Boz, E.; Finley, B.; Oulasvirta, A.; Kilkki, K.; Manner, J.

Mobile QoE prediction in the field

Published in:
Pervasive and Mobile Computing

DOI:
10.1016/j.pmcj.2019.101039

Published: 01/10/2019

Document Version
Publisher's PDF, also known as Version of record

Pervasive and Mobile Computing 59 (2019) 101039

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Mobile QoE prediction in the field

E. Boz, B. Finley *, A. Oulasvirta, K. Kilkki, J. Manner

Department of Communications and Networking, Aalto University, Konemiehentie 2, Espoo, Finland

ARTICLE INFO

Article history:
Received 10 January 2019
Received in revised form 19 June 2019
Accepted 21 June 2019
Available online 25 June 2019

MSC:
00-01
99-00

Keywords:
Quality of experience
Hybrid measurements
Network monitoring

ABSTRACT

Quality of experience (QoE) models quantify the relationship between user experience and network quality of service. With the exception of a few studies, most research on QoE has been conducted in laboratory conditions. Therefore, in order to validate and develop QoE models for the wild, researchers should carry out large scale field studies. This paper contributes data and observations from such a large-scale field study on mobile devices carried out in Finland with 292 users and 64,036 experience ratings. 74% of the ratings are associated with Wifi or LTE networks. We report descriptive statistics and classification results predicting normal vs. bad QoE in in-the-wild measurements. Our results illustrate a 20% improvement over baselines for standard classification metrics (G-Mean). Furthermore, both network features (such as delay) and non-network features (such as device memory) show importance in the models. The models' performance suggests that mobile QoE prediction remains a difficult problem in field conditions. Our results help inform future modeling efforts and provide a baseline for such real-world mobile QoE prediction.

1. Introduction

The usage of personal mobile devices in everyday life has increased dramatically over the past decade. For example, quantitatively the average US adult now spends over 1.5 h daily on their smartphone [1]. Given this ubiquity, the quality of mobile experiences is important for both users (that want to maximize their own utility) and mobile network operators (that want to optimize their service offerings to increase customer satisfaction and prevent churn). Research into the quality of mobile experiences falls into the research domain of mobile Quality of Experience (QoE). A domain with roots in the areas of mobile networks.

Specifically, the mobile QoE domain encapsulates the theory, methods, and metrics for evaluating mobile services in a way that aligns with how end-users actually experience those services. For example, the use of non-linear scales, such as mean opinion score, and contextual information in QoE aligns with the non-linearity and contextual dependence of human perception systems. QoE contrasts with the quality of service (QoS) domain which, nowadays, encapsulates technical and often more easily accessible network measures such as throughput, delay, and loss.

Given these related domains, significant research has focused on modeling QoE via network QoS and/or contextual data [2–6]. Furthermore, such QoE modeling research has been profitably performing field-based experiments with device-based methodologies (data is collected directly from a mobile device through an app).

Similarly to prior work in this area, we also examine the relationship between QoS measurements and QoE evaluations through modeling. However, existing device-based QoE modeling research has limitations regarding the level of realism,

* Corresponding author.
E-mail addresses: eren.boz@aalto.fi (E. Boz), benjamin.finley@aalto.fi (B. Finley), antti.oulasvirta@aalto.fi (A. Oulasvirta), kalevi.kilkki@aalto.fi (K. Kilkki), jukka.manner@aalto.fi (J. Manner).

https://doi.org/10.1016/j.pmcj.2019.101039

sample size, and network quality and therefore may over-estimate the practical performance of such models. Specifically, prior studies have instructed users to perform predefined application (app) actions, such as scrolling the Facebook news feed; thus the situations are only semi-realistic. Additionally, sample sizes in terms of users and sessions have typically been small (less than 50 users and 1000 sessions). Finally, several of the studies did not include significant sessions over LTE or Wifi connections, thus providing only a limited view given the widespread deployment of LTE and Wifi in most developed countries.

Therefore, in this work, we model and analyze mobile (including LTE, Wifi, and HSPA+) app-level QoE data collected from a large (292 users with 64036 experience ratings), diverse group of participants during Summer 2017. The participants installed a custom measurement app that both monitors network quality and prompts participants to rate their experiences with other apps during or after usage of those other apps. Therefore the method is a version of the experience sampling method [7,8]. Additionally, participants were instructed to use their devices normally (without any pre-defined instructions or actions), thus facilitating realistic interactions.

We perform both a descriptive statistical analysis and machine learning, specifically random forest (RF) and support vector machine (SVM), based modeling with a focus on imbalanced dataset learning.[1] Compared to prior work, this greater focus on imbalanced learning including using imbalanced learning specific metrics (rather than less suited general metrics) is theoretically and practically advantageous. For example, general accuracy as a metric can greatly reduce the impact of the minority class (which is of significant interest) [9].

The descriptive analysis results show that the ratings cover a diverse range of app categories, network types (Wifi, LTE, HSPA+, etc.), network conditions, and temporal contexts; though the overall network quality (e.g., download (DL) throughput) is also shown to be significantly better than prior similar studies (that did not include significant LTE or Wifi). This difference reinforces that our prediction problem is substantially different from previous studies. Specifically, the complex non-linear relationships in QoE imply that prediction performance cannot simply be extrapolated to new quality levels (as, for instance, new network technologies are deployed). The analysis also hints at different types of user and app adaptation to network quality.

In terms of modeling results, quantitatively the models improve over baseline naive models by about 20% in terms of well-known prediction metrics (G-Mean). Additionally, many of the important features of the models are non-network features such as a user's smartphone usage in years and the type/quality of the mobile app. Finally, and intuitively, the features that are important for one type of mobile app are not necessarily important for other types of apps. Overall the modeling results illustrate that the prediction of mobile QoE is a difficult problem especially in a context where the general network quality is high. The results have implications for both QoE theory building and the QoE prediction problem in a practical sense. We also publicly release an anonymized version of our dataset under an open non-commercial license (see Section 3.1), thus facilitating future analyses by other researchers.

In practice, mobile QoE prediction models could be used by a variety of stakeholders including network operators, mobile app/platform providers, and regulators. For example, network operators can use such models for improving network configuration and network bench-marking with a goal to optimize human-centric QoE measures rather than technical QoS measures. While, app and platform providers could improve proactive suggestions (e.g., close background apps, switch to cellular) to users suffering QoE issues. Finally, regulators could use the models to help determine their universal service requirements[2] to ensure all citizens have adequate QoE for emerging public teleservices (e.g telemedicine).

In sum, our main contributions are as follows:

- We analyze a significantly larger and more realistic (in terms of both user actions and network technologies) dataset of mobile app experience ratings than previous work. We also publicly release an anonymized version of the dataset.
- We perform both descriptive analysis and ML modeling of the dataset and detail modest performance improvements over baselines. Though we also illustrate the difficulty of mobile QoE prediction in the context of current high quality networks. For the ML modeling we use an imbalanced learning approach that has advantages over prior work.
- Finally, we discuss the potential reasons for the prediction difficulty and the implications for such modeling given future trends.

The remainder of the paper is organized as follows. Section 2 details related studies, while Section 3 describes the dataset and methodology. Finally, Sections 4, 5, 6, and 7 discuss general descriptive results, detail the modeling results, provide general discussion, and conclude the paper respectively.

2. Related studies

Related work can be classified into studies that perform similar *in situ* device-based network measurements, studies that model and analyze mobile app QoE, and studies that do both of the above.

[1] The imbalance ratio between our normal and bad QoE modeling classes is about 26.

[2] Currently Finland has a universal service requirement of a 2 Mbps broadband connection to all residents.

2.1. In situ device-based network measurements

The MopEye [10], Haystack [11], and AntMonitor [12] approaches use the device-based Android virtual private network (VPN) APIs to inspect per-app sent and received traffic and thus measure network quality. Their collected network quality metrics are similar to our metrics including delay and throughput.

However, in contrast to our methodology, these approaches can associate network measurements to specific app traffic, though this comes at the expense of significant computational and energy overheads (due to the need to maintain an additional IP network stack). Additionally, these approaches occupy the device's VPN slot, thus preventing users from using an actual (remote) VPN for privacy. All the studies also include characterization of network quality for a group of test or crowd-sourced users. However, these characterizations do not include QoE as their focus is on presenting the approaches.

Similarly, Qualia [13] uses a device-based QoE monitoring approach that collects a variety of quality metrics including contextual, network, and device based. This collection method is very similar to our methodology; however, the study does not include a characterization of QoE or validation for actual users.

Finally, [6] uses device-based network measurements to characterize network quality including delay and throughput and collects user surveys of general network satisfaction. They then model network user satisfaction via these measurement results. However, their focus is on a general network level rather than an app level and uses (primarily) user driven active measurements rather than passive measurements. Therefore, their approach cannot capture more granular app level QoE relationships.

2.2. General mobile app QoE

An array of studies have looked at general mobile app QoE, though as previously mentioned the vast majority of these have been laboratory studies. For example, [3] modeled mobile QoE for web browsing, file upload, and file download based on data collected from 108 users in controlled laboratory experiments. The network quality metrics include throughput and delay. Ref. [4] similarly modeled mobile QoE based on laboratory experiments but examined at a variety of specific popular mobile apps including YouTube, Facebook, and Google Maps.

2.3. Mobile app QoE and in situ device-based network measurements

The most closely related study is [5]. Ref. [5] used *in situ* passive device-based network measurements and user experience ratings to model the relationship between network quality (and non-network features) and QoE. We reiterate several significant differences. Ref. [5] instructed users to perform specific in-app actions during the measurement period, whereas we do not instruct users to perform specific actions but instead instruct users to use their devices normally.[3] Additionally, our participant group is 292 users with about 64036 experience ratings, compared to 30 users and 700 ratings, and we include LTE and Wifi sessions in our analysis compared to primarily only 3G and 2G sessions. Finally, we use a star-based rating system rather than a MoS based system.

Ref. [2] used passive network measurements for modeling QoE-metrics with a focus the two specific app domains of Video and VoIP. In contrast to our approach, they use QoE-metrics such as PESQ-MOS for VoIP rather than actual subjective rating collected from users.

3. Dataset collection

The dataset was collected from a group of Finnish Android smartphone users that installed a custom measurement app known as the Aalto QoE app and hereafter the QoE-Client. The QoE-Client measures network quality during usage of other mobile apps and occasionally shows a notification prompting the user to rate their experience with the specific app (denoted as an experience rating). In other words, the QoE-Client is a network monitoring client that also employs experience sampling. Experience sampling is a widely used social science method frequently applied in mobile contexts [14]. We first discuss the network measurement aspect of the QoE-Client in Section 3.1 and the experience sampling aspect of the QoE-Client in Section 3.2.

3.1. QoE-Client network measuring method

The QoE-Client network measuring method is based on a hybrid methodology that includes both passive and active measurements and is an extension of the popular measurement app Nettitutka[4] (formerly Netradar).

Specifically, the QoE-Client continuously performs passive network measurements whenever the display is on. Such passive measurements are mainly comprised of network interface level traffic samples and contextual information such

[3] Essentially, in comparison to [5] we trade off some level of control (and thus accept greater variance) in exchange for more realistic usage patterns.

[4] https://play.google.com/store/apps/details?id=fi.aalto.comnet.

Table 1
Comparison of the dataset to related datasets from prior work.

Source	Participants	Ratings	Year	Length	Network Tech[a]
This work	292	64036	2017	2–4 weeks	LTE 41%, Wifi 34%, HSPA+ 21%
[5]	30	∼700	2015	2 weeks	HSPA+ 72%, UMTS 12%, EDGE 11%
[2]	4	∼1000	2012	∼3 months	–[b]

[a]The fraction of ratings for different network technologies. Only technologies with more than 10% of measurements are included.
[b]The number of ratings per network technology were not disclosed.

as the foreground app, location, network type, and signal strength. Additionally, the QoE-Client employs active latency measurements during specific interesting moments based on a set of heuristics. We note that continuous active latency measurements would overtax battery life and bandwidth.

The main heuristic leverages the average packet size of ongoing network traffic and the fact that bulk data is almost always transmitted through maximum sized packets. Specifically, the QoE-Client triggers active measurements when large packet size traffic is detected in order to estimate the network capacity. Additionally, the QoE-Client employs mechanisms such as dynamic thresholds with exponential back-off to avoid excessive active measurements. Furthermore, the QoE-Client also triggers active measurements in cases of threshold back-offs, interactive traffic or potential connectivity loss. Though active measurements are still sparse compared to overall usage, thus the QoE-Client does not always detect highly transient connectivity issues. We also note for reference that the latency measurements ping a server near Helsinki, Finland and also capture variations in access network delays. However, overall delay measurements are inflated because many delay measurement samples coincide with larger bulk transfers in which queuing delay often dominates other delays.

Finally, we note that, due to Android limitations the QoE-Client cannot make network measurements for only the traffic of a specific app. In other words, the network measurements are measurements for the entire network interface rather than, for example, measurements for only the TCP flows of a specific app.[5] We discuss the assumptions and implications of this limitation further in Section 3.5.2.

We also publicly release a version of the dataset under an open non-commercial license.[6] The public dataset is essentially a less processed version of the dataset used in the analyses. Specifically in the public dataset, we only aggregate the demographic variables (for privacy reasons) and leave the non-demographic variables disaggregated. We also do not apply the data filtering and feature engineering steps 2 through 6 denoted in Table 8. This allows for more flexibility for future researchers to apply their own aggregations, filtering, and feature engineering. Finally for reference, Table 1 compares our dataset to related datasets from the prior work of Section 2.3. As far as we know the other datasets are not publicly available online.

3.2. QoE-Client experience sampling method

The QoE-Client experience sampling method is based on a compromise methodology that attempts to balance variety of different objectives.

For example, an important issue involves the timing and frequency of prompting users for experience ratings. The QoE-Client algorithm attempts to balance the different objectives including avoiding user fatigue and obtaining a significant number of experience ratings from a diverse set of apps, network conditions, and user contexts. A rough pseudocode of the algorithm is detailed in the Appendix in Algorithm 1. The thresholds and constants in the algorithm were chosen by iterative experimentation to, as mentioned, balance our objectives. However, it is important to emphasize that the sampling algorithm tightly interacts with the measurement heuristics. For instance echo loss is derived from latency measurements which we perform in scenarios like interactive traffic (high traffic rates in both directions) or potential connection loss (no downlink traffic even with uplink traffic). We also note specifically that network-lite apps are less likely to trigger active latency measurements, and therefore also less likely to prompt experience ratings. In other words, prompting for experience ratings is more likely for network-intensive apps. Though since we have a main focus on network QoE parameters, we do not view this as a significant problem.

In terms of the actual experience rating interface, Fig. 1 illustrates an example experience rating. The use of a five-point star scale is based primarily on both the prevalence of the scale in everyday use [15] and display size considerations. Additionally, given the prevalence of the scale, users do not need to be trained to use the scale in contrast to the mean opinion score scale.

5 We look to evaluate alternative network measuring approaches such as the device-based VPN approach [10–12] in future work.
6 https://userinterfaces.aalto.fi/qoeapprating/.

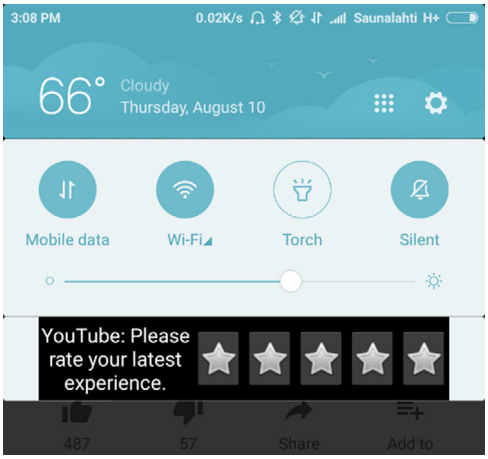

Fig. 1. Screenshot of an experience rating (for YouTube) from the QoE-Client.

Table 2
Comparison of demographics between participants and Finnish smartphone users.

Demographic	Participants	Finnish smartphone users[a]
Young participants (% 19–32 years old)	67.80	25.56 (1.90)
Gender (% male)	47.44	50.26 (2.00)
Home region (% Uusimaa)	67.12	32.64 (1.92)
Smartphone usage[b,c] (years)	6.88 (3.07)	–
Sub limited[d] (% yes)	44.03	–
N	292	4,237,310

[a]Finnish smartphone user demographics are from the Finnish sub-population that reported owning a smartphone (n = 713) from Eurobarometer 87.1 survey of March 2017 [16]. Proportions also include linearized standard errors for population estimates.
[b]Number of years the participant has used a smartphone.
[c]Means also include standard deviations.
[d]Whether the participant's subscription is limited to a maximum DL throughput. In Finland, many mobile subscriptions are differentiated by artificial maximum throughput caps rather than data caps. Note that these throughput caps are not the same as the simple technological limits of certain network technologies (for example HSPA, LTE, etc.).

3.3. Participant recruitment, representativeness, and procedure

The participants were recruited through several different media channels including email lists, word-of-mouth, online ads, and newspaper ads to acquire a diverse group. Some ads were specifically targeted toward media channels of rural areas to attempt to recruit such users under the assumption that their network quality might be lower than urban users.

In terms of representativeness, Table 2 compares the demographics of the participants (collected through the exit survey) to Finnish smartphone users in general. Despite attempts to acquire a diverse participant group, the participant demographics are still skewed toward younger users in the Uusimaa region.[7] Therefore, this discrepancy should be considered when making generalizations.

Finally, in terms of procedure, each recruited participant was directed to a web page that provided instructions (text and video) on installing the QoE-Client and registering for participation.[8] Each participant was rewarded with two movie tickets[9] if the participant kept the QoE-Client installed for at least two weeks, completed at least 60 total experience ratings, at least 50% of all prompted experience ratings, and an exit survey. Hereafter we limit our analysis to the dataset of participants that met these conditions (unless otherwise noted) and refer to them simply as the participants. Overall, the dataset consists of 292 participants that completed 64036 experience ratings. In terms of privacy, the most personally identifying information from the exit survey (name and email address) were used only for providing the movie tickets and subsequently deleted.

[7] The Uusimaa region includes Helsinki and surrounding municipalities including Espoo which contains Aalto University.
[8] http://emergent.comnet.aalto.fi/?page_id=108.
[9] An approximate value of 29€.

3.4. Supplementary app store data

Additionally, we crawl the Google Play app store[10] (hereafter Play store) for data related to the specific app evaluated in each experience rating. Specifically, we collect Play store category, Play store mean review score (scale of 1 to 5), number of Play store reviews, and number of Play store downloads (minimum of the download range). We are able to collect such data for 98.2% of experience ratings. The experience ratings that are missing data are primarily for pre-installed android and device vendor apps that are not available in the Play store.

3.5. Data filtering and feature engineering

We then perform a series of data filtering and feature engineering steps which are described in Sections 3.5.1 and 3.5.2. These steps are summarized in Appendix Table 8 including the alternative parameters we tested in these different steps.

3.5.1. Experience rating quality filtering

The quality of participant experience ratings is potentially influenced by unreliable participants. Unfortunately, methods such as gold standard data and manual re-checking of evaluations by the researchers [17] are not applicable in our case given the uncontrolled nature of the evaluation task. Additionally, any kind of control group (of known reliable participants) is too small to provide a reliable comparison.

We can, however, remove potentially unreliable ratings by filtering out ratings where the time between the session (to be rated) and the actual rating selection is significantly large. In other words, ratings that are unreliable because too much time has passed between the event (session) and the evaluation (rating). We use a semi-arbitrary and generous threshold of 900 s. We also test alternative thresholds (see Table 8) and do not find substantial differences. Therefore we only present the results using the 900-second threshold.

Additionally, we filter out ratings performed more than four weeks after the initial installation and first rating. This filtering helps ensure that all users have a roughly similar number of ratings by removing some ratings for users that kept rating for longer than four weeks. Without this filtering, a few users would have over 1000 ratings (compared to a user median of 167) and could overly influence the results.

Finally, given the possibility of user fatigue over the experiment period, we also test several data subsets including a first-week subset (includes only data from the first week a user is in the experiment). We find no substantial differences, therefore we only present the results from the four week period

3.5.2. Feature engineering

The first and most obviously required feature engineering is the mapping and aggregation of the network measurements (which as mentioned occur whenever the display is on) to specific app sessions and therefore to experience ratings. As mentioned previously, due to Android limitations the network measuring method cannot associate network measurements to the traffic of a specific app. Therefore, we reasonably assume the majority of data traffic observed during any given measurement originates from the foreground app.

However due to mechanisms such as prefetching, often the current network measurement is unrelated to the content that the user is actually consuming in that app at the moment. Consequently, an experience rating should be associated with not only a single measurement but a history of measurements. In doing so, each numeric feature can be calculated over a given lookback period (a time period before a given experience rating, e.g., 1 min, 5 min, 1 h, etc.) and an aggregation method (e.g., mean, median, std. dev., etc.). For example, DL throughput with a lookback period of 5 min and an aggregation method of mean would, for each experience rating, calculate the mean DL throughput of passive measurements for the 5 min period before the rating. We only present results from features aggregated with a lookback period of 1 h, though we also test additional periods (see Table 8) and do not find substantial differences.

Additionally, each feature can be aggregated over every measurement in the lookback period, regardless of the foreground app during that measurement, or only for measurements where the rated app is the foreground app. We only present results from features aggregated over measurements where the rated app is in the foreground. Though we also test the alternative method (see Table 8) and do not find substantial differences.

In terms of further feature engineering, the user (experience) rating (the dependent variable in our modeling) is first normalized for each user by subtracting the median rating of that user. This normalization helps mitigate differences in usage of the scale between users. The resulting normalized ratings have a range of $[-4, 4]$. We then binarize the normalized ratings into $[-4, -2]$ representing a Bad-QoE class (in other words a significant degradation from the user's normal experience) and $[-1, 4]$ representing a Normal-QoE class. We also test a normalization based on the median rating for a user with a specific app (user-app combination), however, we do not find substantial differences. Therefore we only present the median normalization. Additionally, we also include the absolute (non-normalized) user rating in the descriptive analysis.

We also map each app into a meta-category based on the app's Google Play store category (since the Play store categories are too granular). The mapping is given in the Appendix in Table 7 and the final meta-categories are games, entertainment, social, communication, maps, productivity, and other.

Table 3 describes the final list of features including each feature's type and source.

[10] We use version 3.3.1 of google-play-scraper from https://github.com/facundoolano/google-play-scraper.

Table 3
Final features after feature engineering and preprocessing.

Feature	Describes	Type	Source
DL throughput (Mbps)	Network	Numeric	Device API + UDP probing
Delay (ms)	Network	Numeric	UDP probing
Echo loss ratio (%)	Network	Numeric	UDP probing
DL bulk max[a] (Mbps)	Network	Numeric	Device (trafficstats) API
Signal strength[b] (%)	Network	Numeric	Device API
Network type[c]	Network	Categorical	Device API
Sub limited	Network	Categorical	User survey
Device memory available (GB)	Device	Numeric	Device API
Play store mean review score	App	Numeric	Google Play store
Play store num reviews	App	Numeric	Google Play store
Play store num downloads	App	Numeric	Google Play store
Meta-category	App	Categorical	Google Play store
Hour of day	Context	Categorical	Device API
Weekday/Weekend	Context	Categorical	Device API
Age[d]	User	Categorical	User survey
Gender	User	Categorical	User survey
Home region[e]	User	Categorical	User survey
Smartphone usage[f] (years)	User	Categorical	User survey

[a]DL bulk max is the maximum DL throughput during bulk data transfers over the measurement session. In comparison to DL throughput, DL bulk max gives an estimate of application throughput regardless of bottleneck being server or network.
[b]Signal strength is normalized (to [0,100]) for different mobile network technologies via the following [min,max] intervals: $[-21, -2]$ for LTE RSRQ, $[-120, -50]$ for WCDMA receive power, $[-100, -20]$ for Wifi RSSI.
[c]Network type is categorized into the following categories LTE, Wifi, HSPA+, other cellular, and unknown.
[d]Age is categorized into the following intervals $[16-25]$, $[26-40]$, and $[40+]$.
[e]Home region is categorized into the following categories Uusimaa and non-Uusimaa.
[f]Smartphone usage years is categorized into the following intervals $[0, 4]$, $[5, 7]$, and $[8+]$.

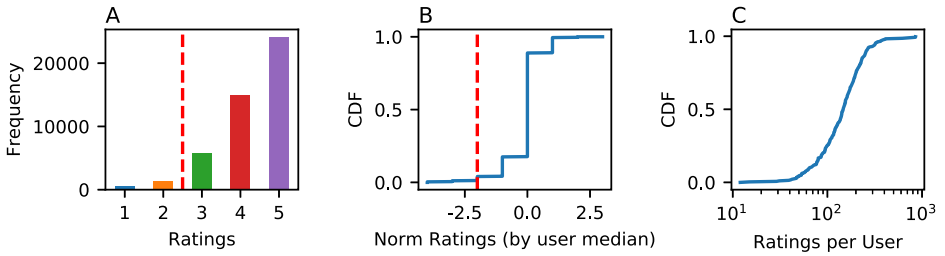

Fig. 2. (A) Distribution of absolute ratings, (B) Cumulative distribution function (CDF) of normalized ratings (by user median) with dashed line representing a binarization for modeling, (C) CDF of ratings per user.

4. Descriptive analysis

We perform a broad descriptive analysis including statistics, cumulative distributions, and cross-correlations of the features and ratings.

4.1. User ratings

First, in terms of the user ratings, Fig. 2 illustrates the distribution of absolute and normalized ratings and number of ratings per user. As discussed, the rating distributions are highly skewed toward higher ratings in both the absolute and normalized cases. Such distributions have been observed in similar previous studies such as [5]. Additionally, the number of ratings per user is typically moderate with a median of 167 and only a few users with more than 300 ratings. This suggests a good spread of ratings across users.

4.2. Network related features

For network features, Figs. 3 and 4 illustrate the cumulative distribution functions (CDFs) or frequencies of such features. In general, the high median DL throughput of 13.09 Mbps helps partly explain the large fraction of higher ratings. Laboratory studies, such as [4], have suggested that many of the most popular mobile apps only require 8 Mbps for QoE saturation. Similar arguments can be made for delay and loss with medians of 51.09 ms and 0% respectively. In other words, in networks with quality beyond a certain threshold, further quality improvements will not affect a user's QoE. In such cases, non-network features such as contextual, device, and task-related features likely become more important.

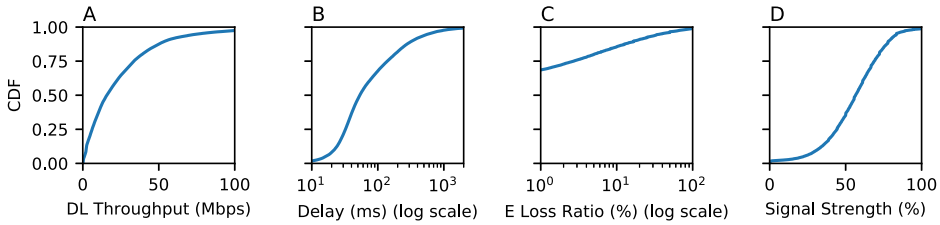

Fig. 3. CDFs of several network and device features (A) DL throughput, (B) delay, (C) echo loss ratio, and (D) normalized signal strength.

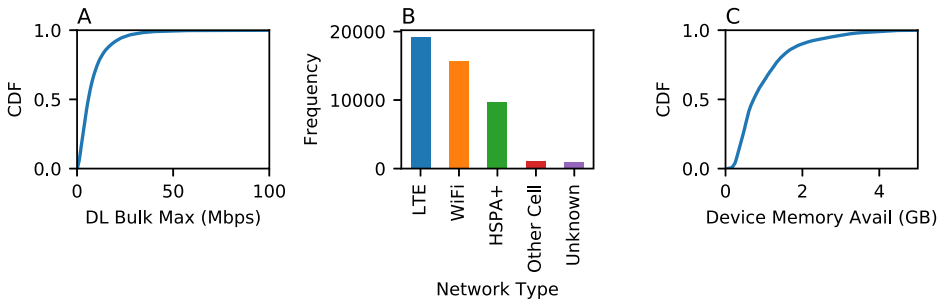

Fig. 4. CDFs/distributions of several network and device features (A) DL bulk max, (B) network type, and (C) device memory available.

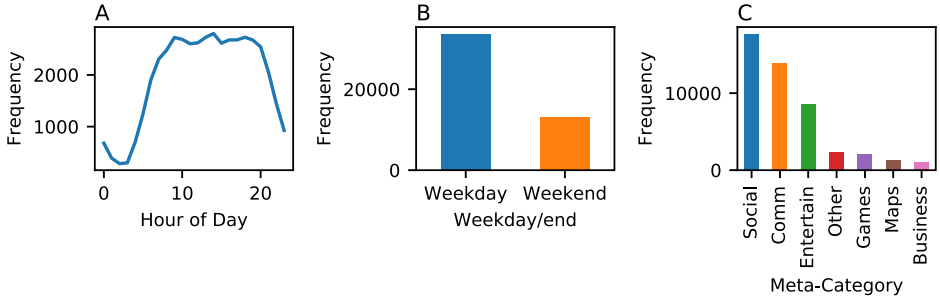

Fig. 5. Distributions of categorical features (A) hour of day, (B) weekend, and (C) meta-category.

4.3. Device related features

For device-related features, Fig. 4 illustrates the device memory available during each rated app session. As seen, the device memory available during different ratings varies significantly (from about 128 MB to over 3 GB) suggesting that device memory might be a potentially important factor. Specifically, apps running on devices with low memory may stutter or freeze more often due to, for example, frequent background garbage collection. Android currently supports smartphones with a total memory as low as 512 MB, though the clear and quantitative effects of available device memory on app quality is highly situational and difficult to study.

4.4. Temporal and spatial context features

Fig. 5A and B illustrate the hour of the day and weekday/weekend frequency of the ratings. The distributions indicate clear and expected diurnal behavior. Additionally, they suggest that our ratings adequately span daily smartphone behavior.

4.5. Task context/media related features

Fig. 5C illustrates the frequency of different app meta-categories for the rated apps. We similarly find that the ratings span many meta-categories and in rough proportion to the overall usage of each meta-category. Fig. 6 illustrates the distributions of the Google Play store features for rated apps. We find over half of the ratings are for extremely popular apps with over one billion downloads, though we also find some ratings for apps with as few as 1000 downloads. The Play mean review scores are less diverse with almost all apps having a score greater than four.

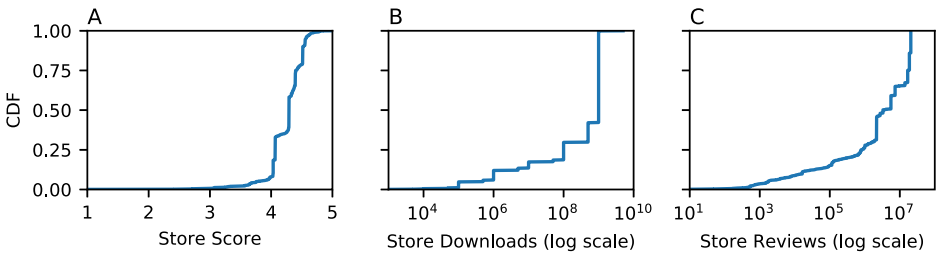

Fig. 6. CDFs of several app features from Google Play store (A) Play store mean review score, (B) Play store number downloads, and (C) Play store number reviews.

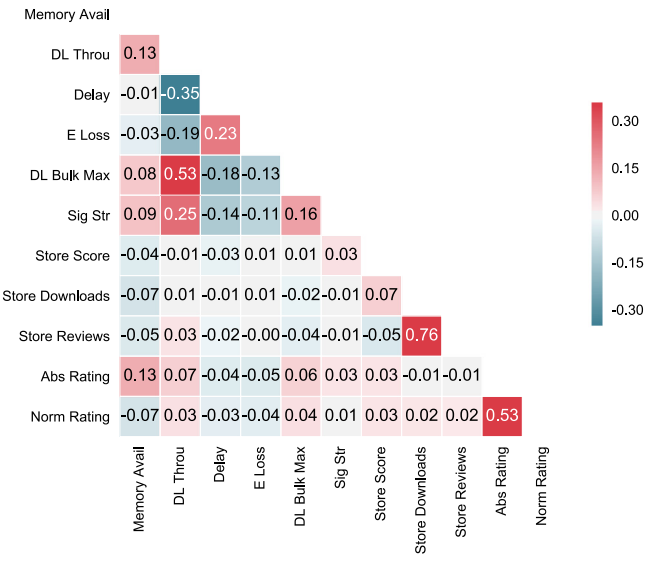

Fig. 7. Spearman correlation matrix of numeric features and absolute and normalized ratings.

4.6. Demographic features

The demographic features are summarized mainly in Table 2 of Section 3. These features include both low-level features such as gender and age, as well as high-level features such as previous experience (in terms of years of smartphone usage). As previously mentioned, we find significant diversity in many of these features.

4.7. Rating and feature cross correlations

Finally, in Fig. 7, we illustrate the Spearman correlations between all numeric features and absolute and normalized ratings. We find that the correlation magnitudes between the numeric features and the normalized ratings are low (all $\rho < 0.08$) suggesting that in the prediction task no single feature will be highly useful. Unsurprisingly, the highest overall correlation is between the Play store number of downloads and Play store number of reviews features ($\rho = 0.76$). Given this very high correlation, we exclude the Play store number of reviews feature from the modeling since very high feature correlations can negatively impact the feature importance calculation.

4.8. User-level descriptive analysis

We also analyze descriptive statistics at a user level whereby all of a user's ratings and features are aggregated (through a function such as the mean). We denote features aggregated through the mean/median as *user-level mean/median* and features aggregated through the Spearman correlation (which aggregates two features) as *user-level correlation*.

In terms of user-level correlations, Fig. 8 illustrates the CDFs of user-level correlations between normalized ratings and three different network features. The significant diversity in the correlations suggests that the low correlations are both an intra-user and inter-user phenomenon. Additionally, significance tests indicate that for each of the three network

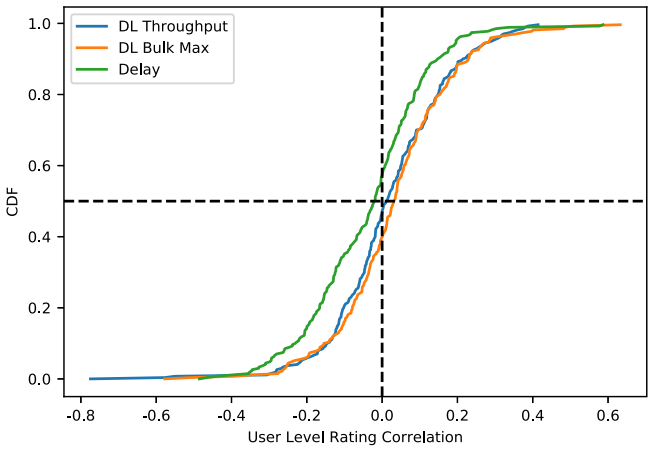

Fig. 8. Spearman correlation distributions at user level.

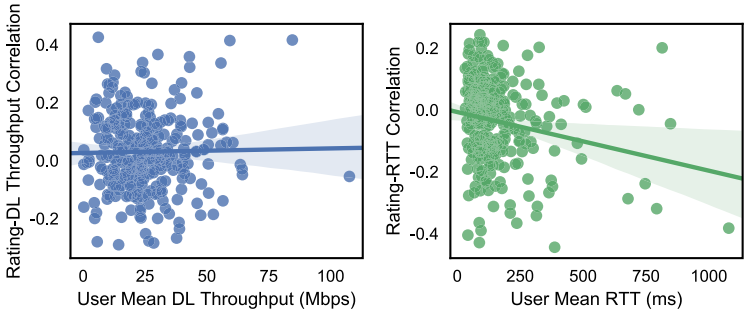

Fig. 9. DL throughput and delay Spearman correlations versus normalized rating at the user level.

feature only between 5% and 16% of users have both the intuitively correct correlation sign and a statistically significant correlation ($p < 0.05$). This further reinforces that the results are not simply due to a small abnormal user group.

Fig. 9A details the user-level correlations between DL throughput and normalized rating versus the user-level mean DL throughput. Interestingly, we find no clear significant relationship ($\rho = -0.04$, $p = 0.53$), even for users with low mean DL throughputs of less than 8 Mbps ($\rho = 0.06$, $p = 0.79$). One possible explanation is that these low DL throughput users adapt their usage (and expectations) to the low throughput environment (either manually or through apps that automatically adapt).

In comparison, Fig. 9B shows the user-level correlations between delay and normalized rating versus the user-level mean delay. The relationship is clearly stronger and more significant ($\rho = -0.22$, $p = 0.0002$). This significant relationship is potentially because adapting delay-sensitive apps to higher delays is generally more difficult than adapting throughput-sensitive apps to lower throughputs. For example, the Skype video system can adapt to bandwidths ranging from 15 to over 2000 Kbps but requires a delay of less than 150 ms and, understandably cannot adapt to higher delays given the interactive nature of the app.[11]

Relatedly, users can also adapt to different network conditions by switching mobile networks from cellular to Wifi and vice versa. Fig. 10 illustrates, on the user level, a positive relationship between the ratio of median DL throughput for Wifi/cellular and the ratio of the number of sessions between Wifi/cellular. Though lurking variables or factors may play a role, such adaptation is likely especially in the case of subscriptions with low artificial (rather than technical) throughput limits.

Additionally, at the user level, we can analyze the relationship between these aggregate features and the user demographics collected from the participant exit survey. Fig. 11 illustrates the mean absolute and normalized user rating for different age groups. We find that absolute rating tends to decrease with age, but normalized rating stays roughly the same. This suggests that older users might be using the rating scale differently (potentially using the center three-star rating as a baseline rather than the five-star rating). In further support, as Fig. 11 also illustrates, we do not find clear patterns for network quality differences between age groups.

[11] https://docs.microsoft.com/en-us/skypeforbusiness/plan-your-deployment/network-requirements/network-requirements.

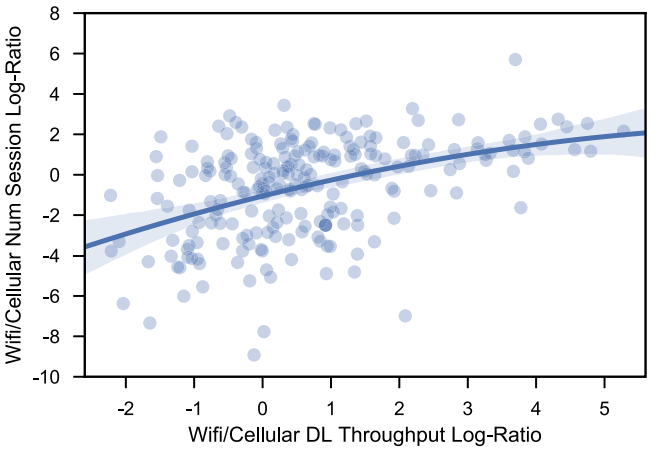

Fig. 10. Ratio of median DL throughput for Wifi/cellular versus the ratio of the number of sessions between Wifi/cellular.

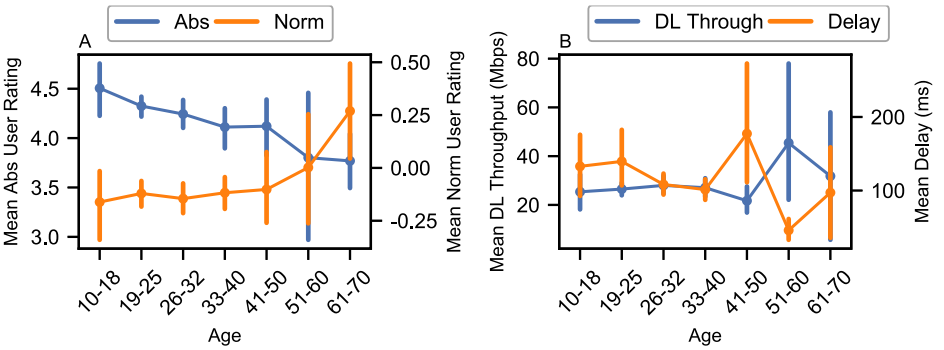

Fig. 11. (A) Mean absolute and normalized user rating for different age ranges, and (B) Mean DL throughput and delay for different age ranges.

Additionally, we did not find significant differences in terms of ratings or features for different genders, smartphone usage years, or home region.

5. Modeling

In this section, we perform supervised learning for predicting the normalized QoE classes. Specifically, we perform two concrete modeling tasks. First, we train and evaluate a general model that includes all app meta-categories and predicts either the Normal-QoE or Bad-QoE class (binary classification). Second, we train and evaluate a separate model (also Normal-QoE or Bad-QoE class prediction) for each specific meta-category. These meta-category specific models help quantify the potential performance on these diverse meta-categories.

For additional context, we first describe the metrics for evaluation of the models, the evaluated model types (e.g., RF, SVM), the training procedure, and the baseline models for comparison. Finally, we detail the modeling task results in Sections 5.5.1 and 5.5.2.

5.1. Metrics

For evaluation, we use several common imbalanced learning metrics including G-Mean, index balanced accuracy (IBA) of G-Mean, and F_1 [9]. G-Mean is the geometric mean of sensitivity (true positive rate) and specificity (true negative rate). Therefore, G-Mean, and essentially all binary imbalanced learning metrics, measures the balance (or trade-off) between positive and negative accuracy. We detail the formulas for these metrics below where TP is the number of true positives (the model predicts positive and observation is positive), TN is the number of true negatives (the model predicts negative and observation is negative), and so on.

$$TP_{rate} = \frac{TP}{TP + FN} = \text{Recall/Sensitivity/Bad-QoE Accuracy}$$

$$TN_{rate} = \frac{TN}{TN + FP} = \text{Specificity/Normal-QoE Accuracy}$$

$$PP_{value} = \frac{TP}{TP + FP} = \text{Precision}$$
$$G_{mean} = \sqrt{TP_{rate} \times TN_{rate}}$$
$$IBA_{G_{mean}} = (1 + 0.10 \times (TP_{rate} - TN_{rate})) \times G_{mean}$$
$$F_1 = \frac{2 \times TP_{rate} \times PP_{value}}{TP_{rate} + PP_{value}}$$

The IBA of G-Mean is a lesser known metric that puts a greater emphasize on the TP rate (as compared to TN rate) through the use of a dominance measure ($TP_{rate} - TN_{rate}$) in conjunction with G-Mean [18]. As previously mentioned, the G-mean itself as a simple geometric mean does not favor TP or TN. Thus the IBA of G-Mean helps in cases where, as we also could assume, that the TP rate is somewhat more important.

5.2. Model types

We train two well-known tree model types: RF and gradient boosted forest (GBF) (with regression trees as weak learners) through the sklearn library [19]. Given the features, the considerations when selecting a model type include highly non-normal features, non-linear relationships, potential feature interactions, and categorical features. In contrast to many other model types, most tree-based model types handle all of these issues natively. For additional reference, we also train an SVM model through the sklearn library.

5.3. Training and testing

For training and testing, we use group splitting validation (with 50 independent splits[12] of 80%/20% for training/test folds) with grouping at the individual user level. In other words, for any given split, no data from the same user are in both the training and testing folds. Additionally, we apply a random under-sampling (RUS) method [20] to each training fold (though crucially not to the test fold) to balance the training data. Balance is necessary to allow the model to learn to predict the infrequent Bad-QoE class. A weighted sample learning approach is also tested but we exclude the results since they are very similar to the RUS.

We also apply a backward feature elimination[13] approach to remove irrelevant features from the model. Specifically, starting with a full model, the feature with the lowest feature importance (calculated as out-of-bag permutation importance of the G-Mean) is removed and the resulting model is retrained. We use a permutation-based importance rather than a traditional Gini impurity importance because such impurity importances are biased in many cases (such as with categorical variables) [21]. Finally, the model with the highest test G-Mean over a conservative sweep of the hyper-parameter space is reported.

We note that for the SVM model we standardize each numeric feature (to zero mean and unit variance), one-hot encode each categorical feature, and use a Gaussian radial basis function kernel. We also do not use feature elimination for this model because the one-hot encoding makes feature elimination problematic in the ML library (sklearn).

5.4. Baseline models

We compare the model results to several baseline (dummy) models including uniform class prediction (predict each class with equal probability) and stratified class prediction (predict each class in proportion to their frequency in the training data). We also compare against a baseline model that uses a popular unsupervised tree-based anomaly detection method known as isolation forest [22]. As an anomaly detection method isolation forest (IF) is also considered useful in cases of large class imbalance.

5.5. Model results

5.5.1. General model with all app meta-categories

As mentioned prior, the general model task is a binary classification (Normal-QoE or Bad-QoE class) that includes all app meta-categories. In term of evaluation, Table 4 details the test results for each model type, including the real and baseline models. Interestingly, we find that the models provide a significant but small advantage over the baselines with only about a 20% higher G-Mean. Additionally, the TP_{rate} and TN_{rate} of 55% and 66% (for RF) are much smaller than about 95% rates found in the previous semi-realistic study of [5]. A potential reason is that [5] does not explicitly mention *group* cross-validation, suggesting user's ratings might be in both the training and test sets. In other words, the reported performance might not represent the generalizability to new users but to a static group of users over time. This problem

[12] Group splitting validation is used instead of simple cross-validation because the group size and label imbalances mean that we need both a large test fraction and many trials to acquire good performance estimates. In any case, we note that with 50 independent splits the probability of all ratings being in at least one test fold is >99.4%.

[13] To maximize performance such irrelevant features need to be eliminated because at each tree node only $\sqrt{n}$ features are checked.

Table 4
Mean test results for general models (real and baselines).

	RF[a]	GBF[b]	SVM[c]
G_{mean}	**0.598 [0.534, 0.662]**	0.597 [0.526, 0.668]	0.586 [0.485, 0.687]
$IBA_{G_{mean}}$	**0.355 [0.274, 0.435]**	0.354 [0.264, 0.444]	0.339 [0.216, 0.463]
F_1	0.106 [0.064, 0.148]	**0.107 [0.061, 0.154]**	0.105 [0.06, 0.151]
	Baseline-UC[d]	Baseline-SC[e]	Baseline-IF[f]
G_{mean}	0.497 [0.463, 0.530]	0.182 [0.120, 0.244]	0.338 [0.276, 0.400]
$IBA_{G_{mean}}$	0.248 [0.213, 0.282]	0.031 [0.010, 0.052]	0.122 [0.081, 0.164]
F_1	0.070 [0.044, 0.097]	0.035 [0.012, 0.057]	0.063 [0.044, 0.082]

[a]Random forest model.
[b]Gradient boosted forest model.
[c]C-Support vector Machine model.
[d]Baseline (dummy) model with uniform class prediction.
[e]Baseline (dummy) model with stratified class prediction.
[f]Baseline model with isolation forest anomaly detection.

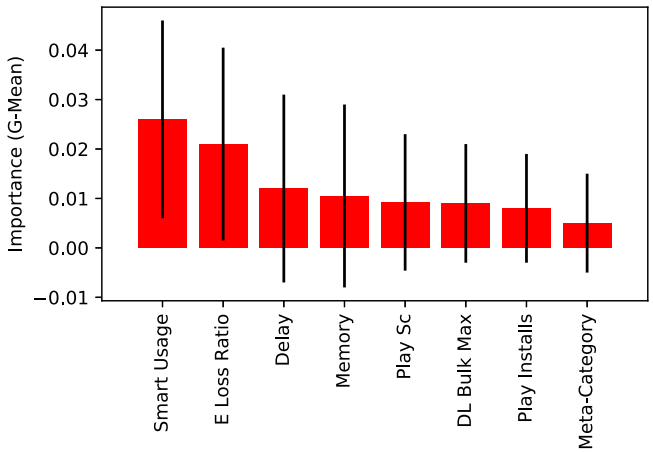

Fig. 12. Feature importances (out-of-bag permutation importance of the G-Mean) of the RF model (including standard deviations of importances).

is substantially easier because a user's ratings over time are fairly static. For illustration refer to Fig. 2B, which shows that over 70% of the ratings are at the given user's median value. Additionally, [5] analyzes network sessions with a median DL throughput of less than 1 Mbps compared to our median DL throughput of about 13 Mbps. As previously discussed, in networks with high quality the importance of network features in QoE modeling likely decreases and the importance of non-network (more difficult to measure and capture) features likely increases. Furthermore, the difficulty in measuring and capturing these non-network features likely decreases model performance.

We note that since the performance differences between the RF, GBF, and SVM models are negligible, we focus further analysis only on the RF model.

In terms of individual features, the backward feature elimination approach leaves eight features in the final RF model including device memory available, DL bulk max, delay, echo loss ratio, Play store mean review score, Play store number of downloads, smartphone usage years, and meta-category. For illustration, Fig. 12 depicts the feature importance of each model feature (as quantified by the out-of-bag permutation importance of the G-Mean).

The most important feature is smartphone usage years which potentially acts as a proxy for technological sophistication through the proclivity for early adoption. Such technological sophistication likely affects the user's expectations of quality, an important factor influencing mobile QoE [23].

We also find that the echo loss ratio is one of the most important features. This finding is intuitive given that a significant connection loss is the most egregious network problem and might affect the QoE of all connection-reliant apps. Contrastingly, given the generally high DL throughputs and low delays, even moderate changes in these features might not affect a user's QoE. Though, we note that DL bulk max and delay still have modest importance in the model.

Interestingly, we find that both the Play store number of downloads and the Play store mean review score are important features. We hypothesize that these app store features could act as proxies for general app quality. For example, these features could capture issues with app user interfaces such as glitches and poor design that are not network related. Similarly, as hypothesized earlier, available device memory is also important.

Table 5
G-mean for each meta-category specific RF model.

Meta-category	G_{mean}
Social	0.597
Communication	0.611
Entertainment	0.632
Other	0.602
Games	0.605

Table 6
Model feature importances for meta-category specific models.

Feature	Social	Comm	Entertain	Other	Games
DL throughput	–	0.01	–	–	–
Delay	–	0.01	0.02	0.05	–
Echo loss ratio	0.04	0.01	0.01		0.07
DL bulk max	–	–	–	0.06	–
Signal strength	–	–	0.01	–	
Network type	–	–	–	–	–
Sub limited	–	–	–	–	–
Device memory available	–	0.05	–	–	–
Play store mean review score	–	–	–	–	–
Play store num downloads	–	–	0.05	0.07	0.06
Meta-category	–	–	–	–	–
Hour of day	–	–	–	–	–
Weekday/Weekend	–	–	–	–	–
Age	–	0.01	0.02	–	–
Gender	–	–	0.02	–	–
Home region	–	–	–	0.03	–
Smartphone usage	0.08	–	–	–	–

5.5.2. App meta-category specific models

As mentioned prior, the app meta-category specific model task is also a binary classification (Normal-QoE or Bad-QoE class) but with a separate model for each meta-category. Thus, we examine how performance varies between different meta-categories which likely have broadly different network requirements. Unfortunately, the maps and business meta-categories have too few Bad-QoE samples for reliable modeling, therefore, we exclude these two meta-categories from the meta-category specific modeling. Additionally, for brevity we focus only on the RF model type and the G-Mean metric.

Table 5 details the G-means for these different meta-categories. We find that all meta-categories models have roughly similar levels of performance and that this level is similar to the performance of the general model. This suggests that even meta-category specific models might not be able to provide the desired performance level indicated by previous work. Though we note that, like the general model, the specific models do outperform the baseline models.

Furthermore, Table 6 details the feature importances of each meta-category model. We find that the different meta-category models have different features with no single feature included in all models. Interestingly, many of the most important features are non-network features. This reinforces the importance of attempting to account for as many features as possible. In fact, at least one non-network feature is important in each model. Quantitatively, the fraction of importance in network features compared to non-network features ranges from 33% to 54% in the models.

In comparison to [5], for our entertainment meta-category model compared to their YouTube model the only feature included and important in both these models is signal strength. While for our social meta-category model compared to their Facebook model, no feature is included and important in both.

We note though that given the modest performance of the models, we remain cautious of over-interpretation of the feature importances.

6. Discussion

Overall in terms of descriptive results, we found that the ratings cover a wide range of app categories, network types, network conditions, and temporal contexts. And that overall network quality is significantly better than in prior similar studies. Our descriptive results also suggests user and app adaptation to network quality. In terms of modeling results, quantitatively our models outperform baseline (naive) models by about 20% in terms of G-Mean. Additionally, many of the most important features of the models are non-network features and the features that are important for one type of mobile app are not necessarily important for other types of apps.

The main takeaway from these findings is that mobile QoE prediction remains a difficult problem, especially in cases where network quality is high in general. In those cases, the users' experience is affected by a number of other factors and the relationship is still poorly understood.

We hypothesize that beyond a certain threshold (hereafter: the QoE saturation threshold), a significantly larger fraction of bad QoE events is related to non-network factors such as app quality, device performance, and user expectations, etc. These non-network features are more difficult to measure and capture and thus prediction performance decreases. Under this assumption, future mobile QoE studies will need to be multidisciplinary as these non-network features are related to many different academic domains, such as user experience, consumer psychology, and economics.

In theory, the QoE saturation threshold may increase as users' expectations change with new applications; for example with the introduction of augmented or virtual reality or a change in media quality (HDR, 4K+, etc.). Alternatively, the threshold may decrease with the improved ability of apps to adapt to poor network conditions.

We can draw some initial hints if we examine these examples in more detail. For VR, delay is the main bottleneck in current LTE networks [24], and device side modifications could already allow medium quality VR on current US-based LTE networks [24]. Whereas for media quality, 1080p content has essentially already reached the limits of the human visual system in terms of resolved detail given the small displays on mobile devices. In practical network terms, for example, Netflix only requires a 5 Mbps connection for 1080p content and a 25 Mbps connection for 4K content.[14] Finally, researchers and industry groups are continually researching and standardizing new network adaptation methods including for novel use cases such VR [25,26]. Therefore, overall the threshold appears unlikely to increase significantly in the near term.

In practical terms, the role of any mobile QoE prediction model in mobile operator networks depends critically on the operator actions prompted by model predictions. These actions (ranging from technical automated network tweaks to direct customer engagement to prevent churn) will determine the costs and benefits of true positives, false positives, false negatives, etc. Understanding these costs and benefits are particularly important given the relatively high false positive and false negative rates from the modeling. Further work in quantifying these costs and benefits could also help in determining future utility of such models.

6.1. Limitations and issues

We note the following limitations in our work. Firstly we focused only on a single country (Finland) with relatively high-quality mobile networks. Therefore, the results may not generalize to countries with significantly different population or mobile network characteristics. Though we argue that many (especially European) countries are similar to Finland in these terms. Additionally, given that our QoE-Client and methodology are not country-specific, future work could include experiments in alternative countries. Relatedly, as previously mentioned, our participants are more concentrated in the capital area (Uusimaa) than the overall population of Finnish smartphone users. Thus our results are not fully representative of this population at large.

Additionally, the dataset does not include or has low coverage of several contextual features including physical device speed, inferred semantic location, inferred travel mode, etc. These features might improve the performance of the modeling as they have shown some importance in previous work [5]. We hope to include such features in future experiments.

7. Conclusion

We have analyzed and modeled mobile QoE data from a large group of Finnish users in the field. Reflecting the development of mobile networks, service quality was high and better than in prior studies. In a setting like this, our results suggest that users opinions are experiences of networks are affected much less by network quality than would be thought in the light of earlier results. Prediction of mobile QoE remains a challenging task with modest performance gains over naive baseline models. We conclude that this is due to the large number of factors that affect experience, which can be often neglected or even controlled out in laboratory studies. In this study, both network and non-network features were found to be important with no single dominant feature and differences between apps in different app categories. In the future, we expect that the development of QoE models will need to shift to more complex multi-variate models that account for user, context, device, and app-related characteristics in addition to QoS.

Acknowledgment

This work was supported by the EMERGENT Project (http://emergent.comnet.aalto.fi/).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Table 7
Meta-category to Google Play store category mapping.

Meta-category	Google Play store categories
Games	GAME, ALL_OTHER_GAME_SUBCATEGORIES, FAMILY_BRAINGAMES
Entertainment	ENTERTAINMENT, MUSIC_AND_AUDIO, VIDEO_PLAYERS, BOOKS_AND_REFERENCE, NEWS_AND_MAGAZINES, COMICS, SPORTS, SHOPPING, FAMILY_MUSICVIDEO
Social	SOCIAL, DATING, EVENTS
Communication	COMMUNICATION
Maps	MAPS_AND_NAVIGATION, TRAVEL_AND_LOCAL, WEATHER
Productivity	PRODUCTIVITY, FINANCE, BUSINESS
Other	ANDROID_WEAR, PHOTOGRAPHY, ART_AND_DESIGN, BEAUTY, AUTO_AND_VEHICLES, HOUSE_AND_HOME, TOOLS, MEDICAL, LIFESTYLE, PARENTING, PERSONALIZATION, LIBRARIES_AND_DEMO, HEALTH_AND_FITNESS, FOOD_AND_DRINK, EDUCATION, FAMILY, FAMILY_ACTION, FAMILY_PRETEND, FAMILY_CREATE, FAMILY_EDUCATION

Table 8
Data filtering and feature engineering steps.

	Steps	Alternate Params	DS[a]	ML[b]	Ratings[c]
1.	Aggregate (featurize) the network measurements performed (1) at most **60** min before the rating selection and (2) while **the rated app** is in the device foreground	(1) 1, 5, 10, 3600 (2) any app	✓	✓	64034
2.	Filter out ratings with more than **900** s between session start and actual rating selection	60, ∞	✓	✓	57931
3.	Filter out ratings performed more than **28** days after the user's initial first rating	7, 14, 21, ∞	✓	✓	46777
4.	Calculate normalized ratings by subtracting the median rating of the **user**	app, category, comb[d], none[e]	✓	✓	46777
5.	Impute missing **DL throughput** with **DL bulk max**	–		✓	46777
6.	Filter out ratings with **any** still missing features	–		✓	30821

[a]Steps applied for descriptive statistics analysis.
[b]Steps applied for modeling analysis.
[c]The number of ratings remaining after the step.
[d]A combination of rating features such as user-app or user-category combination.
[e]Do not use any normalization.

```
Data: dataSession, prevDataSessions, NumRatingsPastHour, connectivityLossFromAPI;
if NumRatingsPastHour < 2 then
    if dataSession.echoLoss > 3 or connectivityLossFromAPI < 2mins then
        wait 10 sec after dataSession;
        prompt rating;
    else if dataSession is heavyUpload then
        wait 10 sec after dataSession;
        prompt rating;
    else
        if dataSession is bandwidthBound then
            if dataSession.bandwith is not in dataSession.app.prevBandwidthsToday then
                wait 10 sec after dataSession;
                prompt rating;
        else if dataSession.latency is not empty then
            if dataSession.latency is not in dataSession.app.prevLatenciesToday then
                wait 10 sec after dataSession;
                prompt rating;
        end
end
```

Algorithm 1: Triggering of Experience Rating Prompts.

Appendix

See Tables 7 and 8 and Algorithm 1.

14 https://help.netflix.com/en/node/306.

References

[1] H. Verkasalo, Metrics that Matter: Two New Approaches from Huawei Connect Europe, http://www.vertoanalytics.com/metrics-that-matter-two-new-approaches-from-huawei-connect-europe/.

[2] V. Aggarwal, E. Halepovic, J. Pang, S. Venkataraman, H. Yan, Prometheus: Toward quality-of-experience estimation for mobile apps from passive network measurements, in: Proceedings of the 15th Workshop on Mobile Computing Systems and Applications, in: HotMobile '14, ACM, New York, NY, USA, 2014, pp. 18:1–18:6, http://dx.doi.org/10.1145/2565585.2565600.

[3] J. Hosek, P. Vajsar, L. Nagy, M. Ries, O. Galinina, S. Andreev, Y. Koucheryavy, Z. Sulc, P. Hais, R. Penizek, Predicting user qoe satisfaction in current mobile networks, in: Communications (ICC), 2014 IEEE International Conference on, 2014, pp. 1088–1093.

[4] P. Casas, R. Schatz, F. Wamser, M. Seufert, R. Irmer, Exploring qoe in cellular networks: How much bandwidth do you need for popular smartphone apps?, in: Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges, in: AllThingsCellular '15, 2015, pp. 13–18.

[5] P. Casas, A. D'Alconzo, F. Wamser, M. Seufert, B. Gardlo, A. Schwind, P. Tran-Gia, R. Schatz, Predicting qoe in cellular networks using machine learning and in-smartphone measurements, in: 2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX), 2017, pp. 1–6, http://dx.doi.org/10.1109/QoMEX.2017.7965687.

[6] B. Finley, E. Boz, K. Kilkki, J. Manner, A. Oulasvirta, H. Hämmäinen, Does network quality matter? a field study of mobile user satisfaction, Pervasive Mob. Comput. 39 (2017) 80–99.

[7] R. Larson, M. Csikszentmihalyi, The experience sampling method, New Dir. Methodol. Soc. Behav. Sci. 15 (1983) 41–56.

[8] M. Csikszentmihalyi, R. Larson, Validity and reliability of the experience-sampling method, in: Flow and the Foundations of Positive Psychology, Springer, 2014, pp. 35–54.

[9] P. Branco, L. Torgo, R.P. Ribeiro, A survey of predictive modeling on imbalanced domains, ACM Comput. Surv. 49 (2) (2016) 31:1–31:50, http://dx.doi.org/10.1145/2907070.

[10] D. Wu, R.K.C. Chang, W. Li, E.K.T. Cheng, D. Gao, Mopeye: Opportunistic monitoring of per-app mobile network performance, in: 2017 USENIX Annual Technical Conference (USENIX ATC 17), USENIX Association, Santa Clara, CA, 2017, pp. 445–457.

[11] A. Razaghpanah, N. Vallina-Rodriguez, S. Sundaresan, C. Kreibich, P. Gill, M. Allman, V. Paxson, Haystack: In situ mobile traffic analysis in user space, CoRR, abs/1510.01419, arXiv:1510.01419.

[12] A. Shuba, A. Le, E. Alimpertis, M. Gjoka, A. Markopoulou, AntMonitor: System and Applications, CoRR, abs/1611.04268, arXiv:1611.04268.

[13] A. Ahmad, L. Atzori, M.G. Martini, Qualia: A multilayer solution for qoe passive monitoring at the user terminal, in: 2017 IEEE International Conference on Communications (ICC), 2017, pp. 1–6, http://dx.doi.org/10.1109/ICC.2017.7997262.

[14] N. van Berkel, D. Ferreira, V. Kostakos, The experience sampling method on mobile devices, ACM Comput. Surv. 50 (6) (2017) 93:1–93:40, http://dx.doi.org/10.1145/3123988.

[15] M. Schöffler, Overall Listening Experience - A new Approach to Subjective Evaluation of Audio (Ph.D. thesis), Friedrich-Alexander-Universität Erlangen-Nürnberg, 2017.

[16] European Commission and European Parliament, Brussels, Eurobarometer 87.1 (2017), tns opinion, Brussels [producer] (2017).

[17] T. Hossfeld, C. Keimel, M. Hirth, B. Gardlo, J. Habigt, K. Diepold, P. Tran-Gia, Best practices for qoe crowdtesting: Qoe assessment with crowdsourcing, IEEE Trans. Multimed. 16 (2) (2014) 541–558.

[18] V. Garcia, R.A. Mollineda, J.S. Sanchez, Theoretical analysis of a performance measure for imbalanced data, in: 2010 20th International Conference on Pattern Recognition, IEEE, 2010, pp. 617–620.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[20] G. Lemaître, F. Nogueira, C.K. Aridas, Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning, J. Mach. Learn. Res. 18 (17) (2017) 1–5.

[21] A. Altmann, L. Toloşi, O. Sander, T. Lengauer, Permutation importance: a corrected feature importance measure, Bioinformatics 26 (10) (2010) 1340–1347.

[22] F.T. Liu, K.M. Ting, Z.-H. Zhou, Isolation forest, in: 2008 Eighth IEEE International Conference on Data Mining, IEEE, 2008, pp. 413–422.

[23] U. Reiter, K. Brunnström, K. De Moor, M.-C. Larabi, M. Pereira, A. Pinheiro, J. You, A. Zgank, Factors influencing quality of experience, in: Quality of Experience: Advanced Concepts, Applications and Methods, Springer, 2014, pp. 55–72.

[24] Z. Tan, Y. Li, Q. Li, Z. Zhang, Z. Li, S. Lu, Supporting mobile VR in LTE networks: How close are we?, Proc. ACM Meas. Anal. Comput. Syst. 2 (1) (2018) 8:1–8:31, http://dx.doi.org/10.1145/3179411.

[25] D. Podborski, E. Thomas, M.M. Hannuksela, S. Oh, T. Stockhammer, S. Pham, 360-degree video streaming with mpeg-dash, SMPTE Motion Imaging J. 127 (7) (2018) 20–27.

[26] T. Fautier, State-of-the-art virtual reality streaming: Solutions for reducing bandwidth and improving video quality, SMPTE Motion Imaging J. 127 (7) (2018) 1–10.