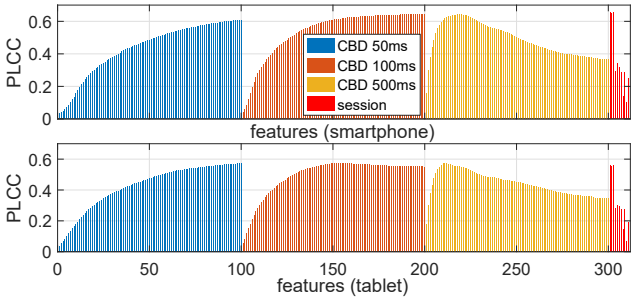


Fig. 8: Input features to RUMSI correlation.

packets downlink/uplink/full, total bytes downlink/uplink/full, and session mean throughput downlink/uplink. Fig. 8 depicts the linear correlation between these input features and the RUMSI metric, for the web tests. Correlation values are slightly higher for smartphone, with stronger correlations observed for CBD features between 5 seconds and 10 seconds, as well as for session-duration features.

IV. Mobile Web and App QoE Inference

Using the generated data and network traffic features, we train multiple regression models to infer the (RUM)SI metric. Given the identified differences between device types in terms of page contents and loading performance, we first consider the case of per device-type models, and then generalize to a multi-device scenario, training single models on all devices data. We then extend evaluations to include apps, and finally test a single integrated model, tackling both web pages and apps, for both device types. **Results presented next correspond to 5-fold cross validation**, and models are parametrized through grid-search.

A. Web QoE Inference per Device Type

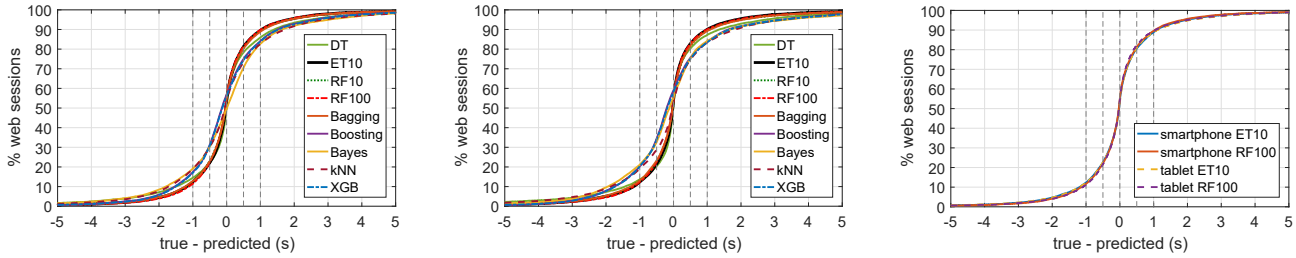
Tab. I reports the RUMSI inference performance attained by 9 different ML models, most of them based on decision trees, for smartphone and tablet devices. These models include single decision tree (DT), multiple types of ensembles using different numbers of trees, such as randomized trees (ET), random forest (RF), bagging trees, and boosting - including XGB optimizations. The list is completed by a plain Bayesian approach, and by the standard k nearest neighbors (kNN). We assess performance using three standard performance metrics for regression problems, including the absolute error (AE), the relative error (RE), and the linear correlation (PLCC). We take both mean (M) and median (m) values for the error metrics, to filter out significantly large errors. Figs. 9(a), 9(b) additionally depict the distribution of the inference errors.

model	dev	MAE-mAE (ms)	MRE-mRE (%)	PLCC
DT	S	1021 – 372	34 – 14	0.770
	T	1082 – 298	31 – 11	0.731
ET10	S	**788 – 354**	**28 – 13**	**0.859**
	T	**804 – 314**	**25 – 11**	**0.867**
RF10	S	813 – 383	29 – 14	0.856
	T	867 – 357	27 – 12	0.852
RF100	S	**764 – 360**	**27 – 13**	**0.876**
	T	**815 – 334**	**26 – 12**	**0.866**
Bagging	S	820 – 380	29 – 14	0.855
	T	874 – 362	27 – 13	0.853
Boosting	S	1067 – 598	42 – 23	0.834
	T	1206 – 642	43 – 24	0.813
Bayes	S	1245 – 668	48 – 26	0.749
	T	1337 – 626	47 – 25	0.697
kNN	S	1205 – 639	46 – 23	0.724
	T	1284 – 592	44 – 21	0.709
XGB	S	1068 – 601	42 – 23	0.831
	T	1207 – 652	43 – 24	0.811

TABLE I: RUMSI inference performance using ML models for (S)martphone and (T)ablet data.

RF100 achieves the best inference performance for both smartphone and tablet, with a median absolute error of 360 ms/334 ms, and a median relative error around 13%. Absolute inference errors are below 500ms for more than 60% of the sessions, and more than 80% of the session RUMSI values are inferred with an error below 1 second. Similar performance is realized by smaller ensembles - e.g., RF10, ET10, and bagging, using 10 instead of 100 trees. Given the training speed improvements attained by the ET10 model, we take it as the underlying model in subsequent evaluations.

As a reference to understand the implications of the achieved errors in terms of user experience, the limits of human perception imply that we find it difficult to correctly order



(a) Smartphone. (b) Tablet. (c) Multi-device.

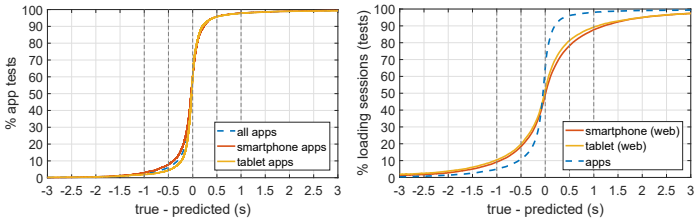
Fig. 9: RUMSI inference performance, using (a,b) per device models and (c) multi-device models.

model	MAE-mAE (ms)	MRE-mRE (%)	PLCC
ET10 $_{S \to T}$	1139 – 510	36 – 18	0.778
ET10 $_{T \to S}$	1028 – 526	38 – 20	0.815
ET10 $_{MD \to MD}$	802 – 345	27 – 12	0.863
RF100 $_{MD \to MD}$	758 – 320	25 – 11	0.873

TABLE II: Inference for cross-device (S)martphone-(T)ablet data, and Multi-Device (MD) data.

device (content)	MAE-mAE (ms)	MRE-mRE (%)	PLCC
APPS	246 – 90	22.6 – 9.5	0.932
S (web)	751 – 349	25.2 – 13.5	0.861
T (web)	763 – 338	25.3 – 11.7	0.865
A	310 – 115	29.7 – 12.0	0.917

TABLE III: Apps QoE inference and integrated model.



(a) Apps model. (b) Integrated model.

Fig. 10: (RUM)SI inference performance, using (a) apps model and (b) integrated, multi-device, multi-contents model. ET10 is used as underlying model.

visual events separated by less than 30ms. Studies summarized in the literature [29] additionally suggest that, depending on the specific task, the minimum perceptual duration of a perceivable stimulus lies between 25ms and 150ms. Therefore, we could already hypothesize that the median error of 350ms has a perceivable yet limited impact on the inference of the user web browsing experience. Finally, using standardized Web QoE (MOS) models for waiting times [6], we verified that for more than 92% of the web page loading sessions, the realized inference error does not result in a change to the inferred QoE class, considering 3 QoE classes in a standard 5-ACR MOS scale [6] – excellent QoE (MOS > 4), good QoE ($3 <$ MOS < 4), and poor QoE (MOS < 3).

B. Cross and Multi-device Web QoE Inference

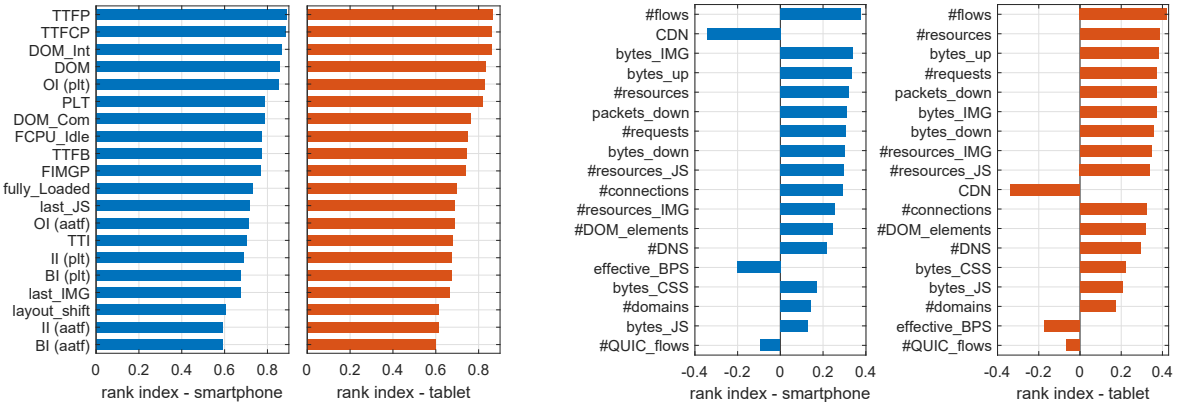
A question that poses regarding generalization of models across different mobile device types is how would cross-device models perform? We refer to cross-device models as those trained for one specific device type, e.g., smartphone, and applied to other device types, e.g., tablet. This is a critical aspect in the practice, which has been generally neglected in the literature [17], [3], [4], [21], [18], [19], where Web QoE models have been tailored for desktop web browsing.

Tab. II reports the performance achieved by cross-device models, using ET10 as underlying ML approach. We use the notation $S \to T$ and $T \to S$ for a model trained using exclusively smartphone/tablet measurements and tested on tablet/smartphone measurements, respectively. There is a strong performance degradation when applying cross-device models, with median absolute errors close to doubling as compared to per-device models. Absolute errors increase by 200ms to 300ms, and relative errors by about 10%. While we do not report it in this paper, the performance degradation when considering cross-device models between desktop and mobile devices is significantly higher, again pointing to the paramount need of considering multi-device measurements to build robust and reliable Web QoE inference models, applicable in the practice.

In this direction, Tab. II also reports the inference performance achieved by multi-device (MD) models, using both ET10 and RF100 as underlying ML approaches. A single MD model is trained on data from both smartphone and tablet devices. Results for MD models are almost identical to those attained by per-device models, with a slight degradation for smartphone and a slight improvement for tablet. This suggests that proper inference generalization can be achieved by considering sufficient device heterogeneity in the training step. Similar conclusions are drawn when additionally considering desktop measurements for MD models training.

C. Apps QoE Inference

The last step of the assessment considers the inference of the SI values for app user interactions. Fig. 10(a) depicts the distribution of inference errors obtained with an ET10 model, trained on top of the complete apps dataset. The first row of Tab. III summarizes the obtained results for this dataset. The obtained performance is significantly better for apps than the results so far obtained for web pages, achieving median absolute errors (mAE) of less than 100ms, and with more



(a) App-layer metrics (rank) correlation to RUMSI.

(b) Content/network metrics (rank) correlation to RUMSI.

Fig. 11: Application-layer metrics, web page content, and network-layer metrics correlated to RUMSI.

than 85% of the instances inferred with an AE below 500ms. Fig. 10(a) also splits the obtained results per device type, evidencing that a single model can properly address apps in both device types. The main difference between web and app datasets, and the corresponding inference performance, lies on the different span of contents on each dataset: while the web dataset targets 500 different web pages, the apps dataset covers only four apps and 12 different user interactions, making it easier for the model to provide more accurate predictions.

Finally, Fig. 10(b) reports the performance of an integrated, multi-device and multi-content model for Web and App QoE inference, built on top of a combined web and app measurements dataset including the over 40,000 measurements generated in the study. Error distributions are presented separately for the application of the integrated model on smartphone web measurements – S (web), tablet web measurements – T (web), and all-device/all-user-actions app measurements (A). Tab. III summarizes the obtained results for this single, ET10 model. Obtained results are consistent with both web and app results presented so far, providing almost the same results as the MD web model evaluated in Tab. II for web contents, and slightly worse results for apps as compared to the ET10 model for apps only (cf., first row of Tab. III).

Conclusion: multi-device and multi-app models can be properly trained to tackle the Web and App QoE inference problem from encrypted traffic, offering reliable monitoring capabilities for ISPs.

V. Mobile Web QoE Insights

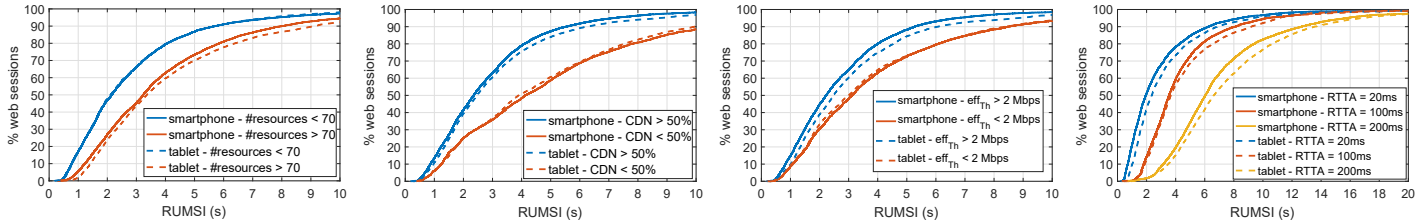
We devote the last part of the paper to dig deeper into the interplays between web page contents, network, and Web QoE. In particular, we shed light upon which characteristics from both the content of the web pages and the network have a stronger correlation to the RUMSI, and how some of them impact Web QoE. We note that the insights here correspond to an end-user point of view, from the single vantage point where the measurements were collected. Fig. 11 shows the rank correlation between (a) Web QoE-related metrics captured at the browser (application layer), and (b) web page content and network layer. As expected, there is a strong correlation between RUMSI and all loading/timing metrics, including TTFP, DOM, TTFB, PLT, the loading of last image and javascript contents, and other progressive or integral-like metrics such as Object/Image/Byte Index. Other timing metrics worth mentioning are those related to readiness for page interactivity, such as Time to Interactive (TTI) and time to First CPU Idle (FCPU_idle). Both reflect the time it takes for the page to be actually usable and *actionable* by the end-user.

From the page content and network points of view, the more flows, bytes, resources, connections, visited domains, and DNS requests needed to load the page, the higher the RUMSI, and thus the worse the Web QoE. The CDN metric reflects the fraction of static contents retrieved from CDNs, and the more static content is served from a CDN, the lower the RUMSI. Similar – but much lower, negative correlations exist for the effective download throughput - eff_{Th} (downloaded volume to fully loaded time ratio), and the number of QUIC flows.

Fig. 12 depicts the impact that some of the flagged metrics have on the RUMSI. Figs. 12(a) and 12(b) evidence how strong is the impact of the number of resources to load (split is done at the median number of resources, cf. Fig. 2(b)) and the fraction of static contents served from CDNs (split is done at 50%). For example, whereas the average RUMSI for pages with CDN>50% is 2.9 seconds, it is 5.5 seconds for pages with CDN<50%. With a median eff_{Th} of 2Mbps, Fig. 12(c) shows that loading sessions with a lower/higher eff_{Th} correspond to an average RUMSI of 4.3/2.8 seconds. Finally, Fig. 12(d) depicts the impact of the RTT set at the access (RTTA). For the sake of completeness, we include an additional set of measurements for RTTA = 200ms. The most interesting observation is how the impact of RTT on the RUMSI gets amplified due to the multiple exchanges to fully download the contents, with an average RUMSI of 3.2/4.5 seconds when increasing RTTA from 20ms to 100ms, and an average RUMSI of 7.2 seconds for RTTA = 200ms.

VI. Concluding Remarks

Mobile Web and App QoE monitoring and analysis are complex tasks. By generating a large dataset of Web and App QoE measurements for mobile devices, we have conceived



(a) SI vs number of resources. (b) SI vs CDN-hosting. (c) SI vs effective throughput. (d) SI vs access RTT (RTTA).

Fig. 12: RUMSI dependence on network and web page content characteristics.

an ML-based approach capable to infer the (RUM)SI of web browsing sessions and app user actions in smartphone and tablet devices with low errors and cross-device, multi-content generalization, using as input features derived from (encrypted) network traffic. We have shown the paramount impact of the device type used for web browsing on the modeling and inference performance, putting into question the applicability of previous approaches in the state of the art, mostly targeting Web QoE for desktop browsing. The extension of the Web QoE problem to consider mobile apps opens the door to a broader perspective for the mobile QoE monitoring and analysis problem, evidencing the complexity faced by ISPs to tackle this monitoring problem. As an additional outcome, we have studied the relationships and impact of multiple web page content and network characteristics on mobile Web QoE, shedding light on important aspects of web content and network characteristics impacting the experience of the end users. As an overall conclusion, and different from the state of the art on Web and App QoE monitoring and analysis, this is the first paper directly inferring Web QoE metrics such as (RUM)SI exclusively from in-network traffic measurements for mobile devices, providing cross mobile device-type and multi-content generalization.

Finally, while the problem of Web and App traffic identification and disentangling is out of the scope of this paper, we acknowledge that we have conceived multiple techniques addressing this problem, such that an end-to-end monitoring solution could be deployed by an ISP in the practice.

REFERENCES

[1] S. Wassermann, P. Casas, Z. Ben Houidi, A. Huet, M. Seufert, N. Wehner, J. Schüler, S. Cai, H. Shi, J. Xu, T. Hofeld, D. Rossi, "Are you on Mobile or Desktop? On the Impact of End-User Device on Web QoE Inference from Encrypted Traffic," in *CNSM*, 2020.

[2] N. Wehner, M. Seufert, J. Schüler, P. Casas, "How are your Apps Doing? QoE Inference and Analysis in Mobile Devices," in *CNSM*, 2021.

[3] A. Huet et al., "Revealing QoE of Web Users from Encrypted Network Traffic," in *IFIP Networking*, 2020.

[4] A. Huet et al., "Web Quality of Experience from Encrypted Packets," in *ACM SIGCOMM Posters and Demos*, 2019.

[5] Cisco, "Cisco Annual Internet Report (2018-2023) White Paper, Updated March 2020," Cisco, Tech. Rep., 2020.

[6] E. Ibarrola et al., "Web QoE Evaluation in Multi-agent Networks: Validation of ITU-T G. 1030," in *2009 Fifth International Conference on Autonomic and Autonomous Systems*, 2009.

[7] S. Egger, T. Hoßfeld, R. Schatz, and M. Fiedler, "Waiting Times in Quality of Experience for Web Based Services," in *QoMEX*, 2012.

[8] "G.1030 : Estimating End-to-end Performance in IP Networks for Data Applications," https://www.itu.int/rec/T-REC-G.1030.

[9] T. Hoßfeld, F. Metzger, and D. Rossi, "Speed Index: Relating the Industrial Standard for User Perceived Web Performance to Web QoE," in *QoMEX*, 2018.

[10] Q. Gao, P. Dey, and P. Ahammad, "Perceived Performance of Top Retail Webpages in the Wild: Insights from Large-scale Crowdsourcing of Above-the-fold QoE," in *Internet-QoE*, 2017.

[11] D. N. da Hora, A. S. Asrese, V. Christophides, R. Teixeira, and D. Rossi, "Narrowing the Gap between QoS Metrics and Web QoE using Above-the-fold Metrics," in *PAM*, 2018.

[12] A. Sackl, S. Egger, and R. Schatz, "The Influence of Network Quality Fluctuations on Web QoE," in *QoMEX*, 2014.

[13] A. Sackl, P. Casas, R. Schatz, L. Janowski, and R. Irmer, "Quantifying the Impact of Network Bandwidth Fluctuations and Outages on Web QoE," in *QoMEX*, 2015.

[14] M. Varela, L. Skorin-Kapov, T. Mäki, and T. Hoßfeld, "QoE in the Web: a Dance of Design and Performance," in *QoMEX*, 2015.

[15] M. Varela, T. Mäki, L. Skorin-Kapov, and T. Hoßfeld, "Towards an Understanding of Visual Appeal in Website Design," in *QoMEX*, 2013.

[16] S. Baraković and L. Skorin-Kapov, "Survey of Research on Quality of Experience modelling for Web Browsing," *Quality and User Experience*, vol. 2, no. 1, p. 6, 2017.

[17] A. Saverimoutou, B. Mathieu, and S. Vaton, "A 6-month Analysis of Factors Impacting Web Browsing Quality for QoE Prediction," *Computer Networks*, vol. 164, p. 106905, 2019.

[18] A. S. Asrese, S. J. Eravuchira, V. Bajpai, P. Sarolahti, and J. Ott, "Measuring Web Latency and Rendering Performance: Method, Tools & Longitudinal Dataset," *IEEE Transactions on Network and Service Management*, 2019.

[19] M. Rajiullah et al., "Web Experience in Mobile Networks: Lessons from Two Million Page Visits," in *WWW*, 2019.

[20] S. Ihm and V. S. Pai, "Towards Understanding Modern Web Traffic," in *Internet Measurement Conference*, 2011.

[21] E. Bocchi, L. De Cicco, and D. Rossi, "Measuring the Quality of Experience of Web Users," *ACM SIGCOMM Computer Communication Review*, vol. 46, no. 4, 2016.

[22] M. Trevisan, I. Drago, and M. Mellia, "PAIN: A Passive Web Performance Indicator for ISPs," *Computer Networks*, vol. 149, 2019.

[23] V. Aggarwal, E. Halepovic, J. Pang, S. Venkataraman, and H. Yan, "Prometheus: Toward Quality-of-Experience Estimation for Mobile Apps from Passive Network Measurements," in *HotMobile*, 2014.

[24] A. Balachandran et al., "Modeling Web Quality of Experience on Cellular Networks," in *MOBICOM*, 2015.

[25] 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, 2016.

[26] S. Wassermann, N. Wehner, and P. Casas, "Machine Learning Models for YouTube QoE and User Engagement Prediction in Smartphones," *SIGMETRICS Perform. Eval. Rev.*, vol. 46, no. 3, 2019.

[27] A. Nikravesh, Q. A. Chen, S. Haseley, X. Zhu, G. Challen, and Z. M. Mao, "QoE Inference and Improvement without End-host Control," in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, 2018.

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

[29] B. Dainton, "Temporal Consciousness", *The Stanford Encyclopedia of Philosophy* (Winter 2018 Edition), Edward N. Zalta (ed.).

[30] P. Meenan, "WebPageTest - Website Performance and Optimization Test," 2020. [Online]. Available: https://www.webpagetest.org/

[31] ——, "Real User Monitoring SpeedIndex (RUMSI) - SpeedIndex Measurements from the Field using Resource Timings," 2020. [Online]. Available: https://github.com/WPO-Foundation/RUM-SpeedIndex