On User-Centric Analysis and Prediction of QoE for Video Streaming Using Empirical Measurements

Maria Plakia

Thesis submitted in partial fulfillment of the requirements for the

Masters' of Science degree in Computer Science

University of Crete
School of Sciences and Engineering
Computer Science Department
University Campus, Voutes, Heraklion, GR-70013, Greece

Thesis Supervisor: Associate Professor *Maria Papadopouli*

Heraklion, April 2016

This work has been performed at the **Foundation for Research and Technology–Hellas, Institute of Computer Science (FORTH–ICS), N. Plastira 100 Vassilika Vouton, GR-700 13 Heraklion, Crete, Greece.**

University of Crete
Computer Science Department

On User-Centric Analysis and Prediction of QoE for Video Streaming Using Empirical Measurements

Thesis submitted by
Maria Plakia
in partial fulfillment of the requirements for the
Masters' of Science degree in Computer Science

THESIS APPROVAL

Author: _____
Maria Plakia

Committee approvals: _____
Maria Papadopouli
Associate Professor, Thesis Supervisor

_____
Panagiotis Tsakalides
Professor, Committee Member

_____
Ioannis Tsamardinos
Associate Professor, Committee Member

Departmental approval: _____
Antonios Argyros
Professor, Director of Graduate Studies

Heraklion, April 2016

Abstract

Over the last years, the increasing number of mobile devices, their capabilities and the access in wireless network have created an enormous rise on wireless traffic demand and use. Wireless networks often experience "periods of severe impairments", causing severe degradation to the performance of the service running on wireless devices and to the respective user experience. However, the impact of the network performance on the quality of experience (QoE) for various services is not understood in depth. Thus, assessing the impact of different network conditions and system parameters on the user experience is important for improving the telecommunication services. In general, depending on the type of service and the context, the QoE can be affected by various techno-socio-economic-cultural-psychological parameters, e.g., by the user preferences with respect to QoE and price, willingness-to-pay, and intrinsic indicators towards a service provider (e.g., brand name, perceived value, reliability), its content (e.g., richness, diversity, searching mechanisms), and even integration with other popular services (e.g., social networking applications). In the related work, the majority of efforts aim to characterize and predict the user experience, analyzing various types of measurements often in an aggregate manner.

Our group developed the uQoE, a modular framework that includes monitoring and data collection tools (uQoE tracker) and algorithms for user-centric analysis and prediction of the QoE (MLQoE prediction algorithm) in the context of video streaming service. The uQoE tracker collects network and system measurements as well as feedback from the user. The MLQoE employs several machine learning (ML) algorithms and tunes their hyper-parameters, given as input the uQoE tracker collected measurements. It dynamically selects the ML algorithm that exhibits the best performance and its parameters automatically based on the input (e.g., network and system metrics). In this thesis, we applied the uQoE for analyzing and predicting the QoE of the video streaming service in the context of two field studies, one performed in the production environment of a large telecom operator and the other at our Institute. The analysis indicated the parameters with the dominant impact on the perceived QoE and revealed that the QoE may vary across users. This motivates the use of customized adaptation mechanisms in video streaming to address the degradation in network performance. The MLQoE results in fairly accurate predictions.

Περίληψη

Η αύξηση στον αριθμό των φορητών συσκευών, των δυνατοτήτων τους όπως η πρόσβαση σε πολλαπλές διεπαφές δικτύου, έχουν δημιουργήσει ταχεία αύξηση στην ζήτηση και κίνηση της ασύρματης πρόσβασης, τα τελευταία χρόνια. Τα ασύρματα δίκτυα όμως συχνά μπορεί να εμφανίσουν «περιόδους οξείας δυσλειτουργίας», μειώνοντας με αυτό τον τρόπο την εκλαμβανόμενη από τον χρήστη ποιότητα εμπειρίας της υπηρεσίας (QoE) που εκτελείται σε ασύρματες συσκευές. Η επίδραση των δικτυακών συνθηκών στην εκλαμβανόμενη από τον χρήστη ποιότητα υπηρεσίας δεν έχει γίνει ακόμη κατανοητή σε βάθος. Η εκτίμηση της επίδρασης των διαφορετικών δικτυακών συνθηκών και παραμέτρων των συστημάτων είναι σημαντική για την βελτίωση των τηλεπικοινωνιακών υπηρεσιών. Γενικά, ανάλογα με το είδος υπηρεσίας και το περιβάλλον η εκλαμβανόμενη από τον χρήστη εμπειρία μπορεί να επηρεαστεί από διάφορους τεχνολογικούς, κοινωνικούς, οικονομικούς και ψυχολογικούς παράγοντες. Στη βιβλιογραφία, η πλειοψηφία παρόμοιων εργασιών προσπαθούν να χαρακτηρίσουν και να προβλέψουν την εμπειρία του χρήστη, αναλύοντας διαφορετικές μετρήσεις και συχνά χρησιμοποιώντας μία συναθροιστική προσέγγιση.

Η ομάδα μας έχει αναπτύξει το uQoE, ένα αρθρωτό σύστημα που περιλαμβάνει εργαλεία παρακολούθησης και συλλογής δεδομένων (uQoE tracker) και αλγορίθμους για την ανάλυση με επίκεντρο τον χρήστη και πρόβλεψη της αντιλαμβανόμενης ποιότητας υπηρεσίας (αλγόριθμος πρόβλεψης MLQoE) στα πλαίσια της υπηρεσίας βίντεο συνεχούς ροής. Το uQoE tracker συλλέγει μετρήσεις του δικτύου και των συστημάτων, καθώς και σχόλια από τον χρήστη. Ο αλγόριθμος πρόβλεψης MLQoE εφαρμόζει πολλαπλούς αλγορίθμους μηχανικής μάθησης και προσαρμόζει τις υπερ-παραμέτρους τους, παίρνοντας ως είσοδο τις μετρήσεις που συλλέγει ο uQoE tracker. Επιλέγει δυναμικά τον αλγόριθμο μηχανικής μάθησης που παρουσιάζει την καλύτερη απόδοση και τις παραμέτρους του αυτόματα με βάση την είσοδο (π.χ., δικτυακές μετρήσεις και μετρήσεις των παραμέτρων του συστήματος). Αυτή η εργασία εφάρμοσε το MLQoE για την πρόβλεψη της αντιλαμβανόμενης ποιότητας για την υπηρεσία βίντεο συνεχούς ροής στο πλαίσιο δύο μελετών πεδίου. Η πρώτη μελέτη πραγματοποιήθηκε στο περιβάλλον παραγωγής ενός μεγάλου τηλεπικοινωνιακού παρόχου και η δεύτερη στο ITE. Η ανάλυση έδειξε τις παραμέτρους με την σημαντικότερη επίδραση στην αντίληψη της ποιότητας και αποκάλυψε ότι μπορεί να υπάρχουν διαφορές μεταξύ των χρηστών. Η παρατήρηση αυτή παρακινεί τη χρήση εξατομικευμένων μηχανισμών προσαρμογής σε βίντεο συνεχούς ροής για την αντιμετώπιση της υποβάθμισης της απόδοσης του δικτύου. Το MLQoE επιτυγχάνει αρκετά καλή απόδοση στην πρόβλεψη του QoE.

Acknowledgements

First of all I would like to thank my supervisor, Professor Maria Papadopouli for showing belief in me and giving me the opportunity to work with her, for all our constructive meetings, and for her great advice and support.

I am also really grateful to Pavlos Charonyktakis, Michalis Katsarakis, Giorgos Borbudakis, and Professor Ioannis Tsamardinos for the continuous support, ideas and valuable help and contribution to this work.

Special thanks also go to the members of my dissertation committee, Professors Panagiotis Tsakalides and Ioannis Tsamardinos for their constructive comments and questions during my MSc studies.

I would like to acknowledge Forthnet S.A. for the sponsorship and the Institute of Computer Science (FORTH-ICS) for providing financial support and all the necessary equipment during this work.

This work would not have been completed without the valuable help and patience of all the volunteers who participated in the audiovisual tests. Guys, thank you all.

I would also like to thank all my colleagues at the Telecommunications and Network Lab and the mobile computing group for their friendship and support during these years. Evripidis, Nikos, Vasilis, Haris and George thank you all for the helpful discussions, the encouragement, and the great atmosphere.

A dedicated thank you to my closest friend Tasos and my girls Konstantina, Elena, Sofia, Despoina and Katerina for being always there to listen and for always providing their advice and support.

Last, but definitely not least, I would like to thank my family for their enormous and selfless support during these years. Σίσσυ, μαμά και μπαμπά σας ευχαριστώ και σας αγαπώ πολύ.

To my family and friends

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 The landscape of wireless networks and Video

Wireless access, use, and traffic demand are on fast growth. Wireless networks are increasingly being deployed to accommodate to the users demand of constant connectivity. It is expected, according to forecasts, that monthly global mobile data traffic will be 30.6 Exabytes by 2020 [1]. The total number of smartphones (including phablets) will be nearly 50 percent of global devices and connections. In the next five years, smartphones will produce 80 percent of mobile data traffic. In addition, average Internet traffic will reach 414 terabits per second (Tbps) in 2019, while busy-hour Internet traffic will reach 1.4 Petabits per second (Pbps) [2].

Video continues to be the major application generator for mobile data traffic growth. Video reached a milestone in 2012, accounting for 51 percent of global mobile data traffic, and it will account for 75 percent of global mobile data traffic by 2020. This forecast covers a variety of applications such us video downloads, video messaging and calling and video streaming. Video streaming is content sent in compressed form over the Internet and displayed by the viewer in real time. With streaming video or streaming media, a user does not have to wait to download a video file to watch it. Instead, the media is sent in a continuous stream of data and the player starts the playback as the file arrives.

1.2 Network Performance and Quality of Experience

Networks experience severe degradation of the Quality of Service (e.g., high latency, low throughput, packet loss or retransmissions), under high-load conditions. Also, wireless networks often experience “periods of severe impairments” (PSIs), characterised by significant packet losses in either or both directions between the wireless Access Points (APs) and wireless hosts, increased TCP level retransmissions, rate reduction, throughput reduction,increased jitter, and roaming/hand-off effects. A PSI can last for several seconds to the point that it can be viewed as an outage.

The quality of experience (QoE) can be defined as "the degree of delight or annoyance of a person whose experiencing involves an application, service, or system. It results from the person's evaluation of the fulfilment of his or her expectations and needs with respect to the utility and/or enjoyment in the light of the person's context, personality and current state" [3]. This definition reflects some of the user-centric and contextual aspects of QoE. In general, depending on the type of service and the context, the QoE can be affected by various techno-socio-economic-cultural-psychological parameters, e.g., by the user preferences with respect to QoE and price, willingness-to-pay, and intrinsic indicators towards a service provider (e.g., brand name, perceived value, reliability), its content (e.g., richness, diversity, searching mechanisms), and even integration with other popular services (e.g., social networking applications).

The impact of the network performance on the QoE for various services is not understood in depth. Specifically, for video streaming service Quality of Service (QoS) indicators, such as, startup delay, buffering ratio, and average resolution, have been employed to quantify network and service performance. For various applications and different QoS indicators, thresholds of "acceptance" have been estimated (e.g., two seconds of startup delay [4]). Although, these "fixed" thresholds of different metrics in the context of a service affect differently the user experience.

1.3 Motivation

Different users perceive the service degradations in various ways, depending on the application, connectivity type and context. For these reasons, it is not enough to understand the effect that QoS parameters have in the QoE for the "average user", moreover we should capture each user's personal preferences. Also, it is important to distinguish the metrics with dominant impact on the performance of certain applications and the conditions that substantially degrade their performance as perceived by users. The metrics with dominant impact serve to learn the most out of the domain and reduce the cost of measuring a high-dimensional feature space. It may be difficult to dynamically capture these aspects and assess to which extend they affect the QoE of a service, especially in a non-intrusive manner. Thus, the design of the appropriate metrics and methodologies to monitor the infrastructure (e.g., network, system, and context), collect the appropriate data, and model the QoE can be challenging.

The assessment of QoE with explicit feedback from users can be intrusive, time-consuming, and expensive. However, to effectively adapt and improve a service, the accurate prediction of the QoE becomes important. A diagnostic tool that indicates whether users perceive the deterioration of the network performance can be very useful. When users do not perceive a performance degradation, an adaptation could be avoided. While traditional network metrics can provide some insight into the quality of the communication, the diverse set of services and heterogeneity of network operators and users make the estimation of the quality of experience challenging and still largely underexplored. Furthermore, in various production network environments, there is no automated monitoring to:

- diagnose customers' poor QoE
- provide recommendations to customers
- send meaningful feedback to the network operators, application/service providers, and terminal vendors.

This will further advance the platform, improve the end-user experience for various applications/services, serve as a large-scale real-world testing environment. Furthermore, providers can achieve the following:

- Learn more about users (e.g., access patterns, traffic, QoE) and classify them into categories.
- Learn more about its network/infrastructure/service performance.
- Assess the effectiveness of content delivery solutions (e.g., infrastructure, peering links establishment, caching placement).
- Predict user QoE in certain regions, even without complete data and investigate the potential for future services, e.g., assess the content delivery in target regions and user populations (e.g., take feedback from users) and under different encodings/qualities/smartphones.
- Infer the network performance to improve the user experience for video streaming and other services: e.g., assist roaming users by suggesting to services/data plans similar to the ones at "home" network.
- Avoid an adaptation, which could be "expensive" for the provider, when users do not perceive performance degradation.
- Deploy, test and evaluate new services. Via the participation of many users in the uQoE testing environment, the testing and evaluation phase can be enriched.

1.4 Objectives

The aforementioned challenges motivate us to extend the analysis of the impact of the network conditions and system parameters on the QoE for video streaming service. The analysis is performed in two different datasets the first one is collected in an "open" and heterogeneous environment and the second one in a more controlled environment. The MLQoE [5,6], a modular framework and set of algorithms for user-centric QoE prediction based on machine-learning, was evaluated using the empirical measurements of the two datasets. The analysis and prediction it is not employed for the "average user" as it happens for the most related works. We perform the statistical analysis and predictions in a *user-centric* manner, taking into consideration each users' preferences and sensitivities.

This thesis builds on the earlier work of the team [5–8].

1.5 Methodology

In the context of a video streaming service provided by a large telecom operator in Greece, in its production environment, we performed the first field study. Volunteers employed the uQoE tracker and evaluated the perceived QoE of the video streaming service. The network and systems measurements (objective measurements, such as startup delay, rebufferring events, packet losses) as well as feedback from users (subjective measurements) of the dataset, are analyzed in order to understand the impact of various parameters on the QoE. This field study took place in a dynamic "open" relatively unrestricted and heterogeneous environment, which imposed several challenges in the analysis. To validate the outcome of the analysis and further extend it, we performed a second more controlled field study at our Institute. This work focused on the user-centric aspects in QoE and the sensitivity of users to different type of impairments.

We specify the system requirements and the desired properties of the MLQoE. The prediction of QoE is considered as a supervised regression analysis task, where the predictors are network and system metrics, such as startup delay, rebufferring events, packet losses, and changes of resolution, and the predicted outcome is the QoE score. The MLQoE consists of several components (algorithms), including the normalization, feature selection, training multiple regressors, the selection of the best ML model and the estimation of its performance. The MLQoE prediction algorithm is modular, since the set of ML algorithms employed by the MLQoE can be easily extended. Currently, the MLQoE prediction algorithm employs Support Vector Regression, Artificial Neural Networks, Decision Trees, and Gaussian Naive Bayes classifiers. The MLQoE addresses the problems of performance overestimation and underfitting. Moreover, it performs dimensionality reduction. The MLQoE takes as input statistics (e.g., mean, median, standard deviation) of the network and system metrics that were recorded during the video session that a user has viewed, along with the corresponding QoE scores. We compared the performance of the prediction algorithm with Weber-Fechner Law (WFL) [9].

1.6 Contribution

The contribution of this work is twofold. First, the joint development of the MLQoE a modular framework for prediction of the QoE. The MLQoE employs several machine learning algorithms (namely, Artificial Neural Networks, Support Vector Regression machines, Decision Trees, and Gaussian Naive Bayes classifiers) and tunes their hyper-parameters. It dynamically selects the ML algorithm that exhibits the best performance and its parameters automatically based on the input (e.g., network and systems metrics), using the Nested Cross Validation (nested CV) protocol.

Second, in this thesis we focused on the user-centric analysis of the QoE based on empirical measurements for video streaming service collected in two field studies. The first dataset was collected in the production environment of a large telecom operator of Greece using the UQoE

tracker. A system that includes monitoring and data collection tools, was developed by our team. Specifically, the uQoE tracker collects objective (e.g. startup delay, buffering ratio, packet loss) and subjective (feedback from the user) measurements. The second field study took place at our Institute and the collected dataset includes system parameters as objective measurements and feedback from the user as subjective measurements. The analysis indicated the parameters with the dominant impact on the perceived QoE and revealed that the QoE may vary across users.

1.7 Innovation

To the best of our knowledge, it is the first framework that analyzes the QoE in a personalized manner and tries to understand and "capture" the sensitivity of users to different type of impairments. The MLQoE prediction algorithm estimates the QoE of a service in a such user-centric modular manner. Its main advantage is its ability to incorporate various algorithms and automatically select the best one (and its parameters) based on the training datasets. It also learns the most out of each dataset.

1.8 Organization of this thesis

The remainder of this thesis is organized as follows: Chapter 2 overviews the related work. Chapter 3 presents the uQoE architecture and describes the two field studies. Chapter 4 presents the uQoE prediction algorithm. Chapter 5 focuses on the performance analysis and specifically, it describes the experimental setup, and it discusses the comparative performance analysis. Finally, Chapter 6 summarizes our conclusions and future work plan.

1.9 Related Publications

The methodology of the proposed algorithm, and its comparative analysis and validation study are contributions of the following publications:

1. M.Plakia, M. Katsarakis, P. Charonyktakis, I. Markopoulos, and M. Papadopouli, "On user-centric analysis and prediction of QoE for video streaming using empirical measurements", International Conference on Quality of Multimedia Experience (QoMEX 2016), Submitted

2. M. Katsarakis, M. Plakia, P. Charonyktakis, and M. Papadopouli, "On user-centric QoE prediction for VoIP and video based on machine-learning", in NSF/FCC Workshop on Tracking Quality of Experience in the Internet, Princeton, October 21-22, 2015

3. P. Charonyktakis, M. Plakia, I. Tsamardinos, and M. Papadopouli, "On user-centric modular QoE prediction for VoIP based on machine-learning algorithms", in *IEEE Transactions on Mobile Computing*, 2015

Chapter 2

Related Work

The earlier work of the team [7, 8, 10] had evaluated the impact and significance of network conditions (e.g., handover, heavy UDP, and heavy TCP traffic), different codecs (e.g., AMR, G 711) on the estimated quality of user experience, for VoIP service, using ANOVA and Tukey's HSD criterion. Firstly, the analysis was performed using a laptop [8, 10] and repeated months later using the same methodology [7] with volunteers using Android smartphones. Specifically, the team employed the E-model and PESQ and performed empirical measurements and subjective auditory tests to analyse the impact of the aforementioned network conditions on the perceived VoIP quality (QoE). It is also studied the impact of the network condition and various network parameters, such as delay, jitter, packet loss percentage, and packet loss bursts, on the perceived quality. Moreover, the accuracy of the E-Model and PESQ in regard to the scores obtained by subjects participating in an auditory test was tested.

The analysis revealed the inability of PESQ and E-model to capture the user experience under specific network conditions and the significant impact of various network parameters, such as packet loss, average packet loss burst interarrival, delay variance, and jitter variance on the perceived quality. Highly statistical significant differences between the estimations of the E-model and PESQ reported by the Student's T-test ($p < 0.01$). Furthermore, it is demonstrated that both the network condition, namely, roaming and type of background traffic, and the support of a Quality of Service (QoS) mechanism exhibit statistically significant differences in terms of their reported opinion scores.

The second analysis compared the results side by side and discovered that Androids and laptops follow, generally, the same trend when performing VoIP calls under certain conditions. Any differences between the experiments with the different devices can be attributed to different levels of wireless medium saturation and different device hardware. The ANOVA analysis indicates that the packet loss, average jitter, burst ratio, variance of jitter, average packet loss burst interarrival and average packet loss burst size have significant impact on the user perceived quality on Android devices. The team concluded that the slightly different important parameters are due to the variance of MOS scores among subjective opinions.

This motivated the need for a more thorough statistical analysis employing causal-based and Bayesian Network-based feature selection methods with larger datasets. The analysis also highlighted the challenges in predicting accurately the QoE opinion score as reported by users. We recently developed a modular algorithmic framework for user-centric QoE prediction, the MLQoE [5]. This framework employs multiple machine learning (ML) algorithms and tunes their hyper-parameters. It selects the ML algorithm that exhibits the best performance and its parameters automatically, given the input. The input involves network and systems metrics based on empirical measurements as well as subjective opinion scores collected from users. In an earlier work, we analyzed the performance of this framework on VoIP and video traces from the LIVE Mobile VQA database, which consists of a number of reference and distorted videos. The distorted videos have been created by varying the compression rate, rate-adaptation, number of frame-freeze, and packet-loss. 18 subjects assessed the quality of some of these distorted videos.

The prediction of QoE for video service can be performed by applying mathematical models based on network and systems parameters (WFL [9]), signal processing techniques (e.g., VQM [11]), or data-mining algorithms (e.g., non-linear regression models [12, 13]).

2.1 WFL

The Weber-Fechner Law (WFL) [9] indicates the relation of QoE and QoS according to the following differential equation:

$$\frac{\partial QoE}{\partial QoS} \sim -\frac{1}{QoS} \tag{2.1}$$

Integrating (2.1) leads to:

$$QoE = \log(aQoS + b) \tag{2.2}$$

This equation describes the concept of "just noticeable differences". According to this, we are able to notice sensory differences only if a physical stimulus (i.e., QoS) changes for more than a constant proportion of its magnitude. The WFL is function of stimuli (e.g., transmission bandwidth, resolution), while network metrics, such as packet loss, are not [14]. Thus, these metrics have to be transformed into stimuli with the Characteristic Stimulus Transformation Function (CSTF) [15], as follows:

$$q(r) = exp(\alpha(exp(-\beta r) - 1)) \tag{2.3}$$

where $q(r)$ is the CSTF of the packet loss r, while α and β are parameters to be estimated. Reichl *et al.* [15] applied the WFL to model the QoE as a function of the bitrate (using logarithmic regression) in the case of Speex codec. Hossfeld *et al.* [**?**] proposed a QoE model based on WFL for YouTube. Also, they compared the influence of startup delay and buffering events on the QoE. Different types of relations between the QoS (network-level traffic characteristics) and QoE (e.g., linear, logarithmic, exponential and power) applied in [**?**] and shown the relationship between

them.

2.2 Signal Processing Techniques

One of the most traditional ways to assess the quality of a video/image are the full reference algorithms. These algorithms takes as input takes as inputs two signals, the reference and the degraded one. Among the oldest FR metrics are Signal-to-Noise Ratio (SNR) and Peak Signal-to-Noise Ratio (PSNR), which are calculated between every frame of the original video signal and the video passed through a system (e.g., an encoder or a transmission channel). PSNR is the most widely used objective image quality metric, and the average PSNR over all frames can be considered a video quality metric. The PSNR is used to measure the quality of reconstruction of lossy compression codecs (e.g., for image compression or transmission). The signal in this case is the original data (image/video), and the noise is the error introduced by the transmission . When comparing compression codecs, PSNR is an approximation to human perception of reconstruction quality. The higher the estimated value of the PSNR metric, we consider that the reference and the degraded signals are more similar.

2.3 Data Mining Models and Statistical Analysis

2.3.1 Video Streaming Service

Classification and regression methods based on ML and statistical analysis have also been employed in order to estimate how objective measurements affect the user engagement or predict the QoE. The most common quality metrics are the startup delay or join time, buffering ratio and average bitrate. While as engagement metrics appears the playing time of a video or the fraction of the viewing time to the total duration of the video. For example, Krishnan and Sitaraman [4] used statistical tests (e.g., Pearson, Kendall) to evaluate the QoE based on user engagement, abandonment rate, and frequency of visits. Hands and Wilkins [16] examine quality and acceptability for video streaming under different network conditions and investigate the effects of loss and burst size. They show that burst size has considerable influence on QoE and acceptability. In [17, 18] they build applications that collect QoS (such as player state/events, statistics of buffering events, and video quality level) in the context of video streaming and web browsing services, parameters that impact the perceived QoE.

Other studies use ML algorithms with the simple *hold-out* estimation [19, 20] or with *cross-validation* [21]. Specifically, Menkovski *et al.* [19,20] uses SVMs and DTs, with labels acceptable or not, to estimate video QoE with 3 device types (Laptop, PDA, Mobiles). To address the issue of the costly training, they developed an approach reducing the training data, while keeping high accuracy. In [21], the authors present Hoeffding Option Trees models to predict QoE (acceptable or not) from continuous real-time customer feedback. Shafiq *et al.* [22], try to characterize

the impact of cellular network performane on mobile video user engagement. For the proposed prediction model they use Decision Trees with bootstrap aggregation, for both classification and regression. Balachandran in [23] builds a predictive model of Internet video QoE, they use decision trees for the prediction of the QoE and in order to produce visual representation of if-then rules for the data. The hold-out estimation separates the dataset into two sets, called the training set and the testing set. Thus, in the hold-out protocol, only a portion of the data is used for training and the rest for testing. In cross-validation instead, the dataset is divided into k subsets, and the hold-out method is repeated k times. Each time, one of the k subsets is used as the test set and the other k-1 subsets are put together to form a training set. Then, the average error across all k trials is computed. Thus, in cross-validation, all samples serve once and only once for testing, and so the estimation of performance has smaller variance than in the case of the hold-out. This is particularly important for datasets with a small number of samples, such as typical subjective studies. The final model returned by cross-validation is trained on all samples (the complete dataset). The cross-validation provides a conservative estimate of the performance of the final model [24]. Both the hold-out and the cross-validation are appropriate when using a single learning algorithm with a single set of hyper-parameter values. However, for a given dataset, it is advantageous to employ several algorithms and tune their hyper-parameters (try different combination of values). In this case, due to multiple induction, the estimated performance of the best-performing classifier is optimistic (upward biased) [25]. The uQoE prediction algorithm tries numerous ML algorithms and hyper-parameter combinations to select the best performing one; it uses the nested cross-validation instead of the simple cross-validation to shield against optimistic estimations of performance. The nested cross-validation is explained in detail in Section 4.6. The Chapter 4 presents a number of ML algorithms for the QoE prediction.

Simple regression models have been also used in order to characterize the user satisfaction [26–28]. For example, Dobrian *et al.* [26] uses Kendall correlation, information gain, and linear regression to estimate the quantitative impact of quality metrics on user engagement. OneClick [27] applies Poisson regression to capture user satisfaction. Q-score [28] learns with Ridge regression a set of performance indicators (network performance indicators, user behaviors and activities, user feedback in the form of trouble tickets) most relevant to user-perceived quality and proactively infers service quality in a single score. Hossfeld *et al.* [29] presents an exponential QoE model for YouTube that takes into account stalling events as predictors. A subjective QoE assessment methodology for multimedia applications based on crowdsourcing is also proposed.

The role of the context on QoE for various streaming services has been highlighted in several studies (e.g., [30]). The evaluation of acceptance, satisfaction, entertainment, and information recognition in different contexts (e.g., train station, bus, cafe) using Anova, Pearson correlation, Spearman, and Chi-square is the focus of [31]. The context and the repeatability of the experiments was analysed in [32]. In the context of video streaming and telepresence, Wu *et al.* [33] characterized the QoS based on interactivity, vividness and consistency and the QoE using as

metrics the concentration, enjoyment, telepresence, perceived usefullness, and perceived easiness of use. It then mapped QoS to QoE by applying Pearson's correlation. Xue and Chen [34] evaluate the influence of contextual factors, such as display size and viewing distance, ambient luminance and user movement on subjective perceived quality. All the aforementioned models estimate the QoE for an average user.

In general, the ground-truth for the QoE has been formed based on either the explicit opinion scores reported by the users (e.g., in the context of listening tests/controlled studies or at the end of their service via a GUI, as in the case of skype) or based on measurements collected using physiological metrics [35,36]. Mandryk *et al.* [35] test the efficacy of physiological measures as evaluators of user experience in entertainment technologies using Multivariate Analysis of Variance (MANOVA). The use of physiological methods to assess the user cost on different levels of media quality using MANOVA is the focus of [36].

The closest paper in our work [37] builds the aggregated model training two different algorithms (DTs and ANN) using 10-fold cross validation for DTs and hold-out estimation for ANN. So the estimated performance is overestimated. They choose the best performed model and train it in a user-centric manner. The models have been trained using all the parameters unlike our work that performs feature selection for dimensionality reduction.

2.3.2 Audio Service

Classification and regression methods based on ML and statistical analysis have also been employed for the prediction of QoE. These methods assume as "ground truth" the MOS that the E-model or PESQ reports. Thus, their estimations propagate the error of the E-model and PESQ [12,13,38–41]. For example, Sun and Ifeachor [12] proposed a method for predicting voice quality for buffer optimization using non-linear regression and a regression methodology [13] for predicting conversational voice quality with the aid of PESQ and E-model. They also investigated the impact of packet loss and different talkers on the QoE and proposed an ANN model for predicting QoE on VoIP [40]. Wu *et al.* [41] also investigated the playout buffer dimensioning in Skype, Google Talk, and MSN messenger using PESQ. Cherif *et al.* [38] used Random Neural Networks (RNNs) to learn the nonlinear relation between network parameters and the perceived user QoE, avoiding subjective tests with PESQ. In [39], the authors presented a method for QoE estimation and prediction based on hidden Markov models (HMM) and multi-homed mobility management protocol using passive probing mechanisms and E-model.

Several studies are also based on the MOS that the E-model or PESQ reports to assess the VoIP quality (e.g., [42–44]). This can lead to wrong conclusions, given the inability of them to capture the user experience. For example, Markopoulou *et al.* [42] focuses on ISP network problems and shows that ISP networks suffer from PSIs affecting the QoE for real-time applications. Birke *et al.* [43] performs VoIP traffic characterization from backbone measurements and highlights that the packet loss affects the QoE. The performance of VoIP over 3G networks

employing a real testbed was the focus of [44]. They conclude that the end user VoIP experience over HSDPA is still significantly worse than with circuit switched solutions and is not acceptable.

Typically, studies that perform subjective tests estimate the performance of their models using the hold-out estimation [45, 46] or the cross-validation algorithm [47–49]. Rubino *et al.* [45] describes a methodology called Pseudo-Subjective Quality assessment (PSQA), based on RNNs, performing QoE prediction. Prometheus [46] uses LASSO regression to estimate QoE for unidirectional VoIP. Mitra *et al.* [47] cross-validates the CaQoEM, a context-aware approach, that uses Bayesian networks and utility theory to predict QoE. In [49], the authors design predictors for the user dissatisfaction and cross-validate Linear Discriminant Analysis (LDA) and SVM models to estimate it. Bhattacharya *et al.* [48] uses acoustic, lexical, discourse features in SVM and k-Nearest-Neighbour (kNN) models to predict QoE.

Chapter 3

Background

3.1 Monitoring and data collection

A major Greek telecom operator has been providing a VoD, LiveTV, TSTV, and TVoD video streaming service. In a joint project, our group developed the uQoE tracker, a monitoring system that collects network and systems measurements (objective measurements) as well as feedback from users (subjective measurements). The uQoE tracker follows the client-server architecture (Fig. 3.1). It runs on the smartphone of the user (*client*), monitors the network in the background, and parses the log messages generated by the video streaming client, when the user performs certain actions. At the end of a video viewing session (from now on called *session*), the user rates the session by providing an opinion score (via the client). The collected measurements "capture" various events, such as resolution changes, buffering events, and user actions with respect to video viewing. The uQoE server (simply *server* from now on) is running on a Linux virtual machine (VM) and collects, stores and analyzes the objective and subjective data uploaded by the clients. The client consists of the monitor, GUI, performance estimator, database (DB), and the back-end interface. The monitor is composed by three sub-modules, namely, the logcat parser, active prober, and localization. The logcat parser parses periodically the log messages of the video streaming client, recognizes various user actions and other events that may occur during the session, and keeps track of the state of the video player. When a video session start (end) is identified, the active prober is launched (terminated), respectively. During its activation, the active prober communicates with the active prober module of the server, for the initiation of network measurements through the iperf tool. The localization sub-module determines the geographical location of the device during a video session. The measurements of the monitor are transmitted to the performance estimator module for the assessment of the performance and for storage in the local database (DB). The performance estimator receives as input events from the video player execution (e.g., buffering events, termination type), network measurements received during a video session, as well as the user feedback and assesses the quality of experience. When specific quality indicator values are not met, the performance estimator may trigger a

questionnaire with a dynamically chosen set of questions to assess the perception of specific problems. The GUI of the uQoE tracker client enables the user to provide its feedback (Fig. 3.17). The user feedback consists of an opinion score (an evaluation of the perceived QoE) and possibly answers to additional questions regarding the problems that might have been encountered. The video streaming client uses only the wireless network. Similarly the communication of the uQoE system takes place via wireless network.

3.1.1 Video session features

For each video session, the following features were collected: the *service type, startup delay* and *the ratio of the startup delay over the session duration, session duration, QoE score, number of buffering events* (and statistics about the duration of them, such as total, min, max, mean and standard deviation), *the mean weighted resolution* and *the ratio of the weighted mean video resolution over the size of the display of the user device, the number of switches of the video resolution* (and statistics based on them, e.g., min, max, mean and standard deviation), *packet loss, jitter,* and *signal strength*. The user activity is characterized by the duration of the pause, seek, and off-screen events. The same statistics are also computed for the *last* 15 sec, 30 sec, and 60 sec of the session. The termination type which indicates whether the session was terminated due to poor connectivity or normally by the user is also obtained.

In the collected dataset we consider two types of parameters. The first type are the system parameters which are directly perceived from the user, such as the startup delay,the buffering event duration ratio, number of freeze events and the weighted mean video resolution. The second type are the network parameters which describe the conditions and impairments of the network, such as packet loss, jitter and RSSI.

3.1.2 Methodology

Inclusion and exclusion criteria: Due to various constraints, our volunteer group were employees at the telecom provider, excluding the members of our group.

The field study encompassed the following phases:

(i) Software development: Our group set the goals, defined the field study protocol and developed the uQoE tracker.

(ii) User study launch: The telecom provider working group installed the video service and the uQoE tracker on the volunteers' smartphones and obtained the volunteers' written consent for participating in the user study. During this process, each subject attended a short tutorial, individually. The tutorial included a demonstration of the uQoE tracker and some instructions for the participation in the study. The volunteers were instructed to use the video service and the uQoE tracker during their leisure time to watch TV broadcasts and movies and evaluate their experience. The users were encouraged to experiment with various networks (e.g., at home, cafés,

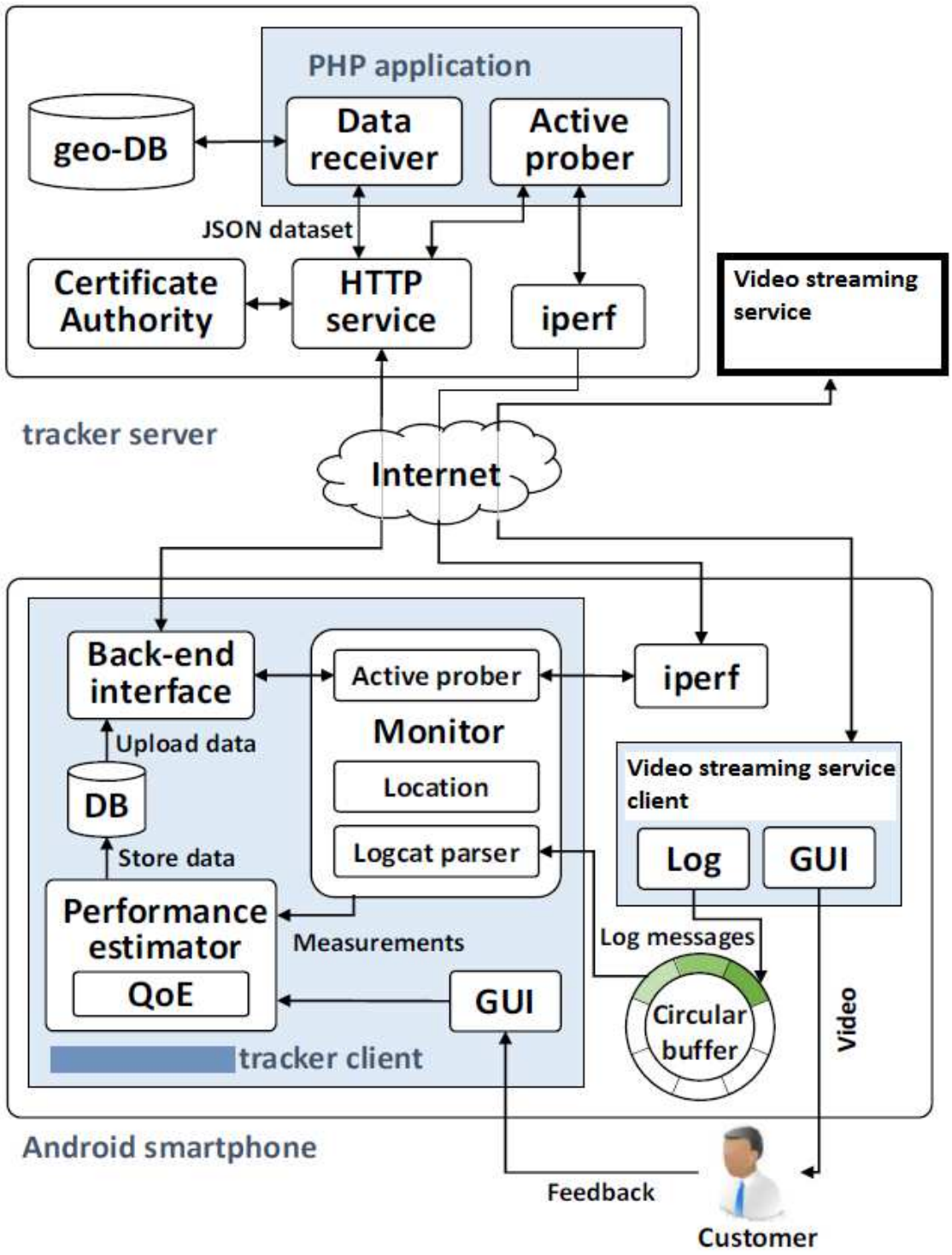


Figure 3.1: The uQoE tracker architecture.

restaurants, hotels), as well as various network conditions (e.g., congested networks, low signal strength conditions).

(iii) User study runtime: For the following 56 days the volunteers used the video streaming service and the uQoE tracker.

Figure 3.2: An example of uQoE servers' GUI.

(iv) Data analysis: After the 56-day data-collection period, our group extracted the collected data from the uQoE tracker for analysis.

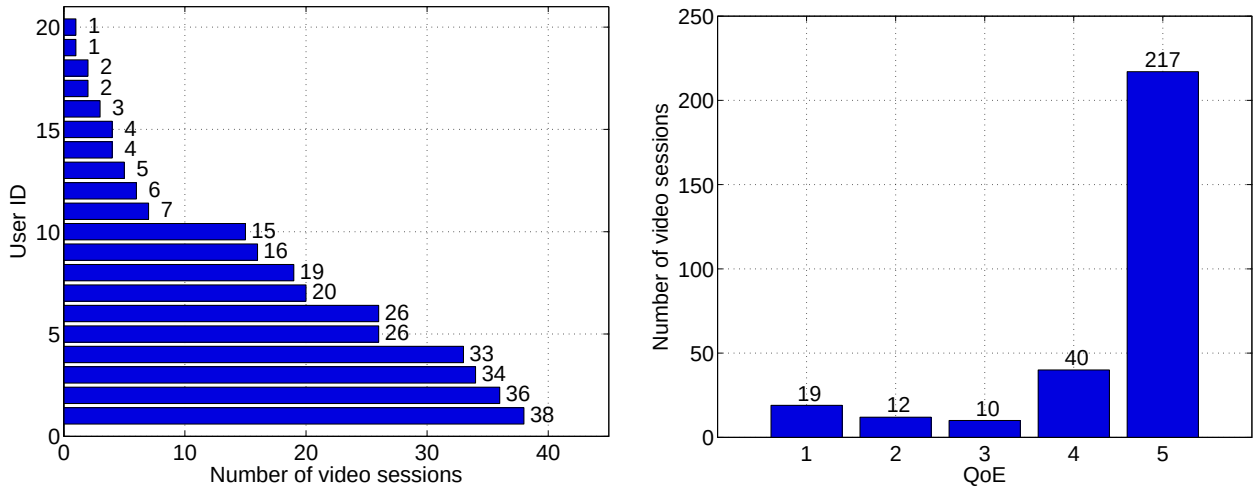
The user study was anonymous. Each participant was identified only by a unique id generated by the uQoE tracker client. No contact took place between the our group (who developed theuQoe tracker and analyzed the collected data) and the volunteers. The telecom providers' working group acted as a middleman, when our team had to direct a notification to the volunteer group.

3.1.3 Testbed of the user study

The testbed includes:

- the uQoE tracker server that runs on a virtual machine (VM) with 2 CPU cores (2.6 GHz), 7.8 GB RAM memory, and 72 GB hard disk size, and
- 20 Android smartphones that run the uQoE tracker client as well as the video streaming service client. The Android smartphones are owned by the volunteers and vary regarding their manufacturer, model, display size, and Android version.

The video streaming service client allows video streaming only over WiFi networks; streaming over cellular networks is not supported, to protect customers from unexpectedly high charges by their cellular network provider. For the same reason, the uQoE tracker client uploads the collected data only over WiFi.



(a) (b)

Figure 3.3: (a) Number of video sessions per user and (b) histogram of QoE distribution.

Table 3.1: Features of the first field study

Metrics	First field study dataset
Startup delay (sec)	0 - 130.86
Number of buffering events	0 - 8
Buffering ratio	0 - 0.99
Average Resolution	0 - 720
Packet loss (%)	0 - 54.72
Jitter (msec)	0 - 161.69
RSSI (dBm)	(-89.60) - (-34.66)

3.2 Analysis of first field study

The first field study took place in the context of the video streaming service provided by a large Greek telecom operator. During this field study, 20 volunteers, customers of the service, participated by viewing videos, uploading at least one *labelled* video session. We consider as labelled session a session that has been rated with a QoE score by the user. The devices of the participants vary in terms of their manufacturer, model, display size, and Android version.

3.2.1 System parameters and network conditions

Some first observations are that 8 % of the total sessions (particular 22) have startup delay higher than 10 sec (Fig. 3.4 (a)). 20 sessions have buffering event duration ratio higher than 0.1 (Fig. 3.4 (b)). The most common weighted mean video resolution is the 620 px, while 8 video sessions never started playing (Fig. 3.4 (c)). 5 video sessions have packet loss higher than 20% (Fig. 3.4 (d)).

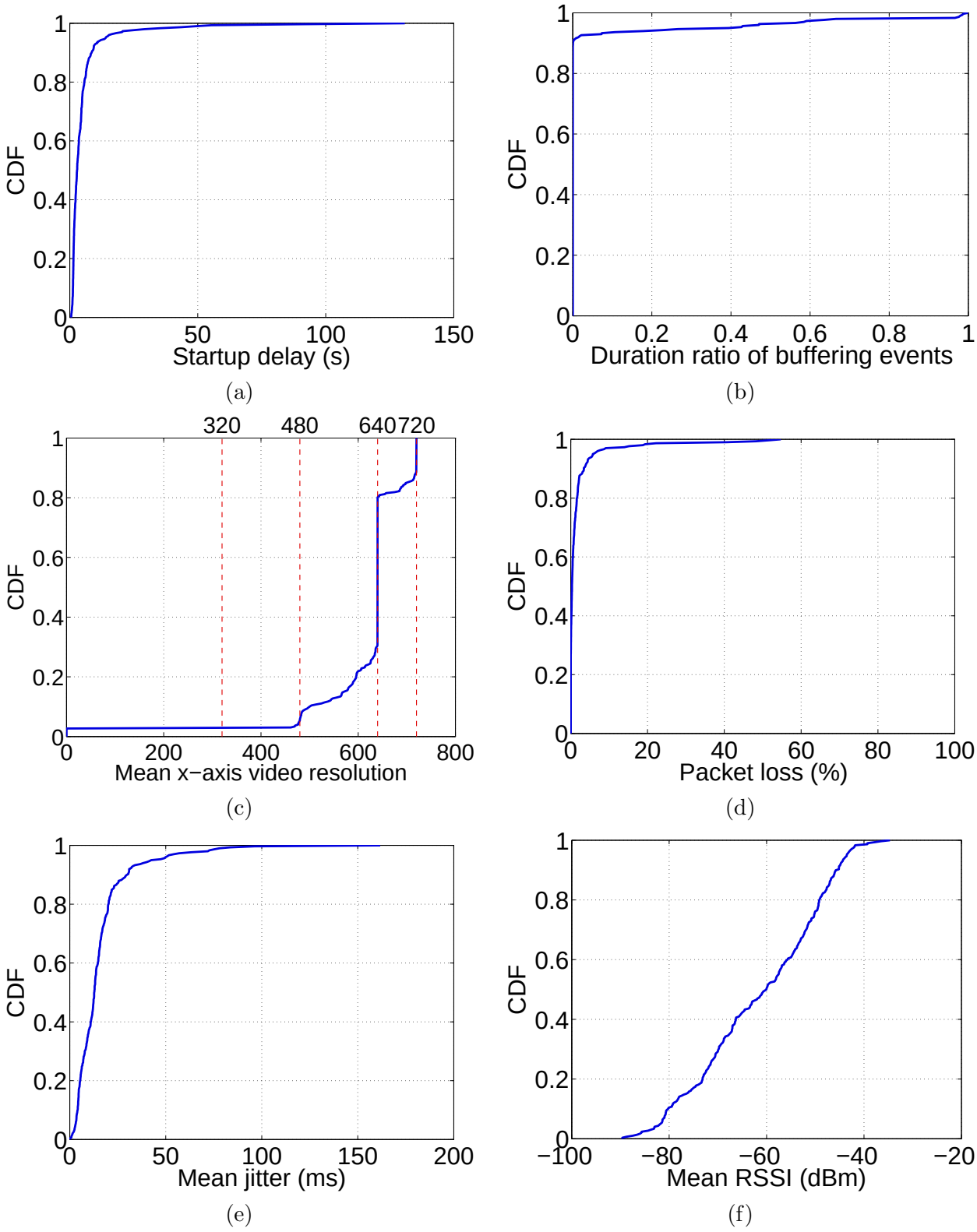


Figure 3.4: System parameters and network conditions for the first field study.

3.2.2 Stationary vs. wireless handover sessions

We investigated the dataset for stationary and wireless handover sessions and the trends which appears in them. In the stationary sessions, the smartphone is associated with only one AP during

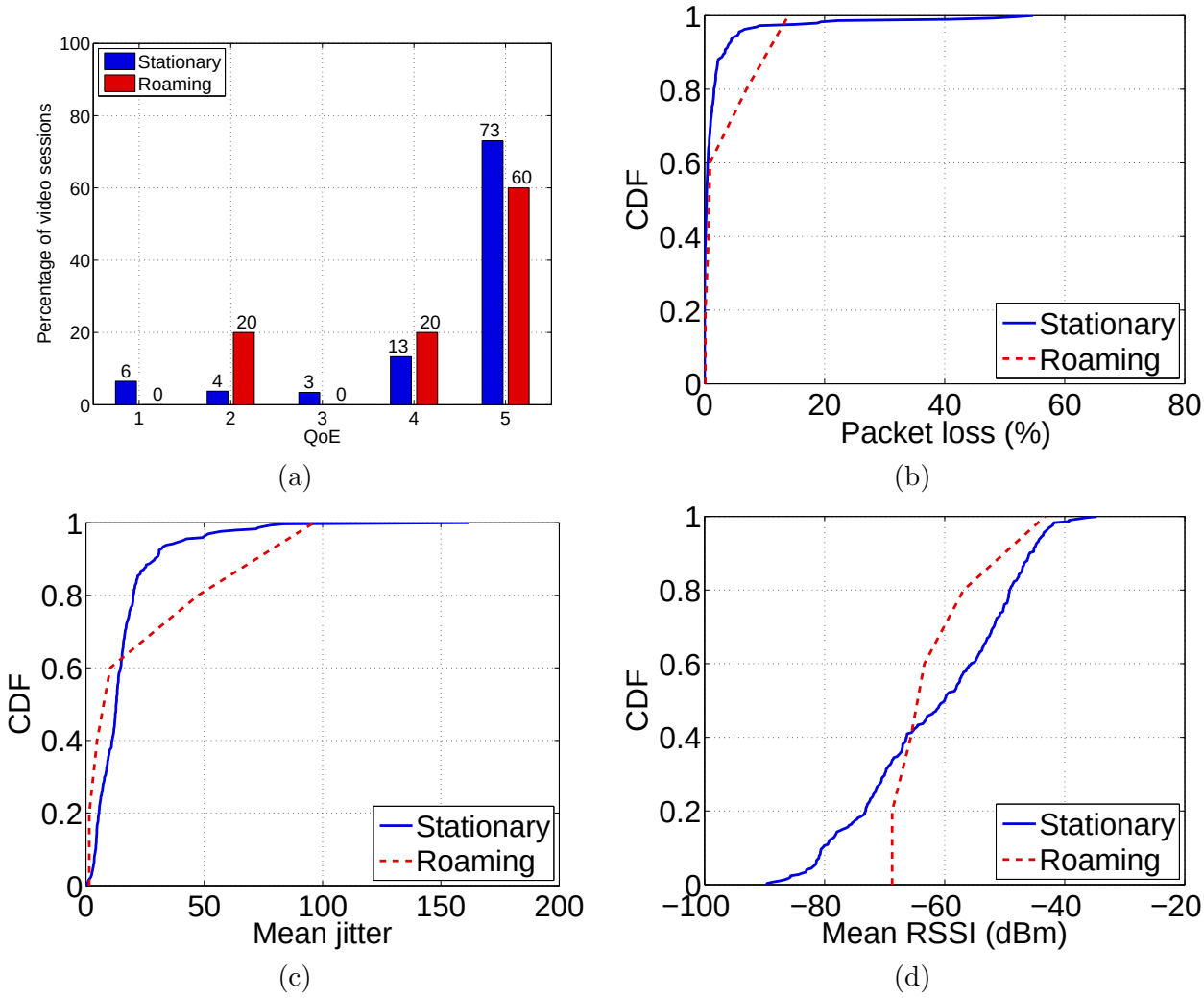
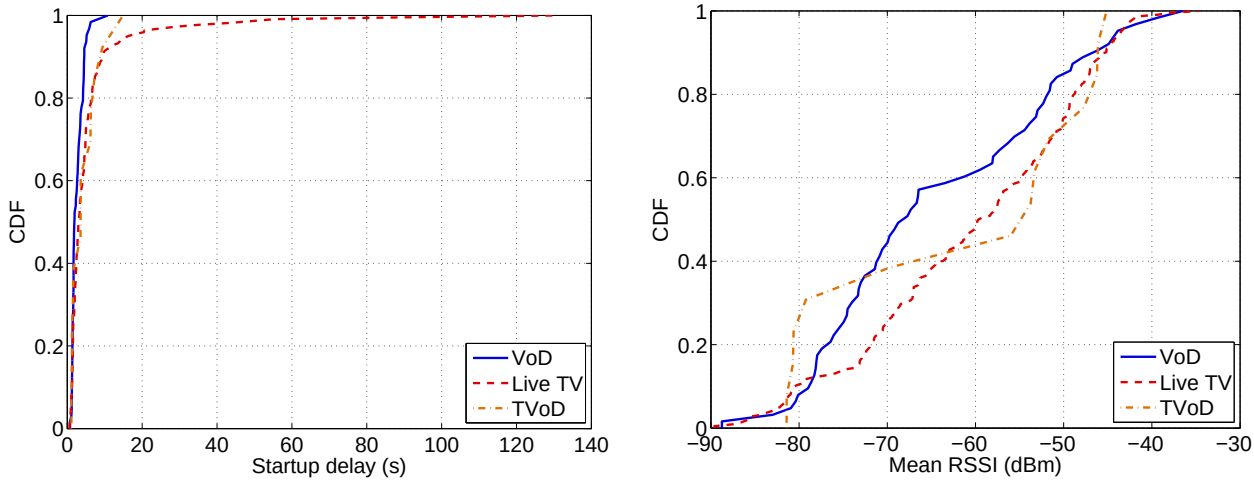


Figure 3.5: (a) QoE histogram and network conditions for roaming vs, stationary sessions (b)packet loss, (c) mean jitter and (d) mean RSSI.

the entire session. During the wireless handover sessions the smartphone performed handovers between APs. In the collected dataset, the 293 of the total 298 sessions were stationary. The roaming sessions have higher packet loss, jitter and stronger signal than stationary (Fig. 3.5).

3.2.3 Service Type

In this field study the three of them were appeared, particularly VoD, LiveTV, and TVoD with 63, 222 and 13 video sessions for each service type, respectively. The LiveTV sessions have lower QoE scores than VoD ones (3.7). LiveTV sessions in few cases have extremely high startup delay. Specifically, 20 out of 22 sessions with startup delay have startup delay higher than 10 sec (3.6 (a)). The RSSI is higher for the LiveTV sessions (3.6 (b)).



(a) (b)

Figure 3.6: (a) Startup delay and (b) mean RSSI for different service types.

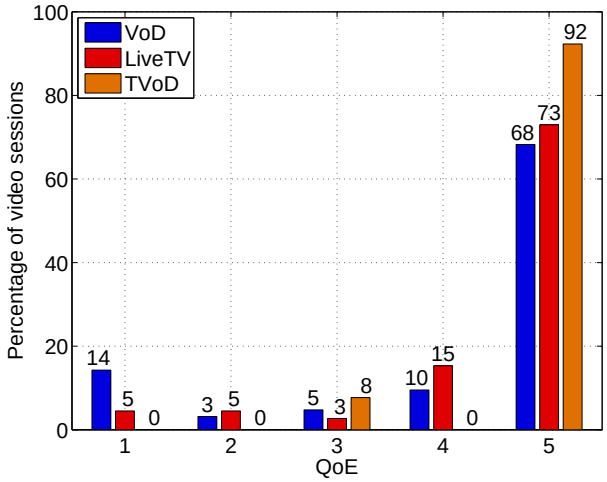
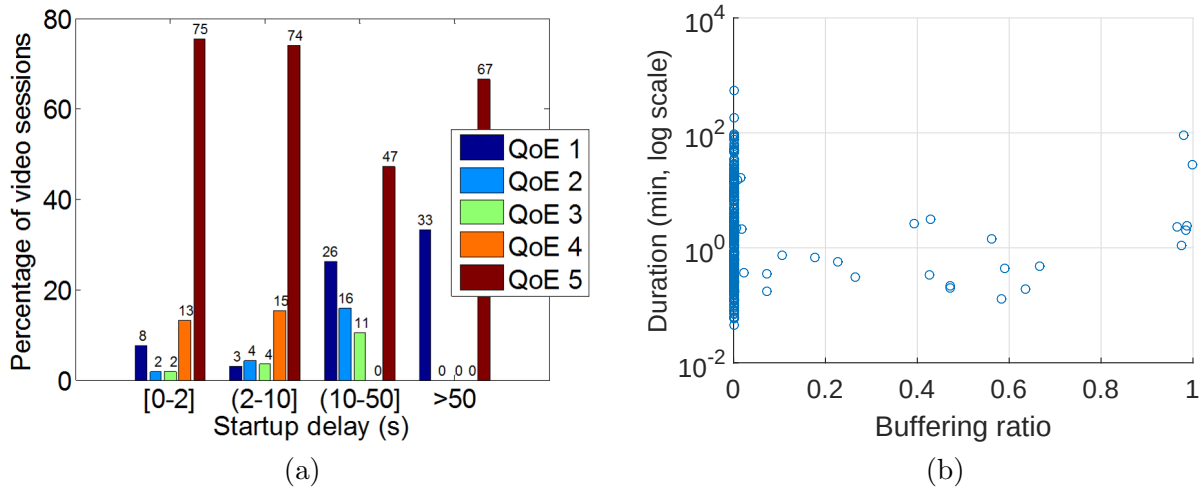


Figure 3.7: QoE distribution for sessions for different type of service.

3.2.4 Startup delay and buffering ratio

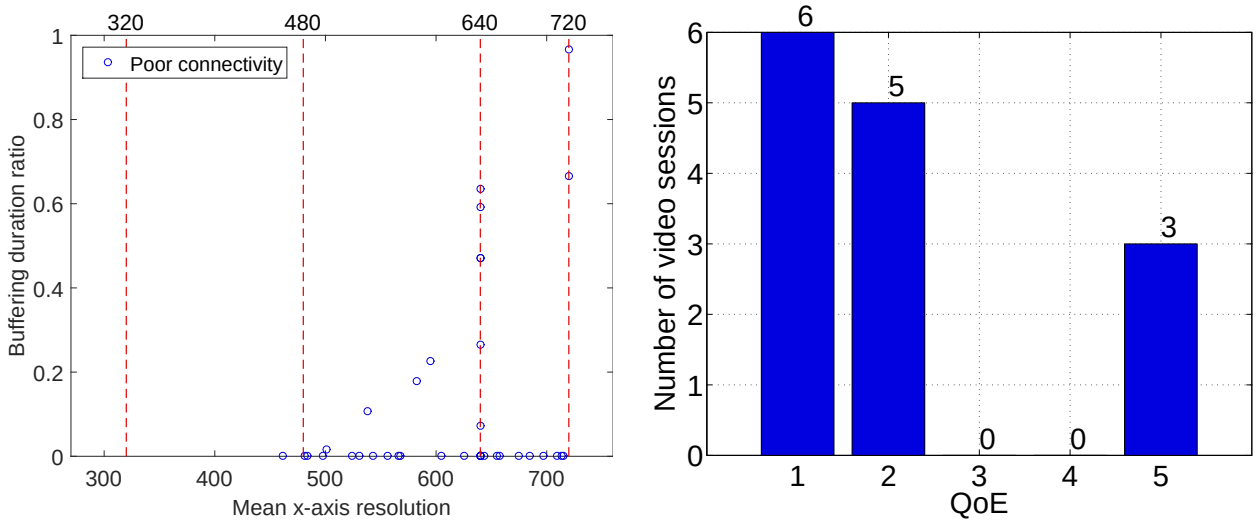
We first analyzed the impact of the startup delay on QoE. Users perceive the degradation (reflected by the low QoE scores) when the startup delay is 10 sec or more (Fig. 3.8 (a)). Prior related research (e.g., by Krishnan *et al.* [4]) reported that a startup delay beyond 2 sec causes viewers to abandon the video. The dataset in [4] contains measurements obtained from wired, wireless, and cellular connections. We speculate that the smartphone wireless network users of our study are perhaps more tolerant in the startup delay than users with fixed devices using a larger bandwidth connection.

The higher the buffering ratio, the smaller the duration of the session. The increased buffering ratio decreases the viewing time (Fig. 3.8 (b)). This trend has been also observed in the related work (e.g., [4], [26]).



(a) (b)

Figure 3.8: (a) Histogram of the QoE distribution for different startup delays and (b) the duration of the session as a function of the buffering ratio.



(a) (b)

Figure 3.9: Sessions terminated by poor connectivity (a) buffering ratio and (b) distribution of QoE for these sessions.

3.2.5 Resolution and buffering ratio

Interestingly, there were several sessions of high resolution that terminated with poor connectivity and exhibit high buffering duration ratio and low QoE scores (Fig. 3.9 (a)). Potentially, by lowering the resolution, the buffering ratio could be, reduced resulting in an improved QoE.

3.2.6 Sessions with severe degradations

As expected, sessions that experience worse network conditions (e.g., in terms of jitter, packet loss and RSSI), high startup delay, and buffering ratio are rated with lower QoE. The larger the packet loss, the lower the QoE score (Fig. 3.10 (a)). These are also more likely to terminate with a poor connectivity status (Fig. 3.10).

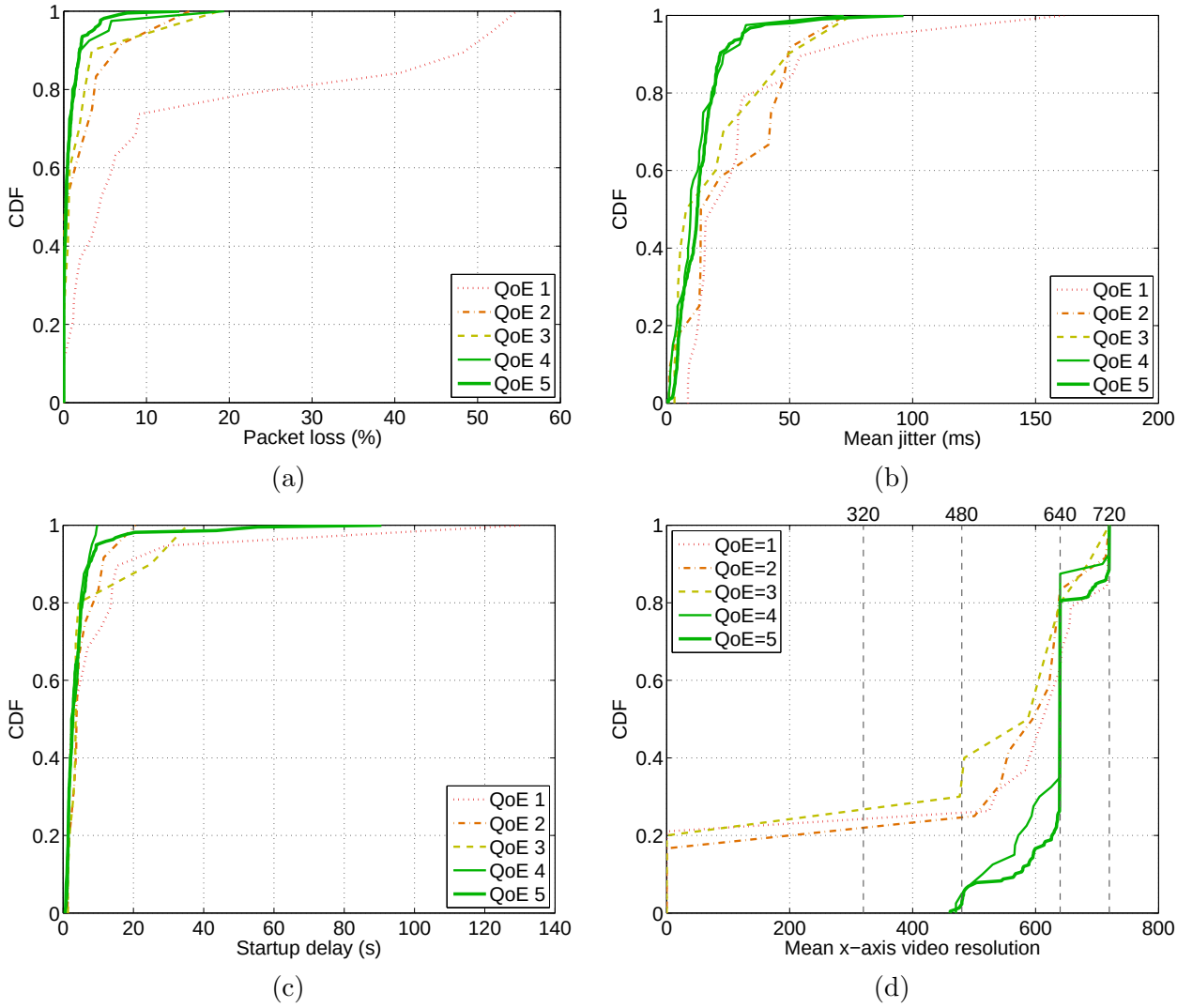


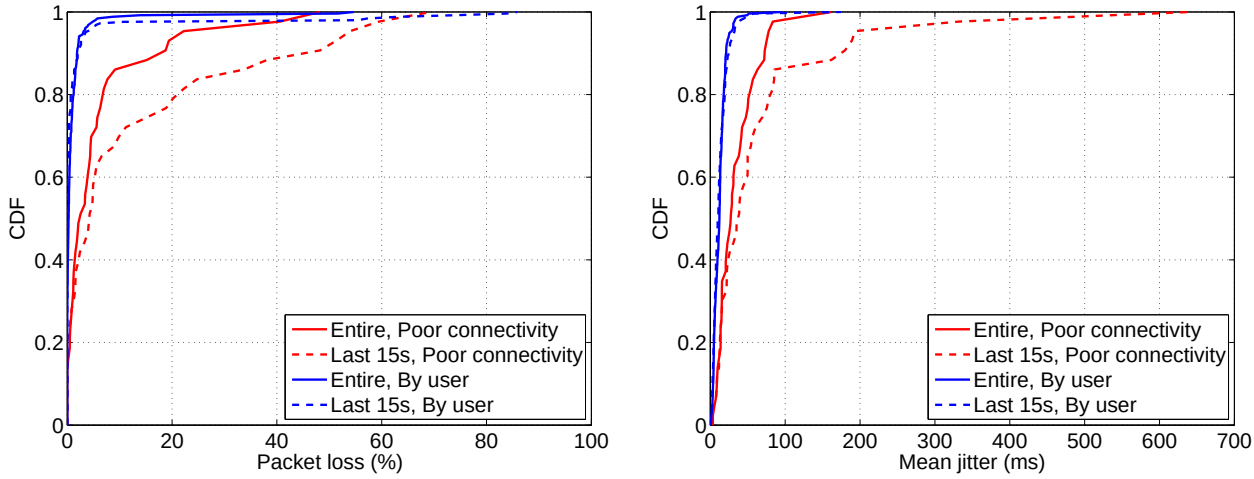
Figure 3.10: (a) Packet loss (b) mean jitter, and high (c) startup delay and (d) video resolution for sessions rated with different QoE scores.

3.2.7 Poor network performance during the last 15 sec of the session

Poor network performance during the last 15 sec of the session may result in termination due to poor connectivity. Specifically, sessions that have been terminated with poor connectivity exhibit higher packet losses, jitter, and buffering ratios during their last 15 seconds than the entire sessions terminated with poor connectivity or the sessions terminated normally by the user (Fig. 3.11).

3.2.8 Interesting characteristics

There are sessions with high buffering ratio that lasted more than 10 min and were rated with a score of 5. Moreover, there are sessions terminated with a poor connectivity status that were also rated with high QoE scores . That is, even though users experienced a degraded performance, they still rated these sessions with high QoE scores. As mentioned earlier, the first field study



(a) (b)

Figure 3.11: (a) Packet loss and (b) jitter, for sessions with different termination types, as well as considering only the last 15 sec of the sessions.

took place in a dynamic heterogeneous and relatively unrestricted environment. We speculate that depending on the context and content of these sessions, the expectations and tolerance of the users may vary.

3.3 Second field study

The limitations of the first dataset motivated us to perform a second (more controlled) field study at our Institute. For the second study, we produced a number of synthetic video sequences that correspond to a wide range of network conditions. The effect of the network conditions (such as packet loss, jitter, RSSI) on the systems parameters, such as startup delay, buffering ratio, and resolution, depends on the specific video codec and application. Moreover the systems parameters affect directly the user perceived QoE.

We generated scenarios of different types of impairment by varying these parameters and created (playback) videos "manifest" these impairments. We used four different reference videos, each corresponding to high quality (i.e., did not exhibit any type of impairment) and displaying a different scene. Each scene has a total duration of 20 sec, while each video consists of 4 chunks with a duration of 5 sec. Each playback video was parameterized based on the startup delay, number of buffering events, ratio of buffering duration, times when buffering events occur, duration of each buffering event, video resolutions for each chunk, and aggregate resolution of the video.

The startup delay and the buffering ratio have been modelled according to the Bounded Pareto distribution [50]. The parameters of the distributions were estimated based on the empirical measurements of the first field study. During a video, up to three buffering events may occur, one after each chunk. The possible resolutions of each chunk are: low (360p), medium (480p), and high (720p). The resolution may remain fixed or vary during the video.

Table 3.2: Features of the second field study

Metrics	Second field study dataset
Startup delay (sec)	0 - 19.30
Number of buffering events	0 - 3
Buffering ratio	0 - 0.90
Duration of Buffering events (sec)	0 - 20
Average Resolution	360 - 720

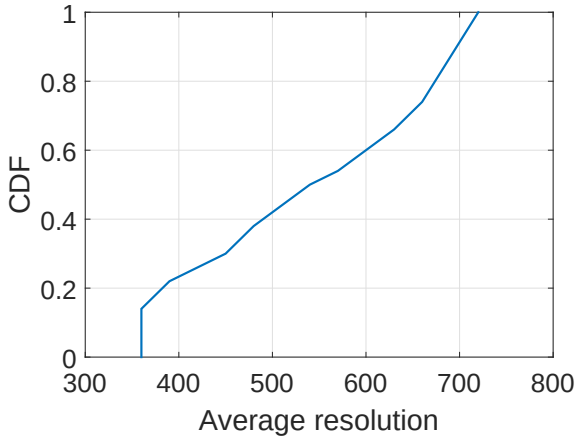
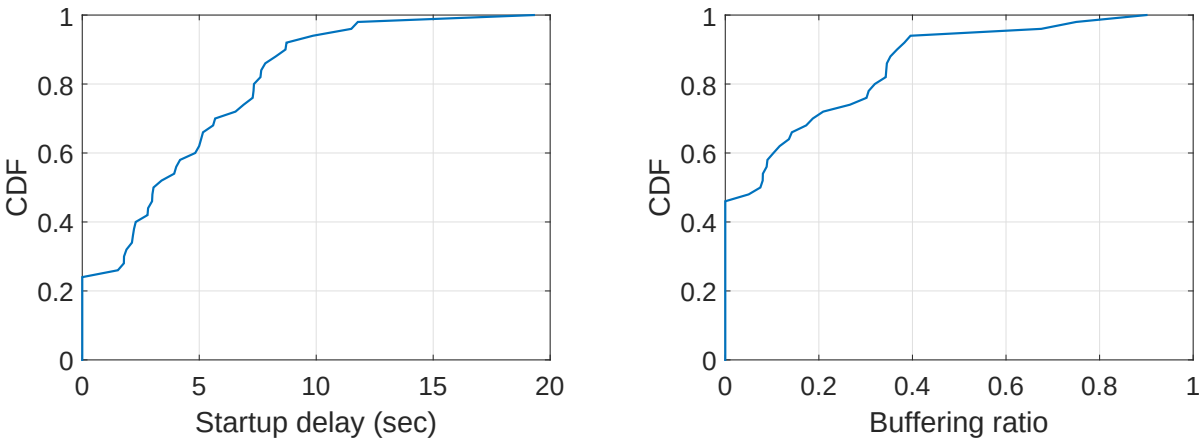


Figure 3.12: CDF for second field study of average resolution.



(a) (b)

Figure 3.13: (a) Startup delay and (b) buffering ratio, for the second field study.

We considered three different scenarios of rate adaptation algorithms, in order to model the resolution of each chunk:

- In the first scenario,the rate adaptation requests the first chunk in high resolution, and only after three continuous events request a chunk with lower resolution.
- In the second scenario, the rate adaptation algorithm, requests the first chunk in a high

resolution,if no buffering event occurs after this chunk, the next chunk is also requested in high resolution. Otherwise, when a buffering event occurs then the resolution of the next chunk is lower.

- In the third scenario, the rate adaptation algorithm requests the first chunk in a medium resolution, the next chunk is requested in higher resolution after the first chunk if no buffering event appeared, otherwise the resolution of the chunk will be even lower.

For the second study, fifty video sequences were produced. 20 participants, volunteers, mostly graduate students in our Institute, assessed the quality of these sessions using the Likert scale with scores from 1 to 5, with 1 (5) corresponding to the lowest (highest) video quality, respectively. Due to the large number of videos to be assessed, each participant viewed the videos during two viewing phases that took place in different days. The subjects viewed and assessed the videos using an Android application implemented on a Nexus 5. To obtain demographic information (e.g., age, sex, frequency of use of mobile applications, video streaming services, audiovisual tests) about the volunteers, each subject had to first answer a short questionnaire. Before viewing the videos, the subject had to read and follow the instructions that appeared in the screen. After that, the training video sequences appeared in order to familiarize the subject with the various types of audiovisual quality degradation. Then, the subject viewed each video sample and indicated his/her opinion score about its QoE via the Android application.

3.3.1 Sensitivity of the users

We also aimed to further explore the subjectivity of the assessments and the sensitivity of users to different types of impairment (e.g., large startup delay, number of rebuffering events, low resolution). Specifically, we considered three types of prominent impairments, namely, the large startup delay, number of buffering events, and low resolution, and created three homogeneous sets with respect to these impairments. Specifically:

- The set with the prominent startup delay include *only* the video sessions of high startup delay, excluding the video sessions with rebuffering events or low resolution.

- The set with the prominent buffering events include *only* the video sessions with buffering events, excluding the video sessions of high startup delay or low resolution.

- The set with the prominent low resolution include *only* the video sessions of low resolution, excluding the video sessions of high startup delay or buffering events.

We then analyzed how users rate the QoE of these videos. Indeed it appears that depending on the type of impairment, some users are more tolerant or strict than others. We define that a user assessed a session in a *lenient (strict)* manner when his/her score belongs to the 90-th (10-th) percentile of the total scores for this video provided by all 20 users of the field study, respectively.

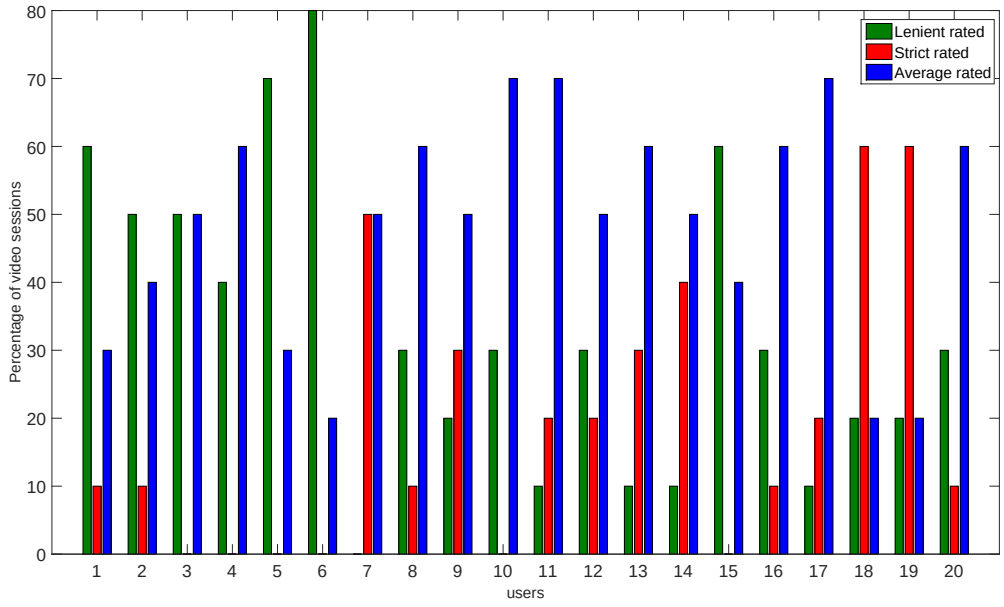


Figure 3.14: Scenario for sessions with high startup delay.

A user is labelled as lenient (strict) when the majority of his/her sessions (more than 50%) are rated in a lenient (strict) manner, respectively. Some users are persistently labelled as lenient (e.g., users 5 and 6) or strict (e.g., users 7 and 19) across all the three types of impairment. Moreover, some users (e.g. users 1 and 2) are more tolerant to some types of impairment (e.g., high startup delay and low resolution) but could not tolerate others (e.g., buffering events). To evaluate if the difference of the scores of users for the various types of impairment is statistically significant, we applied the Student's T-test, on their QoE scores. For the persistently lenient and strict users, the QoE scores among the various types of impairment are not statistically significant different, while for users that are tolerant to only some types of impairment the QoE scores are statistically significant different.

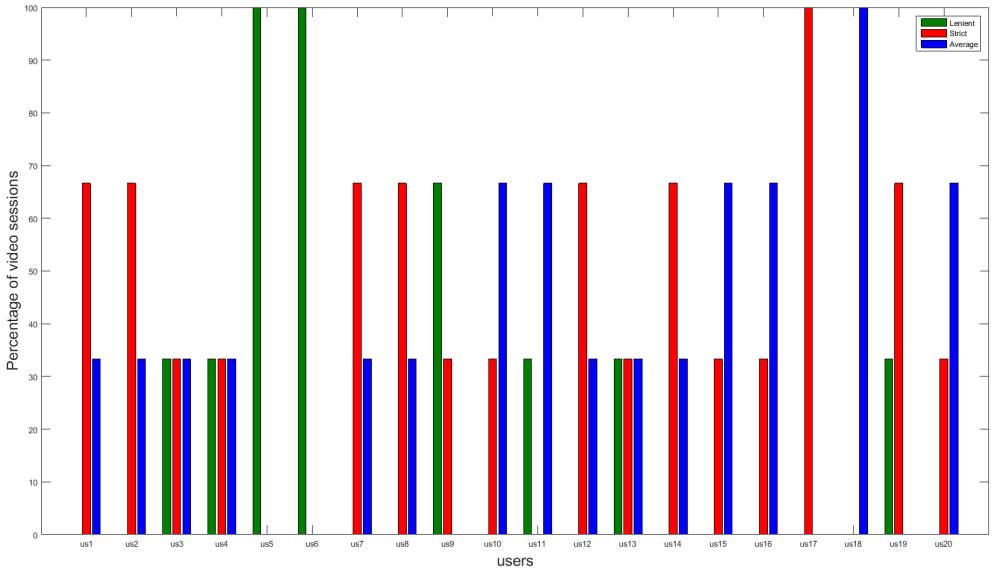


Figure 3.15: Scenario for sessions with buffering events.

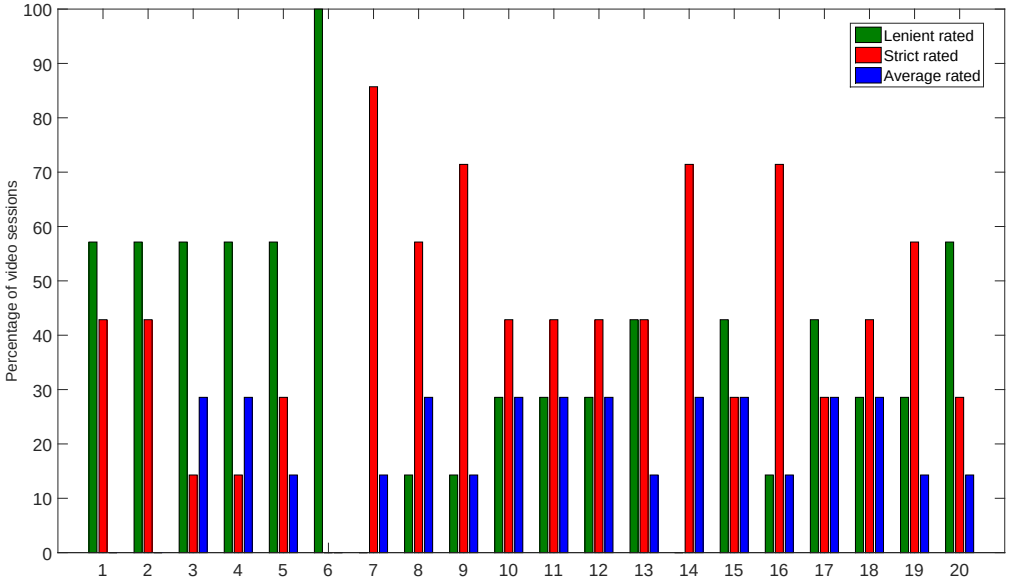


Figure 3.16: Scenario for sessions with low resolution.

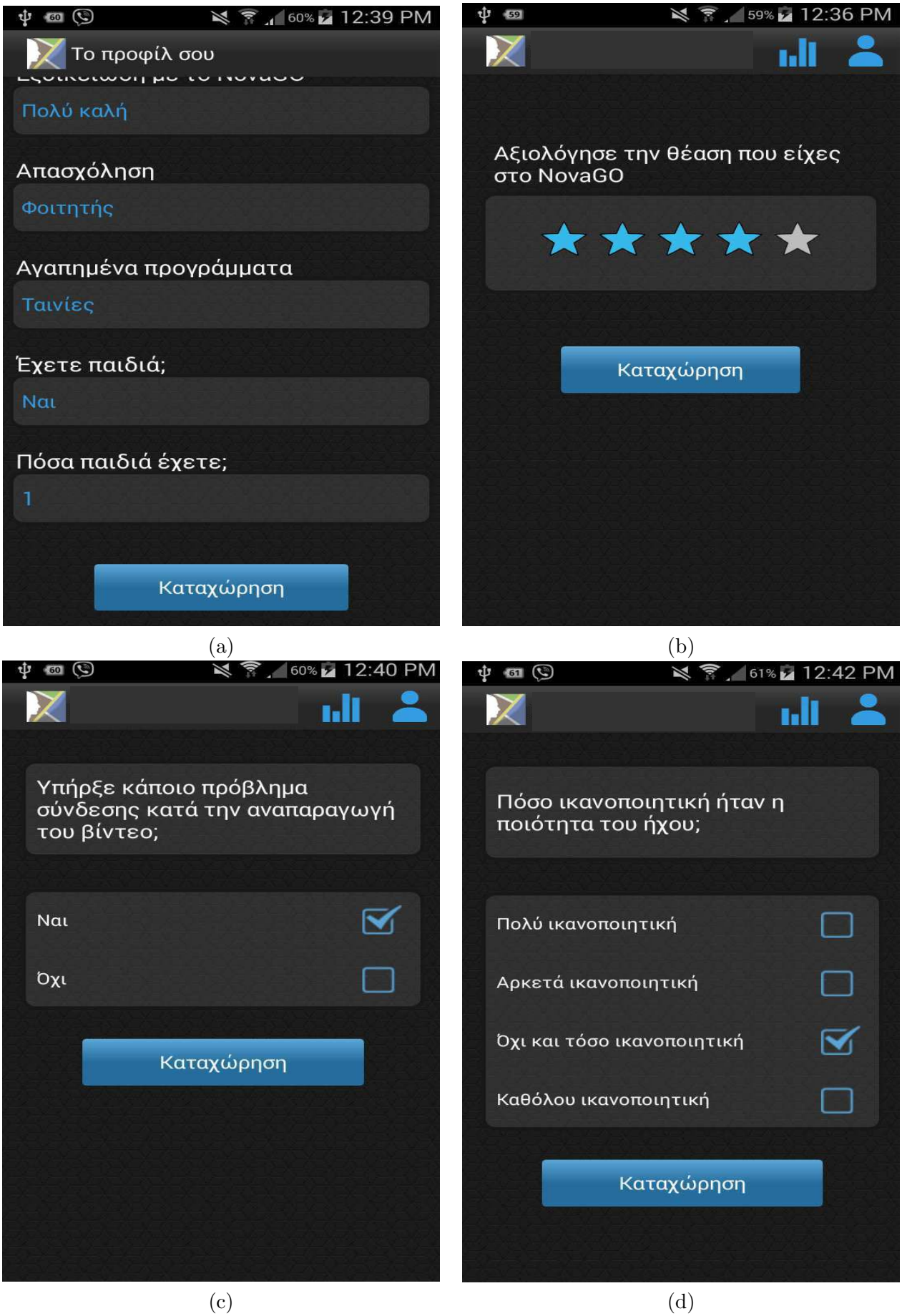


Figure 3.17: GUI screenshots from the uQoE tracker client: (a) questionnaire regarding the user profile, (b) QoE feedback about a video session, (c & d) additional questions regarding the problems that might have been encountered.

Chapter 4

MLQoE Algorithm

The uQoE prediction algorithm QoE is the same as MLQoE algorithm. The MLQoE uses supervised regression, in which the predictors are network metrics, based on RSSI, jitter, and packet loss, and the predicted outcome is the QoE score. The performance metric is the *absolute difference of the predicted QoE score compared to the actual score provided by the user*. Fig. 4.1 presents a general overview of the MLQoE. It consists of several steps, including the normalization, feature selection, training multiple regressors, the selection of the best ML model and the estimation of its performance. The MLQoE is modular, since it includes set of ML algorithms that can be augmented.

The MLQoE aims to find a good prediction model by training ML algorithms. Each such algorithm typically accepts a set of parameters (called hyper-parameters) that affect the complexity of the model. To produce a good predictive model the type of the learning algorithm and the values of the hyper-parameters must be chosen appropriately; otherwise, the predictive performance is impaired, a phenomenon that is referred to as underfitting. To avoid underfitting and improve performance, the MLQoE tries several algorithms and several hyper-parameter combinations and selects the best one automatically. Specifically, the MLQoE employs the Support Vector Regression (SVR), Artificial Neural Networks (ANNs), Decision Trees (DTs), Gaussian Naive Bayes (GNB) classifiers. Unfortunately, estimating the performance of multiple models on the same test set leads to overestimation of the performance of the best-performing model. To provide a conservative estimation, while at the same time avoid underfitting, the MLQoE employs the Nested Cross-Validation (nested CV) protocol [25]. Notice that, the data normalization and feature selection is executed inside the nested CV. The MLQoE takes as input the calls that a user has made along with their corresponding QoE scores. The nested CV then reports the best model for each user (for these calls) along with an estimated conservative bound on its future performance. The following sections present the MLQoE in detail.

4.1 Max-Min Parents and Children for Feature Selection

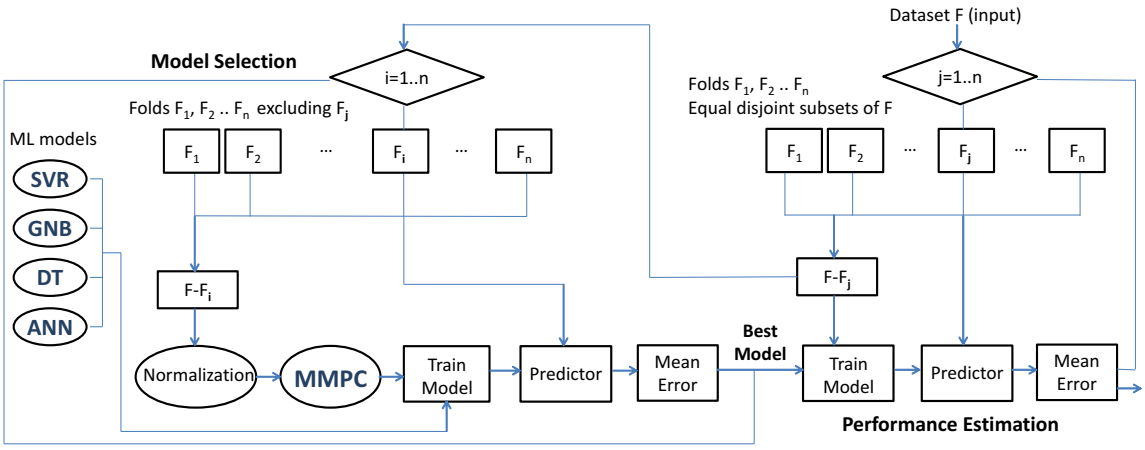


Figure 4.1: The MLQoE consists of two modules, namely, the model selection, and the performance estimation. The model selection takes as input the training set of the performance estimation loop, cross-validates it, and reports the best model. The performance estimation takes as input the dataset, partitions it into folds, estimates the performance of the best model (that the model selection outputs) in each fold and reports (as output) the mean error for the dataset. It can be easily extended to include other ML algorithms.

Feature selection aims to find the minimum set of predictor variables that minimizes the mean absolute error of a model trained with the reduced set of predictors. Feature selection reduces the cost of measuring the required predictor variables. In this work, we employ the Max-Min Parents and Children (MMPC) [51].

The MMPC is based on Bayesian Network theory and tries to identify the parents and the children of the outcome (target) variable T of interest (i.e., QoE) in any Bayesian Network that faithfully represents the joint distribution of the data. A Bayesian network is a directed acyclic graph, where the variables correspond to the nodes of the graph. An edge from a variable X to a variable Y indicates that X is the parent of Y and X provides unique information for predicting Y. The minimum-size, best-predicting variable set for T has been proven to be the set of parents, children, and parents of children of T, also called the Markov Blanket of T. The MMPC approximates the Markov Blanket with the parents and children of T (i.e., the neighbors of T in the Bayesian Network representing the joint distribution of the data).

The algorithm takes as input the data containing the QoE (variable of interest) and network metrics (predictors), and two hyper-parameters: (a) the maximum size of conditioning set k, and (b) the statistical level for accepting probabilistic dependence a. The MMPC determines the best variable set by performing multiple tests of conditional independence of a variable X with T given a subset $\mathbf{Z}$ of other predictors. The null hypothesis of the test is that of conditional independence. If the p-value ($p_{XT|\mathbf{Z}}$) of the test is less than a then the null is rejected, and conditional dependence is accepted. The value of k determines the maximum size of the conditioning sets $\mathbf{Z}$. Since we model the QoE as a continuous variable, we use the Fisher z-test [52] assuming that the partial correlation of two continuous values is zero.

Algorithm 1 MMPC

Input: T the target variable to predict, V the predictors, the maximum size of a conditioning set k, the threshold for rejecting independence a
Output: S is a subset of variables in V

$R = V$ % Remaining to consider
$S = \emptyset$ % Select so far
repeat
  if $\exists X \in R, \mathbf{Z} \subseteq S, |\mathbf{Z}| \leq k, s.t., p_{XT|\mathbf{Z}} > a$ **then**
    $R = R \setminus \{X\}$
  end if
  $M = argmax_{X \in R} min_{\mathbf{Z} \subseteq S \setminus X, |\mathbf{Z}| \leq k}(-p_{XT|\mathbf{Z}})$
  $R = R \setminus \{M\}$
  $S = R \cup \{M\}$
until $R \neq \emptyset$
$\forall X, s.t., \exists \mathbf{Z} \subseteq S \setminus \{X\}, |\mathbf{Z}| \leq k, p_{XT|\mathbf{Z}} > a$
$S = S \setminus \{X\}$
return S

The partial correlation measures the degree of association between two random variables, conditioning on a set of controlling random variables. Typically, the partial correlation between two variables X and Y given a set of n controlling variables $\mathbf{Z}$ = Z1, Z2, ..., Zn, written $\rho_{XY \cdot \mathbf{Z}}$, is the correlation between the residuals R_X and R_Y resulting from the linear regression of X with $\mathbf{Z}$ and of Y with $\mathbf{Z}$, respectively. The partial correlation can be obtained using matrix inversion. This approach allows all partial correlations to be computed between any two variables X_i and X_j of a set $\mathbf{V}$ of cardinality n, given all others, i.e., $\mathbf{V} \setminus \{X_i, X_j\}$, if the correlation matrix (or alternatively covariance matrix) $\Omega = (\omega_{ij})$, where $\omega_{ij} = \rho_{X_i X_j}$, is positive definite and therefore invertible. If we define $P = \Omega^{-1}$, we have:

$$\rho_{X_i X_j \cdot \mathbf{V} \setminus X_i, X_j} = -\frac{p_{ij}}{\sqrt{p_{ii} p_{jj}}} \tag{4.1}$$

The partial correlation $\rho_{XY \cdot \mathbf{Z}}$ is zero if and only if X is conditionally independent from Y given $\mathbf{Z}$. To test if a sample partial correlation $\rho_{XY \cdot \mathbf{Z}}$ vanishes, Fisher's z-transform of the partial correlation used, as follows:

$$z(\rho_{XY \cdot \mathbf{Z}}) = \frac{1}{2} \ln \left(\frac{1 + \rho_{XY \cdot \mathbf{Z}}}{1 - \rho_{XY \cdot \mathbf{Z}}} \right) \tag{4.2}$$

The MMPC begins with the set of the predictor variables as remaining-to-examine variables and a set containing the currently selected variables that is initially empty, as presented in Algorithm 1. Then, until the set of remaining variables becomes empty, the MMPC removes from this set any variable that is independent from the variable of interest, conditioned on any subset of the selected variables. In each iteration, the algorithm heuristically selects a predictor

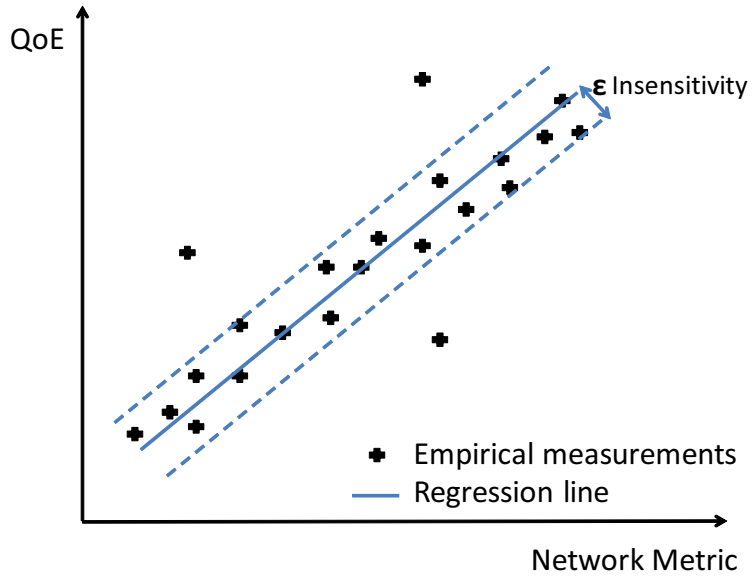


Figure 4.2: An illustration of an SVR with one network metric.

variable and moves it to the currently selected ones.

4.2 Support Vector Regression

The Support Vector Machines (SVMs) [53] is a supervised learning method for classification and regression analysis. The relative insensitivity to the *curse of dimensionality* is one of the advantages of the method. The MLQoE uses SVM algorithms for regression analysis (SVR) machines or ϵ-SVR. The SVR algorithm accepts as hyper-parameters a kernel function $K(x, z)$, where x and z are two vectors of predictors, an *epsilon* parameter, and a cost parameter C. Typical kernel functions include the linear, polynomial of degree d, or Gaussian with variance parameter γ [53]. The SVR solves the optimization problem

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\ &\text{subject to} \quad \begin{cases} y_i - \langle w, \Phi(x)\rangle - b & \leq \quad \epsilon + \xi_i \\ \langle w, \Phi(x)\rangle + b - y_i & \leq \quad \epsilon + \xi_i^* \\ \xi_i, \xi_i^* & \geq \quad 0 \end{cases} \end{aligned} \tag{4.3}$$

where y_i are the output values (QoE) of the training instance i and l is the number of the training instances. Function Φ maps the input training vectors to a constructed feature space; Φ is determined by the choice of the kernel K. The optimization problem is actually converted to its dual before finding a solution, expressed by the following equation:

$$\begin{aligned} \text{maximize} \quad & \begin{cases} -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(x_i, x_j) \\ -\epsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*) \end{cases} \\ \text{subject to} \quad & \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C] \end{aligned} \tag{4.4}$$

The final predictive model is

$$f(x) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)K(x_i, x) + b \tag{4.5}$$

The intuition in SVR is to first map (implicitly through the use of the kernel function and duality theory) the data into the constructed feature space Φ where the problem may be easier. Then, to find a linear regression function (in feature space) $w \cdot \Phi(x) + b$ that has at most *epsilon* deviation from all target values y_i for all training data, while being as flat as possible (in feature space). Training points outside the *epsilon* insensitive region are penalized proportionally to C and their distance from the region. See Fig. 4.2.

4.3 Artificial Neural Networks

Inspired by the structure of biological neural networks, an Artificial Neural Network (ANN) [54] consists of an interconnected group of artificial neurons that model complex relationships between inputs and outputs. The ANNs of this work are feed-forward neural networks with signals propagated only forward through the layers of units.

A typical ANN has three types of layers, namely, the input layer that is fed with the predictor variables (i.e., network metrics), the hidden layers, and the output layer with the unit that emits the outcome (the QoE), respectively. All the layers are interconnected by synapses, associated with weights. Each neuron computes the inner product of the input with the weights on the (simulated) synapses and transfers it to its output after passing through a transfer function, in this work the sigmoid function (except for the output node which uses a linear transfer function).

The weights are adjusted (learned) during the training phase by the backpropagation learning algorithm. This algorithm propagates the error between the estimated QoE and the true QoE backward (from the output layer to the input) and employs gradient descent optimization to minimize the error. This is repeated until the vector of weights best fits the training data.

New samples are fed to the input layer, propagated through the network, and the network outputs the estimated QoE (Fig. 4.3). The number of hidden layers and number of units in each hidden layer are hyper-parameters to be tuned. The ANNs can deal with continuous and discrete domains and solve almost all the regression problems by appropriate tuning of parameters.

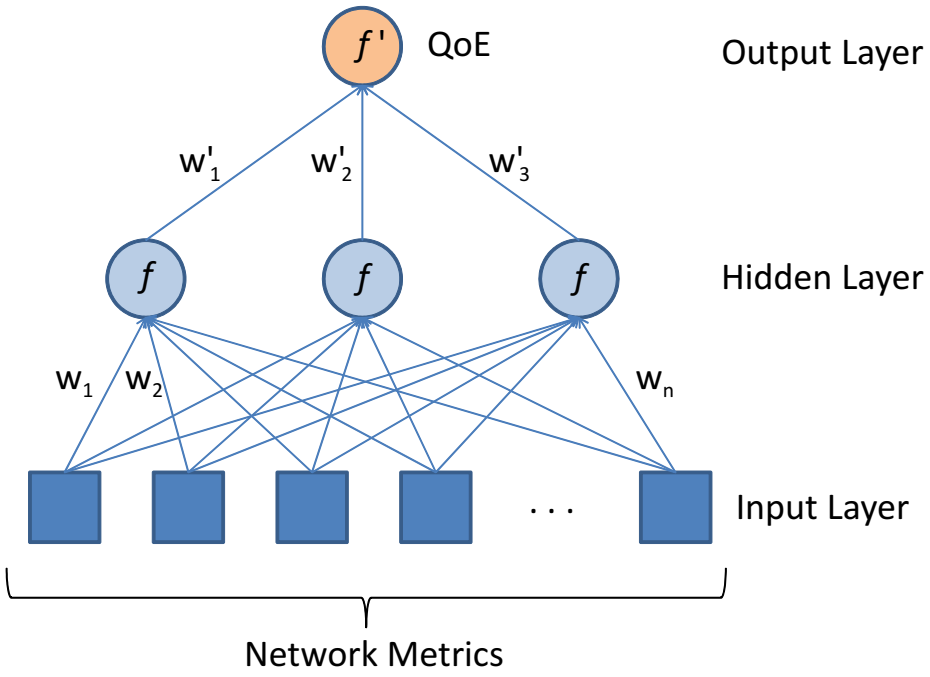


Figure 4.3: An illustration of an ANN with one hidden layer and the weights (w and w').

4.4 Decision Trees

The Decision Trees are prediction models in the form of tree graphs [55]. Each vector x to predict goes through the nodes of the tree. In each node, a value of a variable in x is tested whether it is smaller or higher than a threshold (for continuous variables) (Fig. 4.4). Depending on the outcome the process continues recursively in the left or right sub-tree, until a leaf is encounter. Each leaf contains a prediction that is returned.

The Decision Trees are learnt from the training data typically using a recursive greedy search algorithm. For the root of the tree a variable X and a threshold t are selected using a heuristic. The process continues recursively until a stopping criterion determines that a leaf should be constructed. This work uses regression Decision Trees, which take as input continuous values and estimate continuous outcome. For each leaf of a Decision Tree the outcome is the average value of the set of the examples Y which belongs in this leaf node. The criterion which is used to grow the DT is the mean squared error (MSE). The MSE of a node is

$$MSE = \frac{1}{c}\sum_{i=1}^{c}(Y_i - \overline{Y})^2 \tag{4.6}$$

where c is the number of examples Y in this node and $\overline{Y}$ is the outcome of this node.

During the training phase the model starts with one node containing all the dataset and calculates its MSE. The model will consider all the possible node splits from the data and the MSE for each one will be calculated. The predictor variable which gives the minimum MSE will

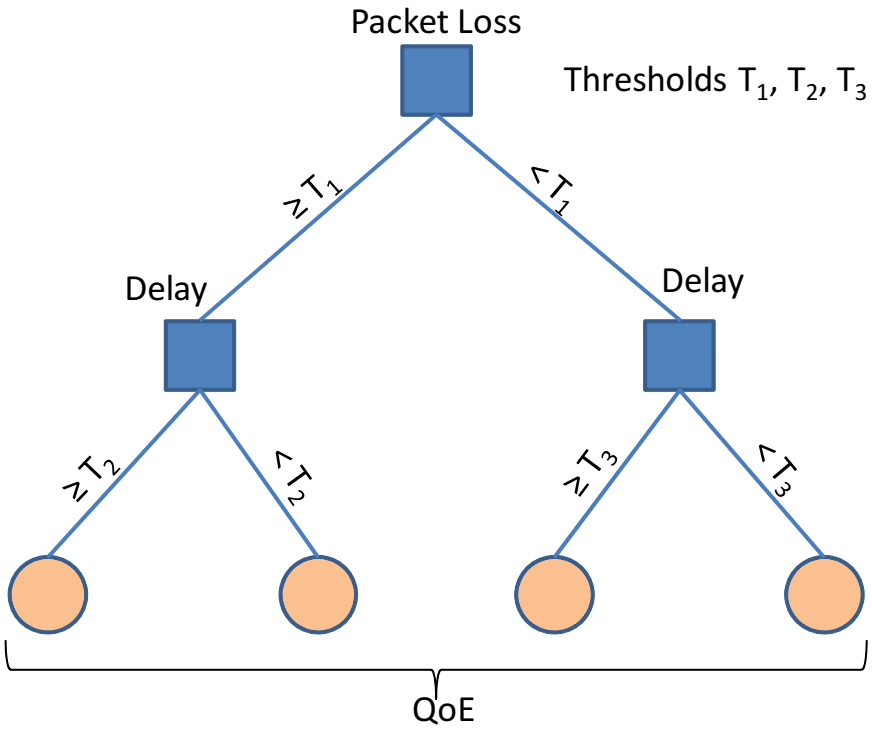


Figure 4.4: An illustration of a Decision Tree based on packet loss and delay.

be selected for the split. This is repeated until the stopping criterion for a node N is satisfied:

$$MSE(N) < qetoler * qed \tag{4.7}$$

where *qetoler* defines the tolerance on quadratic error per node and *qed* is the quadratic error for the entire data.

To avoid overfitting, which occurs when the tree is deep enough to perfectly classify the training samples, the tree is pruned. Specifically, after the Decision Tree is built, we use the *alpha* parameter, a numeric scalar from 0 (no pruning) to 1 (prune to one node). Each node is characterized by a risk value, based on the MSE and weighted by the probability of this node (the number of examples in this node divided to the number of examples in the dataset). For each branch of the tree, the value of the risk is calculated and the branches with small risk are pruned. For a given value of alpha parameter, all the branches with risk less than alpha are removed.

4.5 Gaussian Naive Bayes

The Naive Bayes (NB) classifier [56] is a simple, efficient and widely-used probabilistic algorithm based on the Bayes theorem. For tractability of training, it assumes independence between the variables conditioned on the class. For the different predictors $X_1, X_2 ... X_n$ and the different

labels $Y_1...Y_k$, given the conditionally independence assumption, the probabilistic model of the NB can be described as follows:

$$P(X_1...X_n|Y) = \prod_{i=1}^{n} P(X_i|Y) \tag{4.8}$$

The Bayes Rule is formulated as follows:

$$P(Y = y_k|X_1...Xn) = \frac{P(Y = y_k) \prod_{i=1}^{n} P(X_i|Y = y_k)}{\sum_{j=1}^{k} P(Y = y_j) \prod_{i=1}^{n} P(X_i|Y = y_j)} \tag{4.9}$$

In the case of continuous input, the model typically assumes that the data follow the Gaussian distribution. The conditional probabilities now become Gaussian conditional densities

$$p(X_i = x|Y = y_k) = \frac{1}{\sqrt{2\pi\sigma_{ik}^2}} e^{\frac{-(x-\mu_{ik})^2}{2\sigma_{ik}^2}} \tag{4.10}$$

In the training phase, the GNB takes as input the set of samples which consists of the predictors variables $X_1...X_n$ (i.e., network metrics) and the outcome of each sample Y, in order to estimate the parameters of the conditional densities.

In the testing phase, when a new sample Y_{new} is introduced, the classification rule can be rewritten as follows:

$$Y_{new} = argmax_{Y_k} P(Y = y_k) \prod_{i=1}^{n} p(X_i|Y = y_k) \tag{4.11}$$

The GNB returns the QoE with the maximum *aposteriori* probability, while the rest of the algorithms are regression ones. Therefore, we used the expected value of the GNB estimates to make our results comparable.

4.6 Nested Cross-Validation

The nested Cross-Validation (nested CV) algorithm is an analysis protocol that selects the best training algorithm and hyper-parameter values and estimates the performance of the final, returned model [25, 57–59].

To explain nested CV let us first consider a simplified scenario. Let us assume that we train multiple algorithms and combinations of their hyper-parameters on a train set. We then apply the learned models on a separate test set and estimate their performances. We select the best-performing algorithm (with performance P^* on the test set) and corresponding hyper-parameters and train on *all* data, i.e. the train and the test data, to learn a final model. What is the estimated performance of the final model? If we use P^* as our estimate, it is overoptimistic (upward biased) since it is the maximum of performances computed on the same test set. For

Algorithm 2 Nested Cross-Validation

Input: Dataset F, Integer k, Set of hyper-parameter combinations A
Output: Mean absolute error

partition F randomly into k equal-sized disjoint subsets F_i
for $i = 1$ to k **do**
  $F' = F \setminus F_i$
  for each a $\in$ A **do**
    for $j = 1$ to $k - 1$ **do**
      $T = F' \setminus F_j$
      train the ML algorithm with hyper-parameters a on Set T
      evaluate the trained model to T_j obtaining performance P_j
    end for
    $M(a) = \text{mean}(P_j)$
  end for
  select $\hat{a} = argmax_a M(a)$
  train the ML algorithm with hyper-parameters $\hat{a}$ on Set F'
  evaluate the trained model to F_i obtaining performance P_i
end for
return mean(P_i)

a conservative estimation, we need a second test set that we use only once to estimate the performance of the final model only. Let us rename the first test set that we use for selecting the best algorithm as validation set, and the second test set that we use for estimating performance as test set. Overall, the data have been partitioned to training, validation, and test sets.

The nested CV protocol is a CV extension of the above procedure and presented in Algorithm 2. It partitions the data to different folds F_i forming the set of folds F. Each fold F_i serves once as a test set (outer loop), and for each F_i, each other fold F_j serves as a validation set (inner loop). For the performance estimation, the models are trained on $F \setminus \{F_i\}$. For determining the best-performing algorithm, the models are trained on $F \setminus \{F_i\} \setminus \{F_j\}$ folds. The performances are averaged over all folds to produce more accurate estimates.

Table 4.1: Computational complexity of the algorithms in the runtime phase (m is the number of the network metrics and e the number of examples in training phase)

Algorithms	**Runtime**
SVR	$O(m)$
GNB	$O(1)$
DT	$O(log(e))$
ANN	$O(m)$
E-Model	$O(1)$
Normalization	$O(m)$

4.7 Computational Complexity

The runtime phase of the MLQoE (and all the ML algorithms) is of negligible computational complexity in practice, specifically SVR $O(m)$, GNB $O(1)$, DT $O(log(e))$, ANN $O(m)$, E-Model $O(1)$, Normalization $O(m)$ (where m is the number of the network metrics and e the number of examples in training phase). In contrast, their training phase is of relatively high computational complexity (especially, in the case of SVR, and ANN), though it is performed *off-line*. The complexity of the E-model is low but reports an aggregate QoE estimate. While the PESQ has a high computational complexity [60].

Chapter 5

Evaluation of the MLQoE Prediction

5.1 Parameter Tuning

To evaluate the prediction accuracy of the uQoE in video streaming service, for the first field study, we used the collected datasets and applied the uQoE algorithm in an aggregate and a user-centric manner. In this dataset, only 13 out of 20 users have rated five sessions or more. We consider only them for the performance analysis of the uQoE, so the ML algorithms can be trained and tested. The aggregate approach considers all the video sessions for all users. Due to the small number of samples for some users in the user-centric approach (in the first dataset), we used the leave-one-out nested CV (LOOCV) with random partitioning to folds. For the aggregate approach, a 10-fold nested CV with random partitioning to folds was applied. For the second dataset (i.e., collected in the second field study), in the user-centric approach, we used a 10-fold nested CV, for the evaluation of the prediction, considering all 20 users (since each user has assessed 50 video sessions).

Apart from the original datasets, a normalized version is also maintained. The normalization is performed to handle the variability across the metrics. It transforms the values of each metric to fit a normal distribution of zero mean and unit variance. Each ML algorithm has a number of tuning parameters [5]. At the model selection process, all the combinations of the different parameters are tested. The MMPC algorithm is tested with $k = 0, 1, 2, 3$ and $a = 0.01, 0.05, 0.1, 1$ (a value $a = 1$ corresponds to selecting all variables without feature selection). Each dataset is employed to train the ML algorithms. The LIBSVM Version 3.14 was used for the implementation of the e-SVR algorithm; the hyper-parameters were chosen as follows: for the Gaussian, linear, and polynomial kernels were used with the default values, the cost C was selected among the values $\{0.01, 0.1, 1, 10, 100\}$, and the insensitivity parameter e within values $\{0.05, 0.1, 0.25, 0.5, 1\}$. The ANN was implemented with one hidden layer. The number of nodes for the hidden layer varied from 8 up to 11 and 2 up to 5 [54] for the first and the second field study, respectively. The CART implementation have been used for the DTs and we tested the following values of the alpha $a = \{0.1, 0.01, 0.05\}$, the *qetoler* was set in the default value (i.e., 1e-6).

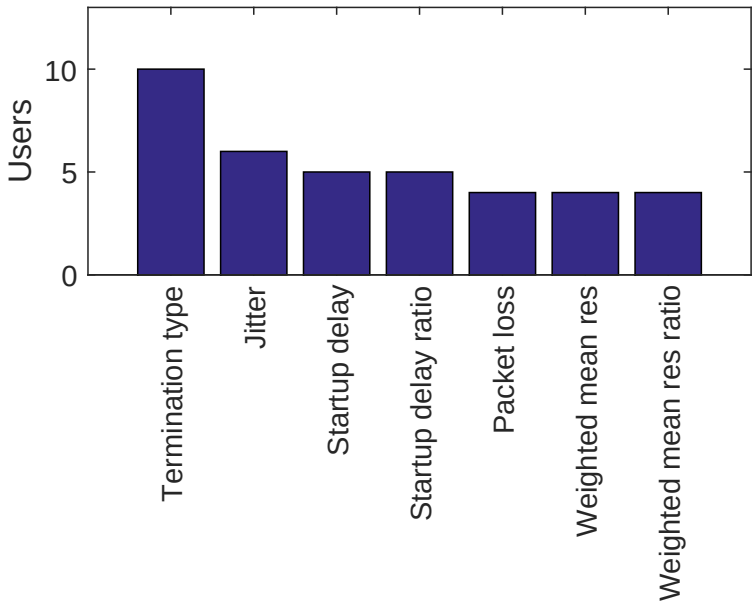


Figure 5.1: The metrics derived from the MMPC in the final prediction models for the first field study.

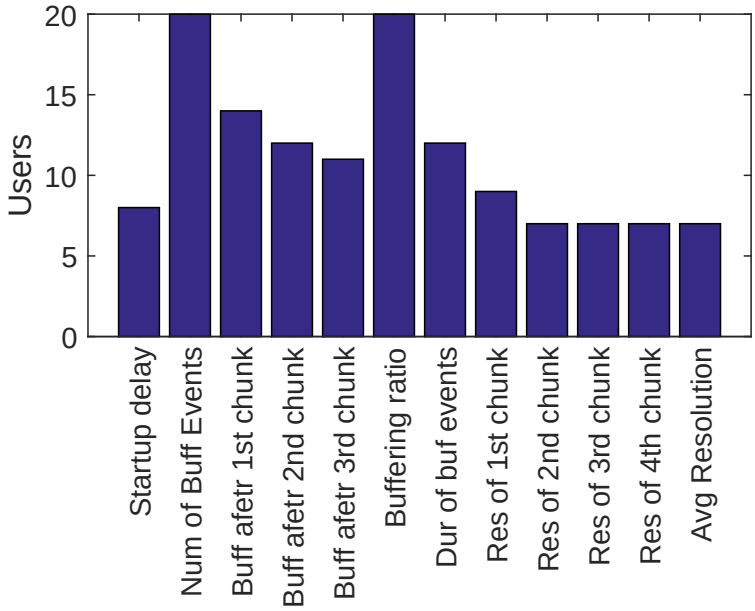


Figure 5.2: The metrics derived from the MMPC in the final prediction models for the second field study.

5.2 Parameter Impact

In the aggregate MLQoE, in the context of the first field study, the MMPC indicates that the parameters with dominant impact on the QoE are the termination type of the session, the buffering events frequency, the weighted mean video resolution ratio, and the packet loss. Considering the

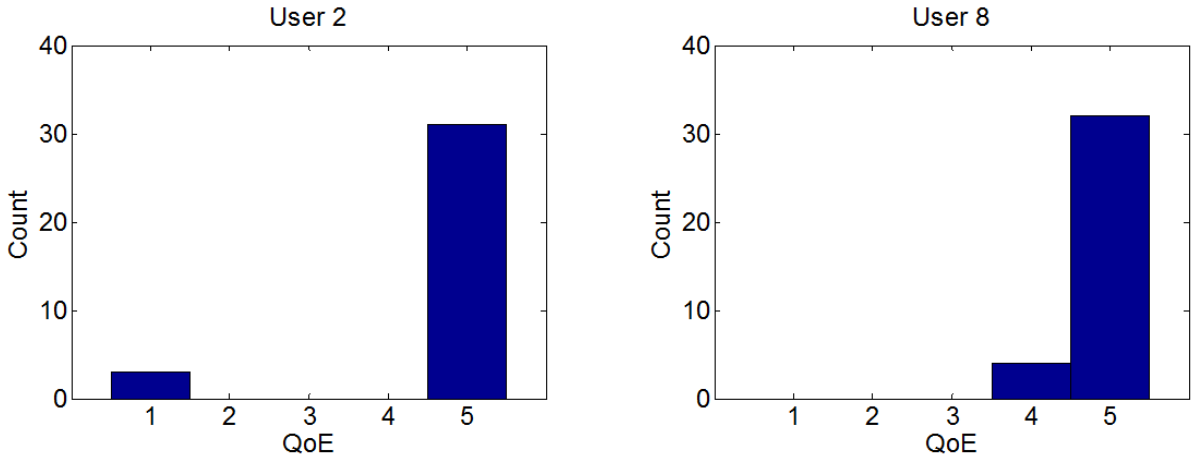
13 subjects of the first field study, the uQoE reported the termination type as a dominant factor for 10 subjects, the mean jitter for 6 subjects, the startup delay and its ratio over the entire video duration for 5 subjects, the packet loss for 4 subjects, and the weighted mean video resolution and its ratio of the weighted mean video resolution over the display for 4 subjects. The diversity in the dominant parameters between the aggregate case vs. the user-centric one is due to the use of different datasets (Fig. 5.1). Specifically, the aggregate approach employs all the video sessions for all subjects, while in the user-centric approach, each subject views a different set of videos sessions, under different network conditions and context.

In the second field study, the number of buffering events and buffering ratio consistently are the parameters with the most prominent impact on the QoE across all users (Fig. 5.2).

5.3 Accuracy of the prediction

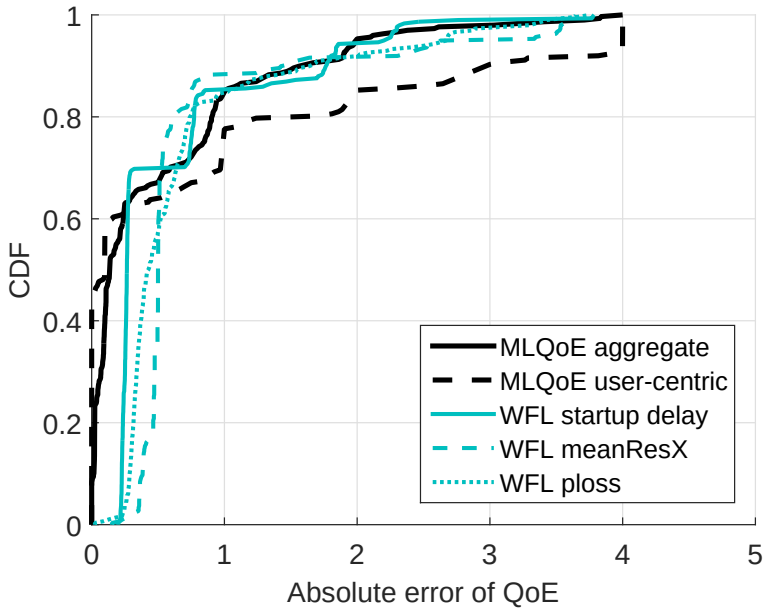
For the first field study, the user-centric uQoE can predict the QoE in a fairly accurate manner with a median and mean absolute error of 0.0991 and 0.7716, respectively. The aggregate uQoE reports a median and mean absolute error of 0.1392 and 0.5185 (Fig. 5.4). The better mean performance of the aggregate uQoE compared to the user-centric approach is due to the specific limitations of this field study (as discussed also in Section 3). Specifically, only ten users have rated more than 15 sessions. Moreover, for several users, the number of distinct QoE scores is very limited (Fig. 5.3). These characteristics impact the training of the data mining algorithms and result in large prediction errors.

For the second dataset, the user-centric uQoE can predict the QoE score with a median and mean absolute error of 0.5517 and 0.6133, respectively (Fig. 5.6). We compared the MLQoE to the WFL, a state-of-the-art QoE model. The performance of the WFL was evaluated using a 10-fold cross-validation and one parameter as input (the packet loss or the mean resolution). The aggregate uQoE exhibits a statistically significant better performance than the WFL in terms of mean and median prediction, while the user-centric MLQoE outperforms WFL in terms of median prediction error. Although the WFL does not capture the interplay and impact of the multiple factors (e.g., mean packet loss, mean video resolution), it still has a reasonably good performance.



(a) (b)

Figure 5.3: Example of the QoE distribution for two of the users for the first field study.



(a)

Algorithm	Mean	Median	Std
MLQoE aggregate	0.5185	0.1392	0.7624
WFL startup delay	0.6183	0.3655	0.6654
WFL packet loss	0.7103	0.4292	0.7173
WFL mean weighted resolution	0.7489	0.4996	0.7563
MLQoE user-centric	0.8026	0.1000	1.2672

(b)

Figure 5.4: The absolute error derived from the WFL and MLQoE for the first field study.

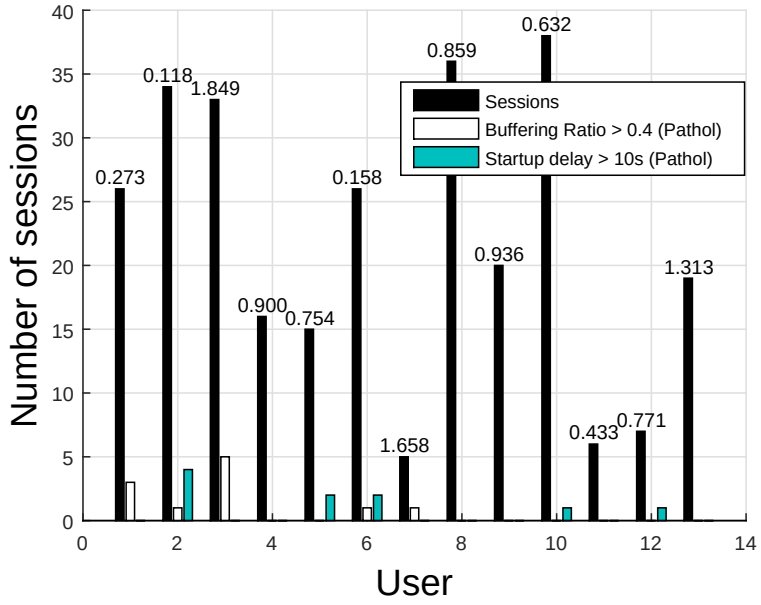
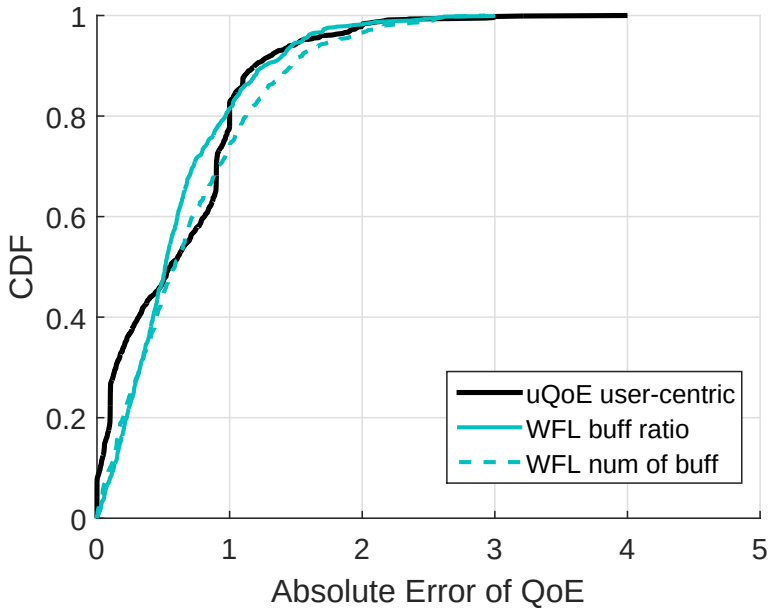


Figure 5.5: The mean absolute error of the uQoE per user, indicated at the top of each column, considering all his/her sessions (left column), the sessions with high bufferring ratio and QoE score of 5 (middle), and the sessions with high startup delay and QoE score of 5 (right column).



(a)

Algorithm	Mean	Median	Std
MLQoE user-centric	0.6133	0.5517	0.5479
WFL buffering ratio	0.6986	0.5322	0.5787
WFL number of buffering events	0.7300	0.6175	0.5575

(b)

Figure 5.6: The absolute error derived from the WFL and MLQoE for the second field study.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The weighted mean resolution, termination type, startup delay, and bufferring ratio affect the QoE. Sessions with startup delay higher than 10 sec obtain lower QoE scores, while sessions with buffering ratio, have typically a smaller duration. The last 15 sec of a session seems to have a strong correlation with the termination due to poor connectivity. In several sessions, we observed that a rate adaptation could reduce the buffering ratio and improve the QoE. In the first field study, the parameters with a dominant impact on the QoE are the frequency of buffering events, weighted mean video resolution ratio, termination type, and packet loss (considering the aggregate prediction model). The median error in the QoE prediction is less than 0.1. In the second (more controlled) field study, the number of buffering events and buffering ratio are the parameters of prominent impact on QoE for each user. The sensitivity of users to the different types of impairment varies across users. Moreover we observed the presence of lenient and strict users (in terms of their QoE assessments) .

The performance of these two field studies enabled us to reflect about the tradeoffs between small-scale studies with homogeneous settings in non-controlled environments and larger-scale (potentially crowd-sensing/sourcing participatory) studies that can reach more people, representing a more realistic set of scenarios/conditions but with several unknown difficult to control dynamic exogenous parameters and heterogeneous settings. Obtaining reliable measurements in such crowd-sourcing/sensing non-controlled field studies can be challenging. In general, it is difficult to obtain the "ground truth" about the QoE. The above tradeoffs also highlight the tension between subjectivity and reliability in the collected measurements.

Through the uQoE tracker, system, and algorithms, the provider can learn more about its customers (e.g., their traffic, usage pattern, end-to-end network performance, QoE requirements/profile), infrastructure and service performance. This can enable the provider to improve the adaptation mechanisms, provide better customer service, assess its agreements with infrastructure/network providers, and potentially perform better pricing.

6.2 Future Work

We plan to apply the MLQoE prediction algorithm in a larger and more complete dataset with videos produced in a similar manner as in FORTH study. We also plan to incorporate the uQoE prediction in the u-map, for bidirectional VoIP calls, in order to be decreased the feedback from the users and to assist in anomaly detection algorithms. The u-map follows the clientserver architecture. The u-map client application runs on Android smartphones and is used as a recommendation tool for selecting the best provider for making a VoIP call (based on the evaluations of users). It detects the start and termination of VoIP calls, performs cross-layer network measurements (during the VoIP call), and requests QoE feedback at the end of each VoIP call (as in the case of skype). These data are regularly uploaded to the u-map server.

We also plan to extend this work by integrating the QoE models with economic parameters (e.g., cost of service) and contextual ones (e.g., time and location). A long-term objective is to apply the proposed methodology for assessing the QoE of various services (e.g., web browsing, teleconference/telepresence applications, gaming) using non-intrusive methods and "indirect" metrics for inferring the QoE. The present work sets a methodological framework that dynamically models the user assessments to predict the QoE for various multimedia network services.

Appendix A

Questionnaire and Android Application for the Audiovisual Tests

This Appendix presents the conduction of the audivisual tests for the second dataset. The audiovisual tests were performed in a small meeting room at the ICS-FORTH. Each subject was left alone in the room, sat, and used the headphones which were set at a specific volume level. Ideally, the auditory test should have been conducted in an isolated free of echo, reverberations, and noise room. Since this was not possible, the room, display settings and headphones setup were selected to minimize the environmental noise and ensure clear video reproduction. To enable the subjects to grade more consistently, each session consists of two phases, namely, the training phase and the grading one. In the training phase, the subjects have to watch the reference file and five training samples in order to familiarize themselves with the degradation in the quality of segments. In the grading phase, the subjects have to watch the samples and grade them using the MOS scale. After grading a sample, the subjects cannot watch it again or change the grade. The 20 % of the samples are repeated segments for consistency check.

The audiovisual test of the dataset 2 was conducted using an Android application that we implemented. In this application, the subject first had to answer a short questionnaire (Fig. A.1 (a)). In the first part of the questionnaire, the subject provided some demographic information (e.g., age, sex, familiarity with video streaming services, mobile applications, and audiovisual tests). Then, she/he had to continued with the sessions one and two (Fig. A.1 (e)). The subject had to read and follow the instructions appearing in the screen (Fig. A.1 (f)), before watch the training samples. Then, the subject had to grade all the samples (Fig. A.1 (g)) and submit the results via the Android application (Fig. A.1 (h)).

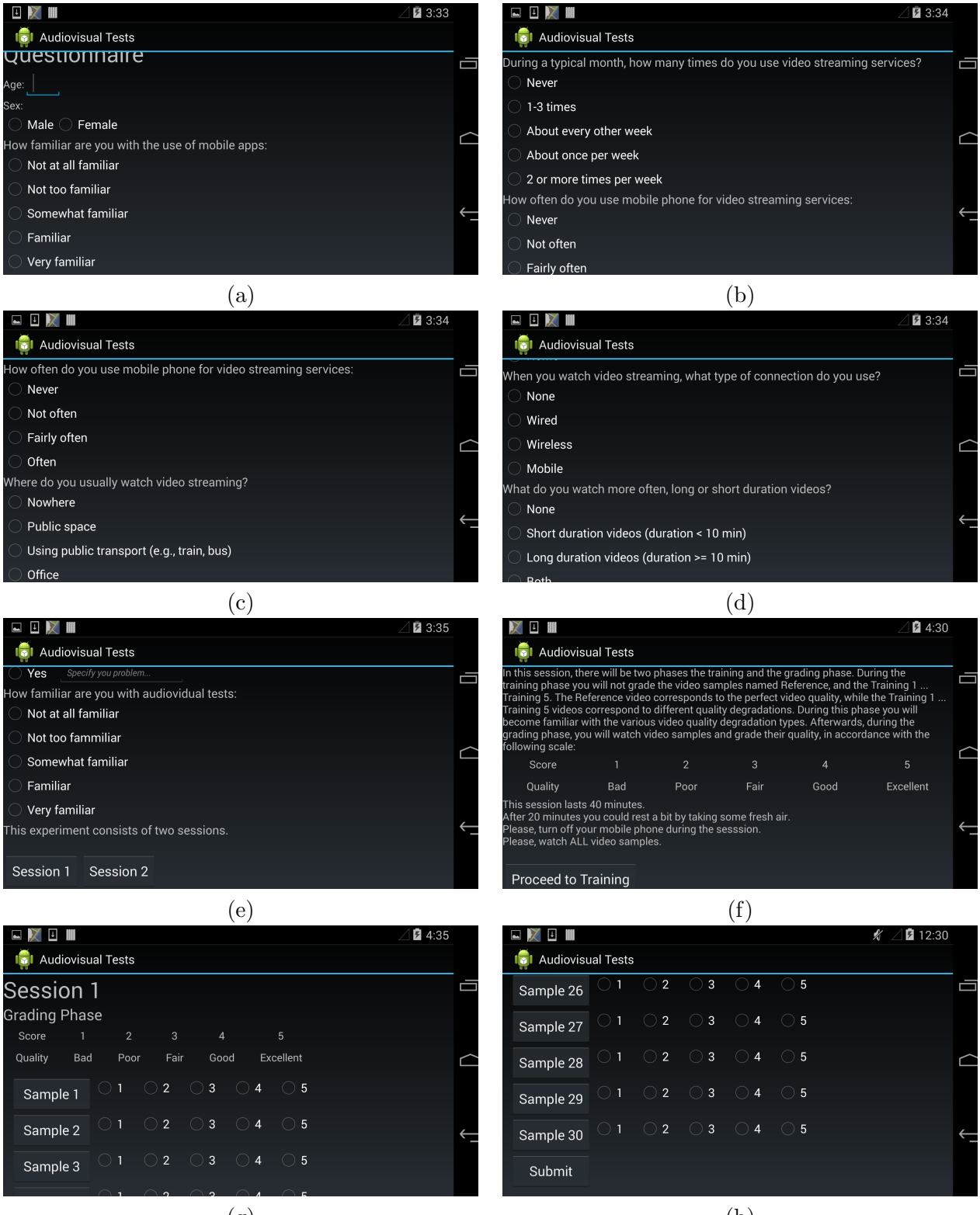


Figure A.1: Screenshots of the Android application for the audiovisual test of the dataset 2.

Bibliography

[1] CISCO, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2014-b•"2019," February 2015.

[2] CISCO, "The Zettabyte Era: Trends and Analysis, 2014–2019." May 2015.

[3] S. Möller and A. Raake, *Quality of Experience*. Springer, 2014.

[4] S. S. Krishnan and R. K. Sitaraman, "Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs," *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 6, pp. 2001–2014, 2013.

[5] P. Charonyktakis, M. Plakia, I. Tsamardinos, and M. Papadopouli, "On user-centric modular qoe prediction for voip based on machine-learning algorithms," *IEEE Trans. on Mobile Computing*, 2015.

[6] P. Charonyktakis, "On user-centric modular qoe prediction for voip based on machine-learning algorithms," in *Msc thesis*. University of Crete, 2015.

[7] A. Makrogiannakis, "The development of applications for smart-phones and their performance analysis using empirical measurements," in *Msc thesis*. University of Crete, 2011.

[8] I. Tsompanidis, G. Fortetsanakis, T. Hirvonen, and M. Papadopouli, "A comparative analysis of the perceived quality of voip under various wireless network conditions," in *WWIC10*, Luleå, Sweden, June 1-3 2010.

[9] P. Reichl, S. Egger, R. Schatz, and A. D'Alconzo, "The logarithmic nature of qoe and the role of the weber-fechner law in qoe assessment," in *Communications (ICC), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1–5.

[10] I. Tsompanidis, G. Fortetsanakis, T. Hirvonen, and M. Papadopouli, "Analyzing the impact of various wireless network conditions on the perceived quality of voip," in *Local and Metropolitan Area Networks (LANMAN), 2010 17th IEEE Workshop on*. IEEE, 2010, pp. 1–6.

[11] M. H. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *Broadcasting, IEEE Transactions on*, vol. 50, no. 3, pp. 312–322, 2004.

[12] L. Sun and E. Ifeachor, "New models for perceived voice quality prediction and their applications in playout buffer optimization for voip networks," in *Communications, 2004 IEEE International Conference on*, vol. 3. IEEE, 2004, pp. 1478–1483.

[13] L. Sun and E. C. Ifeachor, "Voice quality prediction models and their application in voip networks," *Multimedia, IEEE Transactions on*, vol. 8, no. 4, pp. 809–820, 2006.

[14] M. Fiedler and T. Hoßfeld, "Quality of experience-related differential equations and provisioning-delivery hysteresis," in *21st ITC specialist seminar on multimedia applications-Traffic, performance and QoE, Miyazaki, Japan*, 2010.

[15] P. Reichl, B. Tuffin, and R. Schatz, "Logarithmic laws in service quality perception: where microeconomics meets psychophysics and quality of experience," *Telecommunication Systems*, vol. 52, no. 2, pp. 587–600, 2013.

[16] D. Hands and M. Wilkins, "A study of the impact of network loss and burst size on video streaming quality and acceptability," in *Interactive Distributed Multimedia Systems and Telecommunication Services*. Springer, 1999, pp. 45–57.

[17] 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 *Conference on Internet Measurement Conference*, 2014.

[18] F. Wamser, M. Seufert, P. Casas, R. Irmer, P. Tran-Gia, and R. Schatz, "Yomoapp: A tool for analyzing qoe of youtube http adaptive streaming in mobile networks," in *Networks and Communications,*, 2015.

[19] V. Menkovski, A. Oredope, A. Liotta, and A. C. Sánchez, "Optimized online learning for qoe prediction," in *Proc. of the 21st Benelux conference on artificial intelligence*, 2009.

[20] V. Menkovski, A. Oredope, A. Liotta, and A. C. Sánchez, "Predicting quality of experience in multimedia streaming," in *Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia*. ACM, 2009, pp. 52–59.

[21] V. Menkovski, G. Exarchakos, and A. Liotta, "Online qoe prediction," in *Quality of Multimedia Experience (QoMEX), 2010 Second International Workshop on*. IEEE, 2010, pp. 118–123.

[22] 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 Performance Evaluation Review*, vol. 42, 2014.

[23] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang, "Developing a predictive model of quality of experience for internet video," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 339–350.

[24] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern classification. 2nd," *Edition. New York*, 2001.

[25] I. Tsamardinos, A. Rakhshani, and V. Lagani, "Performance-estimation properties of cross-validation-based protocols with simultaneous hyper-parameter optimization," in *Artificial Intelligence: Methods and Applications*. Springer, 2014, pp. 1–14.

[26] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, "Understanding the impact of video quality on user engagement," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 362–373, 2011.

[27] K.-T. Chen, C.-C. Tu, and W.-C. Xiao, "Oneclick: A framework for measuring network quality of experience," in *INFOCOM 2009, IEEE*. IEEE, 2009, pp. 702–710.

[28] H. H. Song, Z. Ge, A. Mahimkar, J. Wang, J. Yates, Y. Zhang, A. Basso, and M. Chen, "Q-score: Proactive service quality assessment in a large iptv system," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. ACM, 2011, pp. 195–208.

[29] T. Hoßfeld, M. Seufert, M. Hirth, T. Zinner, P. Tran-Gia, and R. Schatz, "Quantification of youtube qoe via crowdsourcing," in *Multimedia (ISM), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 494–499.

[30] J. Hecht, "All smart no phone: Cellular carriers are dragging their heels over technology to improve voice quality," *IEEE Spectrum*, pp. 30–35, 2014.

[31] S. Jumisko-Pyykkö and M. M. Hannuksela, "Does context matter in quality evaluation of mobile television?" in *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*. ACM, 2008, pp. 63–72.

[32] M. H. Pinson, L. Janowski, R. Pépion, Q. Huynh-Thu, C. Schmidmer, P. Corriveau, A. Younkin, P. Le Callet, M. Barkowsky, and W. Ingram, "The influence of subjects and environment on audiovisual subjective tests: An international study," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 6, no. 6, pp. 640–651, 2012.

[33] W. Wu, A. Arefin, R. Rivas, K. Nahrstedt, R. Sheppard, and Z. Yang, "Quality of experience in distributed interactive multimedia environments: toward a theoretical framework," in *Proceedings of the 17th ACM international conference on Multimedia*. ACM, 2009, pp. 481–490.

[34] J. Xue and C. W. Chen, "A study on perception of mobile video with surrounding contextual influences," in *IEEE QoMEX 2012*.

[35] R. L. Mandryk, K. M. Inkpen, and T. W. Calvert, "Using psychophysiological techniques to measure user experience with entertainment technologies," *Behaviour & Information Technology*, vol. 25, no. 2, pp. 141–158, 2006.

[36] G. M. Wilson and M. A. Sasse, "Do users always know whatb••s good for them? utilising physiological responses to assess media quality," in *People and Computers XIVb•"Usability or Else!* Springer, 2000, pp. 327–339.

[37] Y. Chen, Q. Chen, F. Zhang, Q. Zhang, K. Wu, R. Huang, and L. Zhou, "Understanding viewer engagement of video service in wi-fi network," *Computer Networks*, vol. 91, 2015.

[38] W. Cherif, A. Ksentini, D. Négru, and M. Sidibe, "A_psqa: Pesq-like non-intrusive tool for qoe prediction in voip services," in *Communications (ICC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2124–2128.

[39] K. Mitra, C. Ahlund, and A. Zaslavsky, "Qoe estimation and prediction using hidden markov models in heterogeneous access networks," in *Telecommunication Networks and Applications Conference (ATNAC), 2012 Australasian*. IEEE, 2012, pp. 1–5.

[40] L. Sun and E. C. Ifeachor, "Perceived speech quality prediction for voice over ip-based networks," in *Communications, 2002. ICC 2002. IEEE International Conference on*, vol. 4. IEEE, 2002, pp. 2573–2577.

[41] C.-C. Wu, K.-T. Chen, C.-Y. Huang, and C.-L. Lei, "An empirical evaluation of voip playout buffer dimensioning in skype, google talk, and msn messenger," in *Proceedings of the 18th international workshop on Network and operating systems support for digital audio and video*. ACM, 2009, pp. 97–102.

[42] A. P. Markopoulou, F. A. Tobagi, and M. J. Karam, "Assessment of voip quality over internet backbones," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1. IEEE, 2002, pp. 150–159.

[43] R. Birke, M. Mellia, M. Petracca, and D. Rossi, "Understanding voip from backbone measurements," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*. IEEE, 2007, pp. 2027–2035.

[44] A. Arjona, C. Westphal, A. Yla-Jaaski, and M. Kristensson, "Towards high quality voip in 3g networks-an empirical study," in *Telecommunications, 2008. AICT'08. Fourth Advanced International Conference on*. IEEE, 2008, pp. 143–150.

[45] G. Rubino, P. Tirilly, and M. Varela, "Evaluating usersb•• satisfaction in packet networks using random neural networks," in *Artificial Neural Networks–ICANN 2006*. Springer, 2006, pp. 303–312.

[46] 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 *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*. ACM, 2014, p. 18.

[47] K. Mitra, A. Zaslavsky, and C. Ahlund, "Context-aware qoe modelling, measurement and prediction in mobile computing systems," 2013.

[48] A. Bhattacharya, W. Wu, and Z. Yang, "Quality of experience evaluation of voice communication: an affect-based approach," *Human-centric Computing and Information Sciences*, vol. 2, no. 1, pp. 1–18, 2012.

[49] D. Joumblatt, J. Chandrashekar, B. Kveton, N. Taft, and R. Teixeira, "Predicting user dissatisfaction with internet application performance at end-hosts," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 235–239.

[50] M. O. Lorenz, "Methods of measuring the concentration of wealth," *Publications of the American statistical association*, vol. 9, no. 70, 1905.

[51] C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos, "Local causal and markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation," *The Journal of Machine Learning Research*, vol. 11, pp. 171–234, 2010.

[52] P. Spirtes, C. N. Glymour, and R. Scheines, *Causation, prediction, and search*. MIT press, 2000, vol. 81.

[53] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[54] T. M. Mitchell, "Artificial neural networks," *Machine learning*, pp. 81–127, 1997.

[55] Mitchell, Tom M, "Decision tree learning," *Machine learning*, pp. 52–80, 1997.

[56] T. M. Mitchell, "Bayesian learning," *Machine learning*, pp. 154–200, 1997.

[57] A. Statnikov, C. F. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy, "A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis," *Bioinformatics*, vol. 21, no. 5, pp. 631–643, 2005.

[58] V. Lagani and I. Tsamardinos, "Structure-based variable selection for survival data," *Bioinformatics*, vol. 26, no. 15, pp. 1887–1894, 2010.

[59] A. Statnikov, I. Tsamardinos, Y. Dosbayev, and C. F. Aliferis, "Gems: a system for automated cancer diagnosis and biomarker discovery from microarray gene expression data," *International journal of medical informatics*, vol. 74, no. 7, pp. 491–503, 2005.

[60] ITU, "P.862: Perceptual evaluation of speech quality (pesq): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs."