

Predicting QoE in Cellular Networks using Machine Learning and in-Smartphone Measurements

Pedro Casas[†], Alessandro D’Alconzo[†], Florian Wamser^{*}, Michael Seufert^{*}
Bruno Gardlo[†], Anika Schwind^{*}, Phuoc Tran-Gia^{*}, Raimund Schatz[†]

[†]AIT – Austrian Institute of Technology, Vienna, Austria
{forename.surname}@ait.ac.at

^{*}University of Würzburg, Institute of Computer Science, Würzburg, Germany
{forename.surname}@informatik.uni-wuerzburg.de

Abstract—Monitoring the Quality of Experience (QoE) undergone by cellular network customers has become paramount for cellular ISPs, who need to ensure high quality levels to limit customer churn due to quality dissatisfaction. This paper tackles the problem of QoE monitoring, assessment and prediction in cellular networks, relying on end-user device (i.e., smartphone) QoS passive traffic measurements and QoE crowdsourced feedback. We conceive different QoE assessment models based on supervised machine learning techniques, which are capable to predict the QoE experienced by the end user of popular smartphone apps (e.g., YouTube and Facebook), using as input the passive in-device measurements. Using a rich QoE dataset derived from field trials in operational cellular networks, we benchmark the performance of multiple machine learning based predictors, and construct a decision-tree based model which is capable to predict the per-user overall experience and service acceptability with a success rate of 91% and 98% respectively. To the best of our knowledge, this is the first paper using end-user, in-device passive measurements and machine learning models to predict the QoE of smartphone users in operational cellular networks.

I. INTRODUCTION

By 2019, more than 75% of all mobile network traffic will be consumed and generated by smartphones [1]. As such, there is an ever-growing interest from cellular network operators to better understand and assess the performance of their networks as perceived by the end users directly at their own smartphones. In this paper we present a set of monitoring tools and machine learning models to enable distributed Quality of Experience (QoE) monitoring and prediction, relying exclusively on in-device passive measurements. In-device network and traffic monitoring provides highly valuable information to network operators, as measurements are taken as close as possible to the end user, shedding light into particular phenomena not observable from traditional, in-network monitoring (e.g., contextual information). Using end-devices as vantage points becomes specially attractive in current traffic-encryption context, as end-to-end encryption is rapidly growing – specially

due to the massive adoption of HTTPS, obfuscating network traffic analysis.

The most common approach to evaluate the performance of networks and services from a QoE perspective is to conduct experiments in the lab [2]–[4]. Their key benefits rely on the full controllability on the evaluation process. Nevertheless, experiments in the lab normally introduce differences as compared to field studies [5], as they might overlook important QoE-relevant features such as contextual information, device usability, or user preferences. Field trials would therefore result in better, more representative evaluations, to the cost of higher complexity in acquiring and processing the results.

We study the QoE of popular apps in smartphones (YouTube, Facebook, and Google Maps), combining passive traffic QoS measurements collected directly in smartphones with crowdsourced QoE feedbacks provided by the end user in a field trial. While we are fully aware that accurate QoE estimation requires measurements collected at multiple levels of the communications stack – including network, application and end-user layers, we take a practical approach and provide models to map network traffic QoS metrics to QoE directly. We do so to maximize the usage of generic network traffic measurements which can be collected in smartphones, independently of the specific app. Application level monitoring is generally more cumbersome, as not every app provides APIs to access relevant metrics, and in most cases, device root access must be granted to perform measurements deeply into the app, hindering large-scale passive monitoring.

The field trial runs in real operational cellular networks, using the personal smartphones and data contracts of the field trial participants. To collect in-device traffic measurements and QoE feedbacks, we developed several tools including a passive monitoring tool to measure the traffic of the field trial participants at their end devices and a QoE-feedback app to gather user experience data in a crowdsourced fashion. Using the collected measurements, we train several supervised Machine Learning (ML) models to predict the QoE undergone by the end user of these apps. ML provides a promising alternative for QoE prediction and assessment based on the combined analysis of multiple traffic descriptors or features. We benchmark the prediction performance of different models for the aforementioned apps, considering overall experience and acceptability as target QoE metrics.

Partly funded by EU project MONROE (H2020-2014-ICT-644399, open call project Mobi-QoE) and by WWTF project Big-DAMA (WWTF-ICT15-129).

The remainder of the paper is organized as follows: Sec. II overviews the related work, focusing on the specific case of mobile devices. Sec. III describes the field trial setup, the tools developed to measure QoS- and QoE-related metrics at the end-devices, and the machine-learning models used as predictors. Sec. IV provides a brief overview on the field trial measurements, and focuses on the evaluation of the prediction performance obtained by different ML-based models, including an analysis of the relevance of the different input features on quality prediction. Finally, Sec. VI concludes this work.

II. RELATED WORK

The study of the QoE for modern web-based applications as the ones we target in this paper has a long list of fresh and recent references [9]. Due to its popularity, the specific case of YouTube QoE deserves particular attention. Previous papers [10]–[13] have shown that stallings, initial playback delays and quality switches are the most important KPIs for adaptive streaming YouTube QoE. A comprehensive survey on adaptive video streaming QoE is provided in [14].

Regarding QoE monitoring studies in cellular networks and mobile devices, there is an assorted list of tools to measure network performance: some examples are Mobiperf [15], Mobilyzer [16], and Netalyzr [17]. In [18], [19] authors introduced YoMoApp, an app to passively monitor YouTube QoE-related features in smartphones. In [20] authors describe an on-line monitoring system for YouTube QoE in cellular networks using in-network measurements only. QoE Doctor [21] measures mobile app QoE, using active measurements at both application and network layers. Other similar papers study Web QoE in cellular networks [22], and video [23]. Close to our work, authors in [24] propose an approach to evaluate mobile apps QoE using passive in-network measurements and machine-learning models mapping QoS to QoE. The main difference to our approach is the usage of in-network measurements, contrary to in-device ones.

While the majority of these papers only focus on QoE-relevant objective metrics such as rebufferings, page load times and interaction latencies, they lack a direct reference to user feedback, e.g., Mean Opinion Scores. Here we take feedback from real users and collect network measurements to provide a holistic perspective to the problem of QoE monitoring in smartphones. This paper builds on top of previous work on QoE for cellular networks [7], [8]. In particular, we consider a machine-learning based perspective to the problem of QoE prediction and assessment, which is not done in [7], [8].

III. MOBILE QOE IN THE FIELD

The field trial took place in Vienna in 2015; 30 users equipped with their own devices connected to their preferred cellular operators evaluated the three apps as part of their normal daily Internet activity. Participants were instructed to perform independent tasks for each app. For YouTube, they were requested to watch short (about 2 minutes long) HD YouTube videos. For Facebook, they were requested to access the app, and browse both the timeline and some photo albums of a fake user. For Gmaps, they had to browse maps of different cities using the satellite-view of the Google maps app.

Table I. METRICS COLLECTED FOR EACH DATA FLOW.

Metric ID	Metric Name	Units	Example
1	device id (IMEI)	–	352668049725157
2	flow start time	s	1430825689
3	flow direction (up/down)	–	downlink
4	flow duration	s	10,24
5	flow size	KB	4041,00
6	avg. flow throughput	kbps	3157,03
7	max. flow throughput	kbps	4320,15
8	app (Android API package)	–	com.android.browser
9	signal strength	dBm	-71
10	operator (MCC.MNC)	–	295.4
11	cell id	–	16815
12	cell location (lat-lon)	deg (°)	{40,198-12,347}
13	RAT	–	LTE

QoE feedback was reported for each session through a customized QoE crowdsourcing app, for a total span of two weeks. In this paper we only focus on the overall experience declared by participants and the service acceptability, but the QoE feedback provided includes more information that we plan to evaluate in the future. The overall experience is rated according to a discrete, 5-levels ACR Mean Opinion Score (MOS) scale [2], ranging from “bad” (i.e., MOS = 1) to “excellent” (i.e., MOS = 5). Acceptability is a binary feedback, stating whether the user would continue using the application under the corresponding conditions or not. Apps network traffic was passively monitored and analyzed at their devices with the tools described next. Participants also indicated their location at the time of the test (e.g., underground, car, home, street, etc.).

A. QoS/QoE Measurement Tools

We implemented two Android apps to monitor participants’ network traffic and to collect their QoE feedbacks: (i) a passive flow-level traffic monitor and (ii) a web-based QoE feedback survey. The flow-level monitor passively sniffs network traffic, associating flows to apps generating them. Tab. I depicts the metrics collected for each traffic flow, i.e., standard 5-tuple flows, using the Android developers’ APIs. The IMEI is a unique device identifier. Metrics with ID 2 to 7 correspond to flow measurements, such as start time, direction, duration, size, and average/maximum flow transfer throughput. Metric ID 8 indicates the app generating the flow, using the Android API notation for app naming. Metrics with ID 9 to 13 are registered at the time of the flow start, and include signal strength, operator and cell to which the smartphone is attached to (including its geographical location), and the RAT - e.g., LTE, 3G, 2G, EDGE, etc. Metrics are locally stored and periodically sent to a server for post-processing and analysis.

The web-based QoE survey app is manually run by the participant immediately after completing a specific test to collect his QoE feedback. In this paper, a QoE feedback entry includes: {timestamp; app; location; MOS; acceptability}. Both tools are time-synchronized, thus a valid MOS score would always have a newer timestamp than that logging the associated flows’ start.

To correlate the traffic measurements and the QoE feedback provided by the field trial participants, we group flows into

Table II. SESSION-BASED KPIS. (U) INDICATES USER-REPORTED.

KPI Name	KPI Description (U – reported by user)
MOS	overall user experience (U)
ACC	service acceptability (U)
ISP	cellular network operator
RAT	radio access technology
SIG	avg. signal strength
TH_{max}	max. session downlink flow throughput
TH_{avg}	avg. session downlink flow throughput
DUR	session duration
VOL	session volume
$FLOW_{ratio}$	ratio (# flows up)/(# flows down)
CELL	cell id
LOC	user location context (U)

sessions. A session corresponds to a group of flows generated by the same app which are continuous in time. More precisely, we define a session as all the flows associated to a certain app app_{MOS} for which a MOS rating has been provided by the participant at time t_{MOS} and which have started within a time window $[t_{MOS} - Th_{session}; t_{MOS}]$. The threshold $Th_{session}$ defines the maximum session duration, and it is set to 4 minutes, which is the average time requested to participants to take to perform a specific task.

The final step is to define session-based KPIS from the flow-based measurements described in Tab. I which are then used to correlate QoS measurements and QoE in terms of MOS scores and acceptability rates. For numerical features such as flow throughput, size, duration and signal strength (metric IDs 4,5,6,7 and 9), we simply consider the average value from all the flows belonging to the session. For non-numerical features such as ISP and RAT (metric IDs 10 and 13), we take a majority voting among all the flows of the session (i.e., we take the mostly observed value). Flow directions (metric ID 3) are combined into a session metric defining the ratio up/down. Combining these nine session-based KPIS with the location, app and QoE feedbacks provided by the participants, we end up with a fully labeled QoS/QoE dataset, which we use to build and train different machine-learning based predictors. Tab. II describes the complete set of features and targets.

B. Machine Learning Models for QoE Prediction

We propose a Machine Learning (ML) based QoE predictor using well-known decision trees. A decision tree is a classification algorithm that classifies instances by repeatedly partitioning the input space, so as to build a tree whose nodes are as pure as possible (i.e., they contain instances of a single class). Decision trees are a very appealing option; they are simple yet very fast and effective. Classification speed is a paramount asset when thinking in large-scale monitoring scenarios, and decision trees are well-known for their speed. They are also very easy to interpret, and directly provide filtering rules. In addition, decision trees explicitly show the importance of different features, as the learning algorithm automatically performs feature selection by choosing the most discriminating features. This is a paramount advantage as compared to other ML approaches, as decision trees are more robust to noisy or loosely correlated-to-class input features.

The C4.5 decision tree is the most frequently used algorithm, so we have conceived the proposed system based on such trees.

Next, we additionally evaluate other ML-based classifiers typically used in the literature. In particular, we consider the following classifiers: Artificial Neural Networks (MLP), Support Vector Machines (SVM), Naive Bayes (NB), and Random Forest (RF). We use the well-known Weka Machine-Learning software tool¹ to calibrate these ML-based algorithms and to perform the evaluations. Parameters are set manually for all the models, performing an extensive trial-and-error testing phase to obtain the best results. We address the interested reader to [6] and to the Weka documentation for additional information on the different configuration parameters of each algorithm.

IV. EVALUATION RESULTS

In this section we assess the proposed QoE predictor based on decision trees, firstly by evaluating its accuracy to predict the correct MOS value of each session without incurring into wrong predictions, and then by comparing its performance to correctly predict both MOS and acceptability values against the other ML-based approaches. Next we provide an overview on the measurements and QoE feedback obtained in the field trial, and then jump directly into the evaluation of the predictors.

A. Field Trial Measurements

Figs. 1(a-b) show the QoE feedbacks' distribution considering (a) number of feedbacks per app and (b) number of feedbacks per location. Figs. 1(c-d) report the distribution of MOS scores regarding (c) different apps and (d) different locations. About 700 feedbacks were reported. The highest number of feedbacks were received for YouTube, which is one of the most popular Internet apps. Home was the most common location, which is in line with previous field-trial results [5]. Interestingly, the underground was ranked second as preferred location for testing. Regarding MOS scores distributions, these are rather similar when considering different apps (cf. Fig. 1(c)) and locations (cf. Fig. 1(d)). This suggests that network performance was rather stable during the span of the study, and uniform for both low and highly mobile locations. Indeed, tests were performed in the city of Vienna, where all cellular ISPs have excellent network coverage, justifying these observations.

B. C4.5 Prediction Performance

Let us first get an initial picture of the MOS prediction capabilities of the C4.5-based approach, separately evaluating each app. We treat the prediction as a classification problem, where inputs correspond to the ten session-based KPIS described in Tab. II, and the output corresponds to one of the five MOS levels, i.e., $MOS = 1..5$. As such, each MOS level corresponds to a different output quality class in our classification problem. For each app, we evaluate the true and false positive rates (TPR and FPR respectively) achieved in the prediction of each MOS class. The TPR indicates how good is the approach to correctly predict all the individual MOS ratings within a specific class, whereas the FPR indicates how many of these predictions corresponded actually to another class.

¹Weka Data Mining, at <http://www.cs.waikato.ac.nz/ml/weka/>.

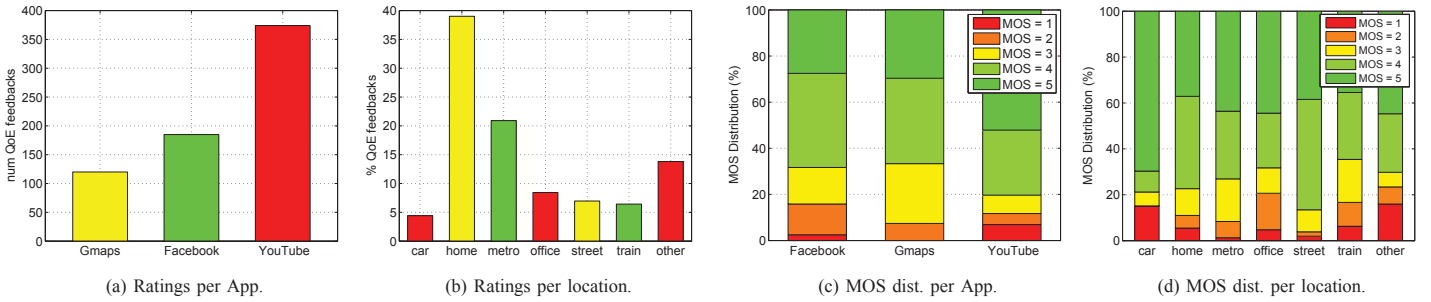


Figure 1. Distribution of QoE feedbacks in the field. The biggest share of ratings were done for YouTube. The preferred location was home, and MOS distributions are rather similar w.r.t. tested apps and selected locations, suggesting that network performance was rather stable during the span of the study.

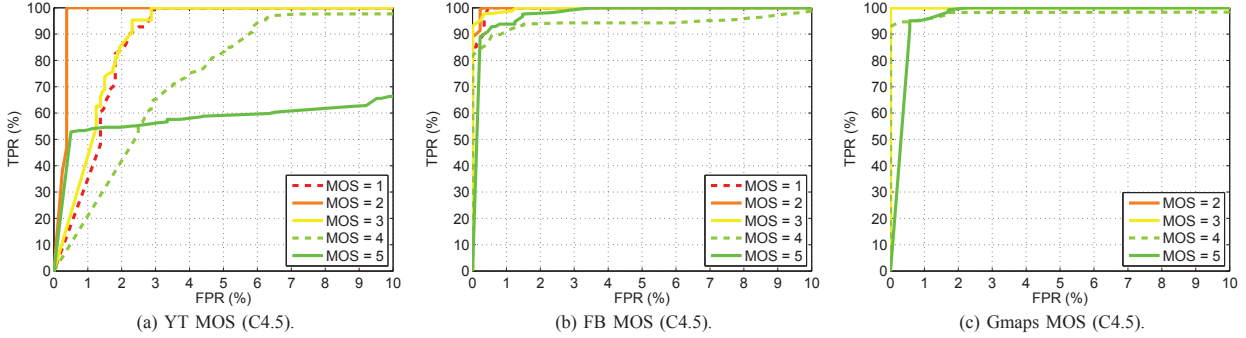


Figure 2. True Positives Rate (TPR) vs. False Positives Rate (FPR) obtained in the prediction of individual MOS ratings, using the C4.5-based predictor.

To reduce biased results, the training and testing of the C4.5 model is performed by 10-fold cross validation. As shown in Fig. 1(c), MOS classes are highly imbalanced – MOS = 4 and MOS = 5 are the biggest classes, which in principle imposes major challenges in the training phase. To counterbalance this problem, we resort to a standard bootstrapping approach, in which we add instances from the under-represented classes, in this case those corresponding to MOS = 1, MOS = 2 and MOS = 3, to perform the training. Such instances are randomly generated, using the empirical distributions of the features as statistical source.

Fig. 2 depicts the Receiver Operating Characteristic (ROC) curves obtained in the prediction of the per-app MOS scores, including (a) YouTube, (b) Facebook and (c) Gmaps. For each app, each ROC curve corresponds to the performance within each specific quality class (MOS = 1, 2, 3, 4, and 5). The first interesting observation is that MOS predictions are worse in the case of YouTube, for all the different quality levels. The reasons for this are quite straightforward, and come from the usage of standard streaming buffering, which partially compensates network QoS variations. Still, YouTube results are satisfactory for quality levels from 1 to 4, achieving almost full positive prediction for a FPR below 7%. Prediction results are almost perfect for the other two apps, suggesting that the mappings QoS-QoE are simpler in the case of Facebook and Gmaps. The last interesting observation is that in the three cases, the highest quality levels (i.e., MOS = 4 and MOS = 5) are the most difficult to predict. A deeper study on the confusion matrix reveals that values of the quality class MOS = 5 tend to be misclassified as quality class MOS = 4 and vice-versa, suggesting that the quality differences are harder to distinguish at the QoS level. As a conclusion, we can claim

that a QoE predictor based on C4.5 decision trees provides very high accuracy for the tested apps.

C. Benchmarking Multiple ML Predictors

We move on now to the evaluation of the QoE prediction and classification capabilities of the C4.5 model, as compared to the other tested machine-learning based approaches. Besides predicting the MOS scores, we additionally train the models to predict the binary acceptability (YES/NO) as declared by the user. Instead of performing an evaluation per app and to simplify the analysis, we consider now a combined approach, building a single model for the three apps together. To evaluate and compare the performance and virtues of the models, we consider three standard metrics: Global Accuracy GA - percentage of correctly classified ratings, Recall - per class accuracy and Precision (fidelity) - $TP_i / (TP_i + FP_i)$.

Fig. 3 reports the performance of the five compared classifiers in the classification of all the five quality levels. As before, evaluations are done following a 10-fold cross-validation approach. Reported results refer to optimal parameter settings, after thorough testing. According to Fig. 3(a), the C4.5 and the RF models achieve high overall classification accuracy, above 90% in both cases, and with a slightly better performance for the RF model. The RF model is built out of 10 parallel C4.5 decision trees, which explains the out-performance of the approach. The SVM and the NB models achieve worse results, clearly suggesting that the underlying hypotheses of both models do not hold in this case. The MLP model achieves an acceptable accuracy, close to 80%. In terms of precision and recall, depicted in Figs. 3(b) and 3(c) respectively, the C4.5-based models (i.e., C4.5 and RF) outperform all the other classifiers in all the quality classes,

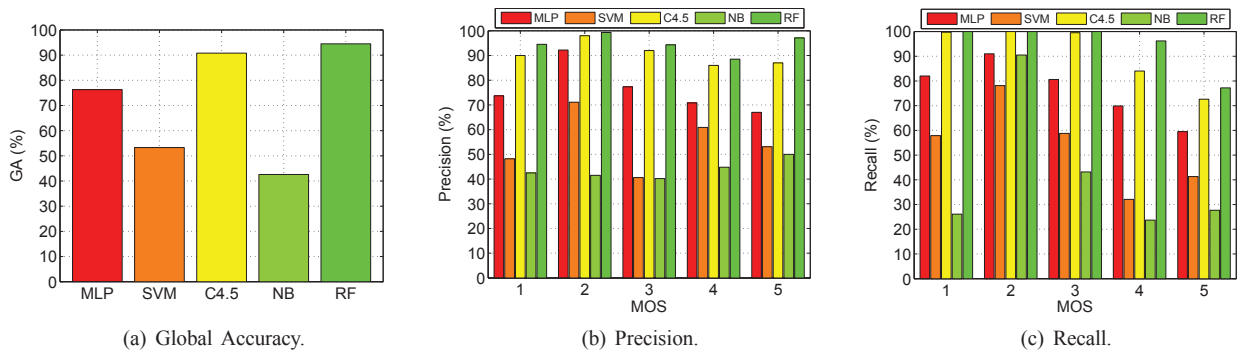


Figure 3. Classification Accuracy, Precision, and Recall for the 5 different MOS quality levels. The performance of the C4.5 and the RF models is almost perfect for all quality levels, with a slightly worse performance achieved for MOS = 5 predictions.

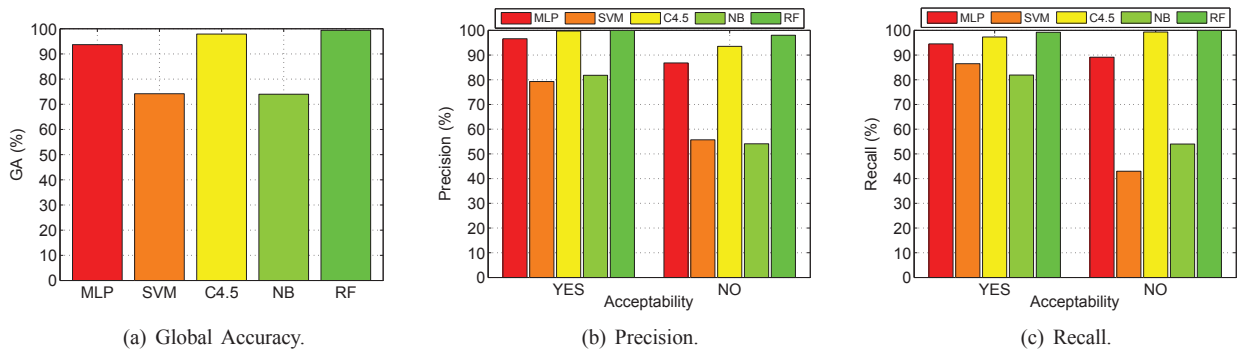


Figure 4. Classification Accuracy, Precision, and Recall for the 2 different acceptability levels. In this case, the performance of the C4.5 and the RF models is almost perfect for both positive and negative acceptability.

evidencing the nice properties introduced by decision trees. Still, as already evidenced in previous results, these models can not fully correctly deal with sessions belonging to the highest quality class, achieving a recall around 75% for the quality class MOS = 5. Similar to our previous evaluation, the main source of under-performance for the C4.5-based models comes from mixing the highest quality classes.

Finally, Fig. 4 reports the same results as before, but for the prediction of the acceptability metric. The same conclusions taken before still hold, but in this case, the MLP model performs very closely to the C4.5 one, suggesting that the underlying mapping QoS-QoE is easier to capture for the neural network. As a general conclusion of the benchmarking, results clearly suggest that using decision trees for QoE prediction provides highly accurate results, and that such models can correctly predict not only the overall experience of the user but also his acceptability perspective.

V. FEATURE ANALYSIS AND IMPACT ON QoE

To conclude, we perform a brief study on the impact of the different input features used by the prediction model on QoE. We consider the standard problem of feature selection, using correlation-based evaluation. This approach basically selects sub-sets of features that are poorly correlated among each other, but highly correlated to the target output. Fig. 5(a) depicts the most relevant selected features for each app and for both prediction targets. Features flagged in green/red are positively/negatively correlated to the target, whereas nominal features are marked in yellow.

Not surprisingly, the most relevant features are the average and the maximum session throughput (TH indicates both). Bulky sessions (i.e., bigger VOL) are generally better perceived and translate into higher acceptability, which might be linked to the specifics of the contents being consumed - e.g., HD video or better resolution maps, and the better usage of the available bandwidth. As expected, signal strength is also positively correlated to QoE, as the higher the signal strength, the better the connection performance - e.g., higher throughput and lower latency. Interestingly, longer YouTube sessions experience a worse QoE; a deeper analysis of longer YouTube sessions shows that many of them are rather small, suggesting the occurrence of stalling. ISP also appears as a relevant feature, and specially for acceptability, suggesting that service quality is different for the different operators on the field trial, as well as the corresponding user expectations. Recall that participants had their own data contracts, so a better look into the characteristics of these contracts in terms of agreed performance and cost would shed light on this.

Location is also relevant, and in particular for the case of Gmaps, where correlations between QoE and LOC are much higher. As we see in Fig. 5(b), this is most probably linked to the underlying mobility of the users under different locations. Fig. 5(b) shows the overall experience of the three apps, discriminated by *mobility pattern*; mobility patterns are simply extracted from participants declared locations. Locations “home” and “office” are aggregated into a *static* pattern, *slow-motion* covers “street” as location, and high mobility locations such as “car”, “train” and “metro” are grouped as *high-motion*. While there is no major impact of mobility pattern

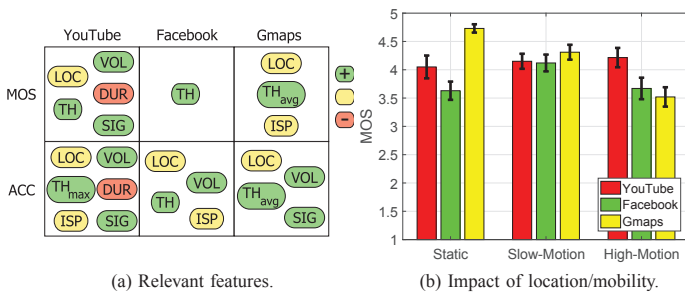


Figure 5. Relevance of different monitoring features on app QoE.

on YouTube and Facebook, there is a clear trend in Gmaps, and the more dynamic the user, the lower the QoE. We believe this is linked to the higher interactivity requirements of Gmaps w.r.t. the other apps, which might be impacted by handovers and/or network quality fluctuations while on the move.

VI. CONCLUDING REMARKS

In this paper we have addressed the problem of QoE monitoring, assessment and prediction in cellular networks, relying on in-smartphone QoS passive traffic measurements and QoE crowdsourced feedback. Using a rich QoS/QoE dataset derived from a field trial study conducted on multiple operational cellular networks, we have trained different QoE prediction models based on supervised machine learning techniques. The finally selected model is based on decision trees, which are very attractive for large-scale monitoring scenarios, not only because of their excellent performance but also due to their prediction speed.

All in all we have conceived a two-phase system which is capable of (i) generating a rich dataset of QoS/QoE measurements, which can be used to train the operational model, and (ii) predicting QoE in smartphones for popular apps in a distributed fashion, using only in-smartphone passive traffic measurements. Evaluations show that the proposed prediction features and model can correctly forecast the individual, per-user overall experience and service acceptability of popular apps in 91% and 98% of the monitored sessions. As a final contribution, we have performed a preliminary analysis on the impact of the selected input features on QoE, which could potentially enhance future applications of our proposal for diagnosis issues.

REFERENCES

- [1] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015-2020 White Paper," 2015. [Online].
- [2] Int. Telecommunication Union, "ITU-T Rec. P.800: Methods for Subjective Determination of Transmission Quality," 1996.
- [3] —, "ITU-T Rec. P.910: Subjective Video Quality Assessment Methods for Multimedia Applications," 2008.

- [4] —, "ITU-T Rec. P.1501: Subjective Testing Methodology for Web Browsing," 2013.
- [5] R. Schatz and S. Egger, "Vienna Surfing: Assessing Mobile Broadband Quality in the Field," in ACM W-MUST, 2011.
- [6] R. Duda, P. Hart, and D. Stork, *Pattern Classification (2Nd Edition)*, 2000.
- [7] P. Casas, B. Gardlo, M. Seufert, F. Wamser, and R. Schatz, "Taming QoE in cellular networks: From subjective lab studies to measurements in the field," in CNSM, 2015.
- [8] P. Casas, M. Seufert, F. Wamser, B. Gardlo, A. Sackl, and R. Schatz, "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, 2016.
- [9] P. Casas and R. Schatz, "Quality of experience in cloud services: Survey and measurements," *Computer Networks*, vol. 68, 2014.
- [10] T. Hobfeld, M. Seufert, M. Hirth, T. Zinner, P. Tran-Gia, and R. Schatz, "Quantification of youtube qoe via crowdsourcing," in IEEE ISM, 2011.
- [11] R. K. Mok, E. W. Chan, X. Luo, and R. K. Chang, "Inferring the qoe of http video streaming from user-viewing activities," in ACM W-MUST, 2011.
- [12] B. Lewcio, B. Belmudez, A. Mehmood, M. Wältermann, and S. Möller, "Video Quality in Next Generation Mobile Networks: Perception of Time-varying Transmission," in IEEE CQR, 2011.
- [13] P. Casas, R. Schatz, F. Wamser, M. Seufert, and R. Irmer, "Exploring QoE in Cellular Networks: How Much Bandwidth do you Need for Popular Smartphone Apps?" in ACM ATC, 2015.
- [14] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hobfeld, and P. Tran-Gia, "A Survey on Quality of Experience of HTTP Adaptive Streaming," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, 2015.
- [15] Mobiperf, "Mobiperf, Measuring Network Performance on Mobile Platforms," 2013. [Online]. Available: <http://mobiperf.com>
- [16] A. Nikraves, H. Yao, S. Xu, D. Choffnes, and Z. M. Mao, "Mobilyzer: An open platform for controllable mobile network measurements," in MobiSys, 2015.
- [17] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson, "Netalyzer: Illuminating the edge network," in ACM IMC, 2010.
- [18] F. Wamser, M. Seufert, P. Casas, R. Irmer, P. Tran-Gia, and R. Schatz, "Understanding YouTube QoE in Cellular Networks with YoMoApp – a QoE Monitoring Tool for YouTube Mobile," in ACM MOBICOM, 2015.
- [19] —, "YoMoApp: a Tool for Analyzing QoE of YouTube HTTP Adaptive Streaming in Mobile Networks," in EuCNC, 2015.
- [20] P. Casas, M. Seufert, and R. Schatz, "YOUQMON: A System for On-line Monitoring of YouTube QoE in Operational 3G Networks," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, 2013.
- [21] Q. A. Chen, H. Luo, S. Rosen, Z. M. Mao, K. Iyer, J. Hui, K. Sontineni, and K. Lau, "QoE Doctor: Diagnosing Mobile App QoE with Automated UI Control and Cross-layer Analysis," in ACM IMC, 2014.
- [22] A. Balachandran, V. Aggarwal, E. Halepovic, J. Pang, S. Seshan, S. Venkataraman, and H. Yan, "Modeling Web Quality-of-experience on Cellular Networks," in ACM MOBICOM, 2014.
- [23] M. Z. Shafiq, J. Erman, L. Ji, A. X. Liu, J. Pang, and J. Wang, "Understanding the Impact of Network Dynamics on Mobile Video User Engagement," in ACM SIGMETRICS, 2014.
- [24] 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 ACM HotMobile, 2014.