Master's Thesis

Monitoring Network Congestion in Wi-Fi, based on QoE in HTTP Video Steaming Services

Muhammad Umar Nasir

Department of Electrical and Information Technology,
Faculty of Engineering, LTH, Lund University, 2016.

LUNDS UNIVERSITET
Lunds Tekniska Högskola

Master's Thesis

Monitoring Network Congestion in Wi-Fi, based on QoE in HTTP Video Steaming Services

Author:
Muhammad Umar Nasir

Supervisor:
Prof. Maria Kihl (EIT)
Examiner:
Prof. Stefan Höst (EIT)

Department of Electrical and Information Technology
Faculty of Engineering, LTH, Lund University
SE-221 00 Lund, Sweden

July, 2016

Department of Electrical and Information Technology
Lund University
Box 118, SE-221 00 LUND
SWEDEN

This thesis is set in Computer Modern 11pt,
with the LaTeX Documentation System

Printed in Sweden by *Tryckeriet i E-huset*, Lund
July 2016.

Abstract

Improvements in Internet technology, development of multimedia applications, protocols and improvement in user devices have led to the popularity of multimedia applications, among which video streaming applications are most popular. Streaming video services are sensitive to network conditions, thus making Quality of Experience (QoE) of end users sensitive to network conditions. QoE is affected by small disturbances in network conditions and end users observe this as blurred video or lost scenes. This may lead to end users giving up the service or switching to another network operator. To avoid this, network operators and service providers need to maintain QoE at a satisfactory level. The purpose of this study is to develop a monitoring method, which can monitor network congestion in Wi-Fi, based on QoE in HTTP video streaming services. This study proposes a QoE assessment method based on Machine Learning (ML), which allows network operators and service providers to predict QoE from network level measurements.

This study was conducted in four steps. Initially, network monitoring probes were designed to measure key metrics that affect QoE, which involved development of QoE assessment model based on relationship between Quality of Service (QoS) and QoE, and implementation of an active measurement protocol called Two-Way Active Measurement Protocol (TWAMP) for network level measurements. Subsequently, a direct link was established between subjective QoE and objective network measurements by designing various test cases. Data was collected by performing network measurements on a Wi-Fi testbed to study the impact of wireless rate adaptation and link utilization on QoE by loading WLAN with cross traffic on downlink or bi-directional paths along with YouTube video. A ML approach was then used to classify network level measurements into QoE levels. A set of ML algorithms: SVM, KNN and Logistic Regression were tested and evaluated to build a classification model to be used in network monitoring system module within network management system. Ultimately, the performance of the proposed QoE assessment method was evaluated using five test cases.

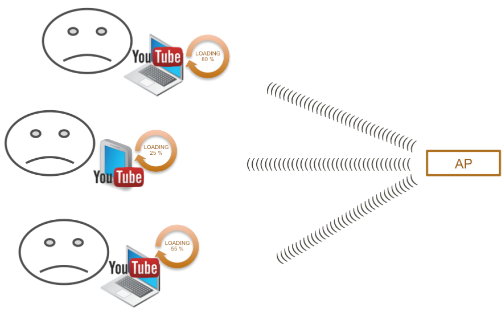
The results show that this method performed well and give high classification accuracy in all cases. Outputs from this work may be used by network operators and service providers to modify their network management system by developing effective congestion management solutions to bring back QoE to satisfactory level.

Keywords- QoE; Video Streaming; QoE-QoS Relationship; Packet Loss Pattern; RTT; Active Measurements; TWAMP; Machine Learning Approach.

YouTube Video in Dense Wi-Fi Network - Loading ...

Loading ... Loading ... Loading ... is what we observe while watching streaming videos in areas like downtown, shopping malls, university, airports, etc., while connected to Wi-Fi. This causes irritation to users if the video loads a lot or takes too much time to load. This study proposes a method, which predicts user experience. Network operators and service providers can use this prediction to manage their networks more efficiently.

Video streaming applications are very popular among multimedia applications. Video traffic on the Internet is predicted to grow further and the share of video is expected to be 82% as compared to other applications by 2020. Video requires high bit rates. Video consumes high bandwidths of about more than 10 times as compared to other popular applications, for example, Facebook and music-streaming applications. If there is not enough bandwidth, users observe *video re-buffering*, while watching videos. This is more common in low capacity networks like Wi-Fi. Consequently, leading to bad user experience and as a result, users can either switch network operator or quit watching video. Thus, there is an increasing interest from network operators and service providers to monitor user satisfaction.



User experience low due to video re-buffering

This study is aimed to design a method to predict user experience of YouTube video users, using Wi-Fi as access network. The results show that this method performed well and give high accuracy for all test cases. Prediction of user experience is a first step in user satisfaction. It is very important for the network operators and service providers to predict user experience. Consequently, they can tune their services accordingly to bring back the user satisfaction to acceptable level. This study provides means for network operators and service providers to predict user experience.

Acknowledgements

This Thesis would not exist without the support and guidance of

- Supervisor: **Maria Kihl** for introducing me to world of Academic Research. Her leadership, commitment, attitude, assistance, discussions and feedback have encouraged me to do my best. She has been an inspiration and will continue to inspire me in future.
- Examiner: **Stefan Höst** his support and feedback.
- **Jens Andersson** for his support and help with technical problems while conducting experiments at Bredbandsgruppens forskningslaboratorium.

Finally, I would like to express my sincere gratitude to my family for their love, support and encouragement.

Muhammad Umar Nasir   Lund, July 6, 2016

Contents

List of Figures

List of Tables

List of Acronyms

3GPP	Third Generation Partnership Project
2G	Second Generation
3G	Third Generation
4G	Fourth Generation
ANN	Arificial Neural Network
AP	Access Point
BER	Bit Error Rate
CBR	Constant Bit Rate
CDN	Content Distribution Networks
CPU	Central Processing Unit
DASH	Dynamic Adaptive Streaming over HTTP
EIT	Elektro och Informationsteknik
FTP	File Transfer Protocol
HD	High Definition
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPDV	Inter Packet Delay Variance
IPPM	IP Performance metric
ISP	Internet Service Provider
ISPs	Internet Service Providers
ITU	International Telecommunication Union
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
KNN	K-Nearest Neighbor

KPI	Key Performance Indicator
LAN	Local Area Network
LTH	Lunds Tekniska Högskola
ML	Machine Learning
MOS	Mean Opinion Score
MTU	Maximum Transmission Unit
NPM	Network Performance Metric
NPMs	Network Performance Metrics
OTT	Over-The-Top
OWAMP	One-Way Active Measurement Protocol
PSNR	Peak Signal to Noise Ratio
QoE	Quality of Experience
QoS	Quality of Service
RBFN	Radial Basis Function Networks
RAM	Random Access Memory
RBF	Radial Basis Function
RFC	Request for Comments
RFE	Recursive Feature Elimination
RSSI	Recieved Signal Strength Indicator
RTOs	Retransmission Time Outs
RTT	Round Trip Time
SLA	Service Level Agreement
SSIM	Structural Similarity
SVC	Support Vector Classification
SVM	Support Vector Machine
TCP	Transmission Control Protocol
TV	Television
TWAMP	Two-Way Active Measurement Protocol
UDP	User Datagram Protocol
UE	User Equipment
UMTS	Universal Mobile Telecommunications Services
URL	Uniform Resource Locator
VOIP	Voice over Internet Protocol
VQM	Video Quality Metric
Wi-Fi	Wireless Fidelity
WLAN	Wireless LAN

1

Introduction

INTERNET HAS BECOME a global medium of communication that covers all aspects of life. Internet Protocol (IP) traffic is growing at a very high rate since decades. The demands on broadband access networks have increased both regarding bandwidth and Quality of Service (QoS). The traffic has increased more than fivefold in the past five years, and is expected to increase three fold over the next five years [1].

The innovations in technology are leading to change in the Internet usage trends [2]. Current Internet usage trends show that users not only use traditional elastic applications such as web, e-mail and File Transfer Protocol (FTP) applications, but also trends show an exponential increase in usage of multimedia applications with more emphasis on applications like social networking, video streaming applications, remote access and online transactions. This trend is likely to increase in the future with the popularity of Over-The-Top (OTT) applications. An increasing number of users recently have gained interest in video streaming applications (for example, YouTube[1]), Internet television (on services like Netflix[2], Huhu[3], Kankan[4]), interactive applications (for example online gaming), video chat (for example, Skype[5]) and cloud storage applications.

Streaming media content already dominate global traffic mix and is ex-

[1] `www.youtube.com`
[2] `https://www.netflix.com/se-en/`
[3] `http://www.hulu.com/`
[4] `http://www.kankan.com/`
[5] `http://www.skype.com/en/`

pected to be nearly doubled by 2020 [1]. With improvements in Internet technology, development of multimedia applications, improvements in design and features of User Equipment (UE), video streaming services have become one the most popular services. Many of users watch sports, Television (TV) programs and news on online streaming services. In 2015, IP video traffic was 70% which is predicted to be 82% of all consumer Internet traffic by 2020 [1].

Video applications are bandwidth intensive. With this increase in IP data, user experience while using non-elastic applications can greatly be affected as Internet was originally designed for elastic applications, which can tolerate variations in throughput and loss while multimedia applications traffic is composed of high volumes with various traffic mixes and is bursty in nature.

Each type of application has its own service requirements of network performance, i.e., QoS. The service requirements of video streaming applications varies from that of traditional elastic applications. They require high throughput that can consume the available bandwidth. Wireless networks, for example Wireless Fidelity (Wi-Fi), have lower capacity as compared to their wired counter parts. Thus, they are heavily loaded due to amount of video data, which leads to network congestion.

End users expect ubiquitous delivery of high quality services (high speed and reliable network performance). Data traffic is increasing rapidly both in volume and per user subscription. Currently, more and more users are connected and each user has multiple devices, resulting in a huge volume of IP traffic. The number of devices connected to IP networks are expected to be three times as high as the global population by 2020 [1]. With development in mobile communication standards, the devices are either connected to IP networks by Third Generation Partnership Project (3GPP) or Wi-Fi access networks.

Thus, there is a need to address users' high bandwidth demands. Also, with the popularity of smart phones, tablets, laptops and many other mobile gadgets, most of the users are connected to wireless networks. Cisco predicts that the traffic from wireless and mobile devices will account for 66% total IP traffic by 2020 [1]. Huge volumes of data is constantly crossing through wireless access networks by UEs as users are mobile and prefer to use their devices on the move.

Network operators and service providers are searching for effective and efficient data communications. They face challenges to maintain their services with users' growing bandwidth requirements and are continuously updating their networks. They need to plan how to deal with users' increased data requirements, manage their system to deal with high data and keep upgrading their systems with advancements in technology while keep expanding the coverage area. Their networks should be capable to support all applications especially popular applications, which are sensitive to network parameters like Voice over Internet Protocol (VOIP), video streaming, gaming and video conferencing. Video streaming is one of the most challenging service for network operators to deliver with assured service levels especially in wireless networks due to high Bit Error Rate (BER) and bandwidth constraints. In mobile networks, capacity is growing at a much slower pace as compared to the explosive increase in data traffic. In order to deal with this, mobile operators employ Wi-Fi into their network core for data offloading. It is a very cost-effective approach and operators use it to provide an immediate capacity relief to the congested areas in their network.

Success of network operator and service providers partly depends on the QoS offered. The QoS should be robust enough to cope with poor radio conditions. Quality of Experience (QoE) is the main criteria for adoption of a service as it is the end users who determine the success of service. Network operators and service providers have gained interest in QoE aware management of networks in order to fulfill end user demands and gain competitive edge in market.

1.1 Problem Formulation

In order for network operators and service providers to provide their services in the best possible way i.e., to achieve end user satisfaction, it is crucial for them to monitor how their services are perceived by end users. User perception is subjective in nature and therefore, it is a challenging task for network operators and service providers to assess end users satisfaction. Low QoE can result in end user dis-satisfaction that can lead to end users quitting the service and switching to other competing network operator. Thus, there is a need for developing an effective monitoring method by which network operators and service providers can monitor the perception of end users in real-time. This can be a key element and a first step in end user satisfaction. This monitoring method should be based on parameters that could be measured precisely and are reliable enough to provide a realistic view of video streaming service, as perceived by the end-users.

Internet supports a large variety of multimedia applications that can be classified as in Figure 1.1.

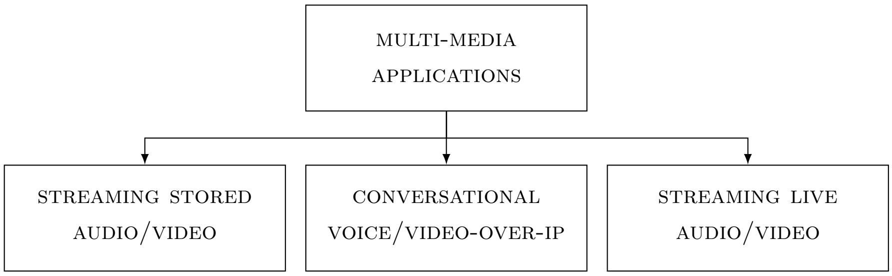


Figure 1.1: Types of Multimedia Network Applications

This Thesis will focus only on Hypertext Transfer Protocol (HTTP) video streaming services (streaming stored video) due to their popularity and growing number of end-users.

1.2 Methodology

This Thesis consists of four steps. The first step focuses on designing network monitoring probes to measure Key Performance Indicator (KPI) that affect QoE. This includes:

- In-depth study of QoE.
- Identifying QoS metrics that affect QoE and can be measured at network level.
- Development of QoE assessment model.
- Implementing a method to extract KPIs.

The second step focuses on the establishment of a link between subjective QoE and objective network measurements. This includes

- Designing test cases.
- Data collection by performing network measurements and recording subjective QoE in Wi-Fi testbed for supervised learning.

The third step focuses on building a classifier, which translates network level measurements into QoE levels. This includes:

- Building classification model using three Machine Learning (ML) algorithms: Support Vector Machine (SVM), K-Nearest Neighbor (KNN) and logistic regression.

Finally, the fourth step focuses on evaluating the performance of QoE assessment method in various test cases.

1.3 Thesis Contribution

QoE monitoring is very important for network operators and service providers. They need to ensure if end users are satisfied with the services provided by them. With the popularity of streaming media applications and high bandwidth consumed by video as compared to other applications, monitoring QoE is gaining more significance. Low QoE can lead to customer deterioration. Monitoring video QoE is the first step network operators and service providers need to take in order to provide high QoE to end users. This Thesis provide a method of *monitoring network congestion in Wi-Fi based on QoE in video streaming services.* Network operators and service providers can use the information from QoE assessment for effective congestion management in case of low QoE for example, to manage network dynamically by traffic steering or other methods to bring back QoE to satisfactory level.

1.4 Limitations

The monitoring method proposed in this thesis has the following limitations:

1. QoE is subjective in nature. In this Thesis, QoE is considered as a function of derivatives of packet loss (loss distance and loss period) and RTT. Other parameters influencing QoE are assumed to be constant.

2. This method have not taken into account losses due to lossy coding techniques and the losses due to the dynamic nature of IP networks.

3. Only packet losses at access network and Access Point (AP) queue are considered.

4. Experiments are conducted at video quality of 480p only.

5. Static network conditions are assumed during network measurements.

1.5 Thesis Outline

This Thesis is divided into six chapters. After introducing the reader to the field, the remainder of this thesis has the following structure:

- **Chapter 2** covers theoretical background on key concepts, such as HTTP video streaming, QoE, network measurements and Machine Learning to help the reader to understand the basic terminology used throughout the thesis along with related work that have already been carried out to determine QoE.
- **Chapter 3** describes how an end user observes network congestion, proposed approach adapted in this thesis to monitor network congestion and video QoE metrics.
- **Chapter 4** describes the proposed QoE assessment method and the steps leading to development of it.
- **Chapter 5** includes the experiment test bed, methodology that has been used for data collection and development of experimental cases.
- **Chapter 6** illustrates an analysis of selected KPIs, building classification models and performance evaluation of proposed QoE assessment model.
- Finally, **Chapter 7** gives conclusion and discusses possible improvements and future work proposals.

2

Background and Related Work

VIDEO STREAMING APPLICATIONS are very popular among multimedia applications. Current user trends show an increase in the popularity of video applications. Video traffic already dominate global traffic mix. Video consumes high bandwidth as compared to the other popular applications, for example, Facebook, music-streaming applications. Due to high bandwidth requirement, end users are likely to observe re-buffering events in dense Wi-Fi network. Consequently, this can lead to bad QoE. As a result, end user can switch to another network operator. This has increased the interest of network operators and service providers to monitor QoE. But,

- What is QoE?
- How does HTTP video streaming work?
- How can the network operators and service providers predict QoE?
- What is the need of network measurements?
- What are network measurement methods?
- What is Machine Learning (ML)?

This chapter will give a brief overview of theoretical background on these key concepts, which are needed to understand the thesis problem and solution. Also, the related work that has been carried out in other researches is also discussed in this chapter.

2.1 HTTP Video Streaming

HTTP streaming videos consist of videos such as movies, music, user generated video (as in YouTube), which are stored on an HTTP server as an ordinary file with a specific Uniform Resource Locator (URL). Streaming videos have distinct features like streaming, interactivity and continuous playout. When end users want to see a video, the following steps takes place, which can be seen in Figure 2.1:

1. User i.e., HTTP client establishes a Transmission Control Protocol (TCP) connection with the server by issuing an HTTP GET request for URL of that specific video.

2. The server then sends information about the metafile within an HTTP response message.

3. Browser launches the media player and passes it the meta file.

4. The media player uses the URL in the metafile and establishes a TCP connection with the server and sends the HTTP request to access the video file.

5. The video starts to play at client's media player after the bytes in the client buffer exceeds a pre determined threshold called *playback time*.

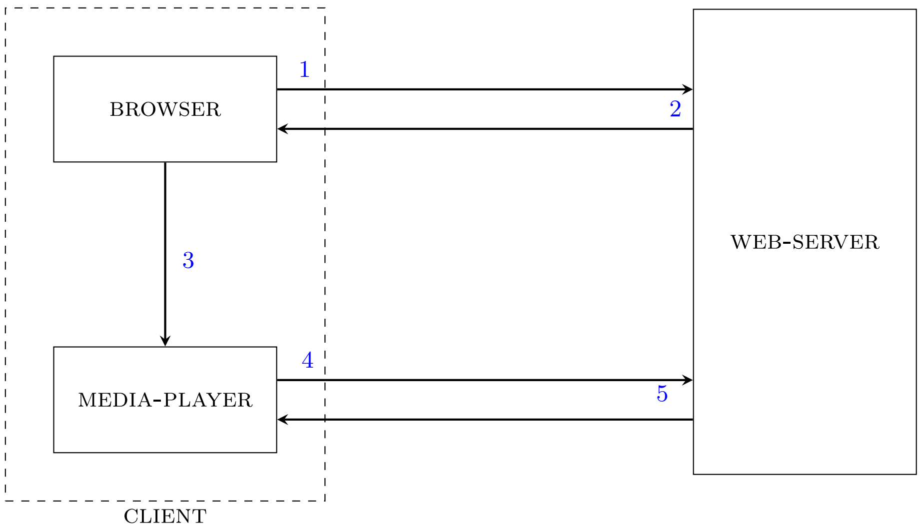


Figure 2.1: HTTP Video Streaming

Modified form of HTTP based streaming is called Dynamic Adaptive Streaming over HTTP (DASH). In DASH, a video is coded in several different versions, each version with different bit rate leading to different quality levels. Using this method, the client can request chunks of video segments of few seconds in length from different versions depending on how much bandwidth is available to the client. When the available bandwidth is high, the client can select chunks from a high-rate version while when available bandwidth is low, the client can select chunks from a low-rate version.

2.2 QoE Assessment

QoE [3] is a complex concept. QoE is subjective in nature and is a function of both technical and non-technical parameters. QoE comprises of complete end-to-end system effects including a UE and a user which makes it subjective in nature. As it is user dependent, there are a lot of factors that can affect user perception and the QoE can be different for each user. QoE relates to users feelings and experience. Some of the factors affecting user perception can be seen in Figure 2.2.

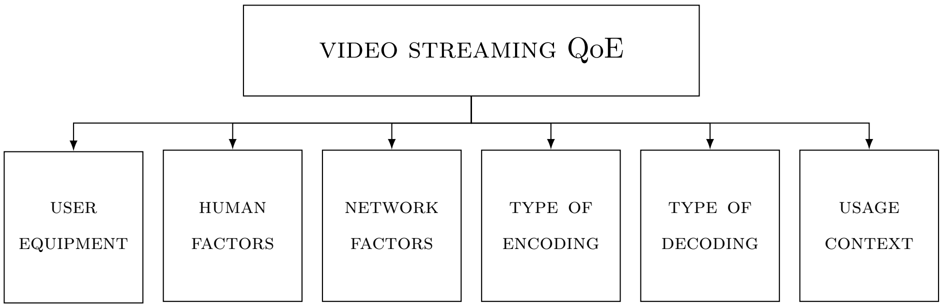


Figure 2.2: Contributing factors in Video Streaming QoE

There are two types of methods of QoE assessment.

1. Subjective Assessment
2. Objective Assessment

2.2.1 Subjective Assessment

QoE is subjective in nature and is expressed by users in qualitative terms i.e., excellent, good, bad. User perception varies among different users and can even vary for the same user over time. Subjective assessment is a time consuming process. These assessment methods cannot monitor in real time because at least 15 subjects and a special facility are required [4]. Subjective evaluation of services can be obtained by various methods like user complaints, interview, surveys, subjective tests, etc [5]. Subjective assessments are time consuming.

Mean Opinion Score (MOS) [6] described in Appendix A (on page 75), is a popular subjective assessment method used widely in research. MOS is a numerical scale that translates user qualitative option to quantitative scale.

2.2.2 Objective Assessment

Objective assessment methods are computational models that use measurable parameters to assess the user experience. These models take some parameters of video and network conditions as input and then predict QoE. These methods are different from the QoS assessment as they takes into account several additional parameters beside network performance indicators (for example, as loss, delay), in order to assess the user-perceived performance of a service. The objective assessment requires the following steps:

1. Computational models, which take into account several parameters for example, network dependent factors, application specific factors, context of use, user expectation, prior expectations etc., in order to approximate the perceived quality.

2. Methods are required, which allow these models to be implemented practically in a real-time scenario.

Objective computational models can be categorized into:

1. Full Reference Models.

2. Reduced Reference Models.

3. No Reference Models.

2.2.2.1 Full Reference Models

Full Reference models estimate QoE using source video and degraded video signals. These models need source video information.

2.2.2.2 Reduced Reference Models

Reduced Reference models estimate QoE using degraded video signals and quality features derived from the source video.

2.2.2.3 No Reference Models

No Reference models estimate QoE using only degraded video signals.

Objective assessment methods are typically used in planning and designing of networks, applications and UE.

2.3 Network Measurements

The Internet Engineering Task Force (IETF) IP Performance metric (IPPM) working group has proposed several metrics and procedures for accurately measuring metrics for network measurements. Each Network Performance Metric (NPM) is measured in terms of certain sub-metrics or KPIs which are obtained by measuring network performance. The following Request for Comments (RFC)s have been published:

- Connectivity (RFC 2678)[1]
- One-way Delay (RFC 2679)[2]
- One-way Packet Loss (RFC 2680)[3]
- Round-trip Delay (RFC 2681)[4]
- One-way Loss Patterns (RFC 3357)[5]
- IP Packet Delay Variation (RFC 3393)[6]
- Packet Reordering Metrics (RFC 4737)[7]
- Round-trip Packet Loss (RFC 6673)[8]

2.3.1 The Need for Network Measurements

Internet is no longer dominated by transferring data such as FTP applications. Now, the Internet supports multi service communications i.e., various traffic mix. From network point of view, video streams and other services traffic transfer throughout a network via IP packets. There are various

[1] https://tools.ietf.org/html/rfc2678
[2] https://tools.ietf.org/html/rfc2679
[3] https://tools.ietf.org/html/rfc2680
[4] https://tools.ietf.org/html/rfc2681
[5] https://tools.ietf.org/html/rfc3357
[6] https://tools.ietf.org/html/rfc3393
[7] https://tools.ietf.org/html/rfc4737
[8] https://tools.ietf.org/html/rfc6673

applications in the Internet and each application have its own service requirements. Each service is sensitive to network disturbances of a certain level in the network. Video streaming quality changes in relation to network parameters (for example, transmission errors, buffering, etc.) and non-network parameters (e.g., coding issues). Therefore, there is a need of network measurement of performance parameters to support operation, management, testing and planning of networks. Also, verification of Service Level Agreement (SLA) have gained huge importance with the evolution of UE and network applications. Thus, it is important for the network operators and service providers to ensure that the QoS is abiding SLA by monitoring network properties (link characteristics, etc.). In order to guarantee QoS, network operators use Network Performance Metrics (NPMs).

2.3.2 Network Measurement Methods

There are two network measurement approaches:

1. Active Measurement
2. Passive Measurement

2.3.2.1 Active Measurements

Active measurement approaches determine end-to-end QoS by injecting additional traffic into the the network. Because of this they consume some of the bandwidth of network that can be used for end-user applications. Active measurement protocols typically inject additional packets into the network from one network end-point to another end-point with timestamps and other information. The packet send and received timestamp along with other fields are compared in order for deriving metrics of interest.

2.3.2.2 Passive Measurements

Passive measurement approaches gather data by listening to network traffic at hubs, link splitters or by monitoring buffers and routers to duplicate a link's traffic. Most of the network devices have built-in passive measurement mechanisms, which are used to gather different type of data from devices such as the number of sent bytes, lost packets and other interface statistics.

2.3.3 Two-Way Active Measurement Protocol (TWAMP)

Two-Way Active Measurement Protocol (TWAMP) defined in RFC 5357 [7] is an active measurement approach to measure two way IP performance between two hosts in a network. It is currently a proposed standard[9] and it is in a development phase and will probably be standardized in near future. TWAMP is based on One-Way Active Measurement Protocol (OWAMP) [8] and adds round-trip measurement capabilities. TWAMP architecture comprises of two hosts and does not require synchronization of clocks of hosts participating in the protocol. TWAMP is used to measure the quality of an IP network and to gather knowledge of its characteristics. Network metrics like latency, packet loss, RTT, etc., can be extracted using TWAMP. TWAMP consists of two protocols namely

1. TWAMP-Control

2. TWAMP-Test

2.3.3.1 TWAMP-Control

TWAMP-Control protocol initiates, starts and stops test sessions and extract results.

2.3.3.2 TWAMP-Test

TWAMP-Test protocol exchanges test packets between two network nodes used to obtain metrics.

Figure 2.3 shows the logical model of TWAMP, which comprises of Control-Client, Session-Sender, Session-Reflector and Server. These are described below:

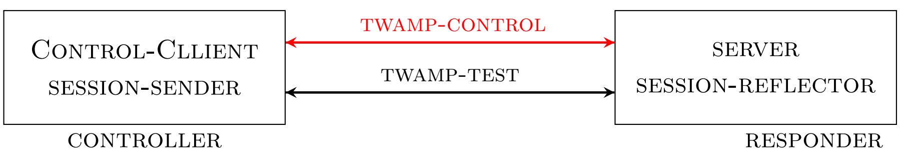


Figure 2.3: Logical Model of TWAMP

[9] as of 03/03/2015

2.3.3.2.1 Control-Client

Control-Client is a network node that starts and stops TWAMP test sessions.

2.3.3.2.2 Session-Sender

Session-Sender is a network node, which sends test packets to the Session-Reflector and receives test packets from Session-Reflector during a test session. All metrics are obtained, analyzed and published by Session-Sender only.

2.3.3.2.3 Session-Reflector

Session-Reflector reflects test packets sent by Session-Sender, as part of a test session.

2.3.3.2.4 Server

Server is a network node, which facilitates one or more test sessions.

2.4 Popularity of HTTP Video Streaming

HTTP video streaming is the most popular among multimedia applications and it is fast growing as it is a better way of presentation of information as compared to text or images/graphics. Streaming media content dominates the global traffic mix. Streaming media have further gained more popularity with video based social networks like YouTube. Video is used as a medium of communication for education, marketing, entertainment, business purposes etc. YouTube alone is responsible for more than 30% of the overall Internet traffic [9].

With the popularity of YouTube, Internet Service Providers (ISPs) try on their end to provide high QoE for example, Google uses Content Distribution Networks (CDN) and has employed massively distributed server infrastructures to replicate content and make it accessible from different Internet locations. Google has deployed thousands of servers inside Internet Service Provider (ISP) through their Google Global Cache approach.

There is a lot research work carried out how to increase the QoE of HTTP video streaming applications in Wi-Fi such as in [10], where a study of traffic patterns of YouTube in a large municipal Wi-Fi network is conducted. Detailed analysis of demand in different geographical areas over time are presented in [10]. The study has proposed that the QoE will increase by serving repeated requests for YouTube videos from caches placed either at network head end, at the wireless AP or in the UE and then to optimize caching performance by exploiting the content demand locality. Authors in [11] talk about caching policies to provide sufficient QoE in order to meet the QoS requirements in case of heavily loaded networks.

There is a lot of research in caching and there are challenges involved. Caching reduces network load but low access network capacity (in case of Wi-Fi) still remains an issue.

Even with this, in order to meet demands of user high band-width requirements, network operators and service providers are also required to take some measures.

2.5 Machine Learning

Machine Learning (ML) is an interdisciplinary field, which includes the study of computer algorithms that provide semi-automated extraction of knowledge from data. ML enables computers to *learn* from previous data and *adapt or modify* their decisions (e.g., making predictions) accordingly. This is done by building models from ML algorithms. These models learn from previous data and then make data driven decisions based on their learning, thus increasing the accuracy of making correct decision where accuracy is how well the predicted decision matches to the correct decision. ML is divided into two main categories:

1. Supervised Learning
2. Unsupervised Learning

2.5.1 Supervised Learning

Supervised learning is the most common type of learning. These algorithms are used when the goal is to predict an specific outcome. This is accomplished in two steps. In the first step, the ML model is first trained using labeled data during which the model learns the relationship between features of data and labels. This is called *model training.* In the second step, predictions are made on out of sample data based on the model training in the first step. This is also known as *predictive modeling.* The goal of supervised learning to to build a model that generalizes i.e., accurately predict the labels of future outcomes based on learning from labeled data. There are two types of supervised learning:

1. Classification
2. Regression

2.5.1.1 Classification

A classification system predicts the label of out of sample data to be one of N classes based on its learning from previous data of each class during the model building step. In classification, labels have categorical values or discrete values. This means that mostly all output space is covered and every sample value is going to belong to one class.

2.5.1.2 Regression

A regression system predicts the label out out of sample data by first finding a mathematical function describing a curve/line, so that the curve/ line passes as close as possible to all the data points and then by using *interpolation*. In regression, labels are ordered and continuous.

2.5.2 Unsupervised Learning

Unsupervised learning algorithms are used when the goal is to extract a structure from data. In many circumstances, it is difficult to obtain labeled data and in such cases where there is no labeled data, then the algorithms tries to identify similar patterns among inputs and then categorize then accordingly.

2.6 Related Work

The popularity of video streaming, availability of technological advanced devices and high performing networks have caused two fundamental shifts in end user behavior: more user sensitivity to video quality and expectations of higher peak bandwidth levels. End users get dissatisfied if a video re-buffers while they are watching it. This dissatisfaction further increases if there are several re-buffering events or if the re-buffer events are of longer duration and this can lead the end user to switch to another operator or to leave the service. This has increased the pressure on network operators and service providers to develop ways to monitor user perception to ensure quality of their networks and services.

QoE is one of the main factors in determining end user satisfaction. QoE of multimedia applications, including HTTP video streaming services have caused great interest in the research community. There is a significant amount of research on evaluation of video performance from the end user perspective on various access networks like Wi-Fi, 2G, Universal Mobile Telecommunications Services (UMTS) and 4G. This research is further subdivided into development of predictive models, measurement and analysis of QoE.

YouTube QoE is basically determined by stalling patterns. Analysis of user perception with respect to number of stalling events and their duration has been addressed in multiple works [12], [13], [14] and [15]. Authors in [12], [13], have studied the relationship between user perception and the number of stalling events, while authors in [14] have studied the relationship between user perception and the total stalling duration, and authors in [15] have discussed the effect of both the number of staling events and the duration of stalling on user perception. In studies conducted in [12], the authors have concluded that a single re-buffering event can cause reduction in user perception of about 1.5 in MOS scale. Similarly, authors in [13] conducted experiments and analyzed user perception at low, medium and high number of stalling events and concluded that when there are a low number of stalling events, MOS is 3.5 while at medium it drops to 2.7-2.0 and at a high number of stalling events, it further drops to 2.5-1.9 depending on the type of video. Further, authors in [14] have concluded that MOS decreases as the total stalling duration increases and MOS decreases when the initial loading time increases. Authors in [15] have concluded that have concluded that even one single stalling event can impact YouTube QoE, reducing the video quality with up to 1 MOS point (i.e., from good to fair). Moreover, a second stalling event has also a very deep influence on QoE and after it, saturation starts i.e., from 2 till 10 stallings there is a slight reduction in QoE from around 2.5 to 2. Authors in [15] also concluded that the duration of stalling events also have a strong influence in Youtube QoE but less effect as compared to the stalling events itself. Longer stalling duration rapidly decreases QoE. Network operators have gained interest in end user's opinion in terms of QoE and have realized the importance of evaluating video quality from users' prospective to ensure that end users are satisfied with the services the operators are providing.

Multiple studies have been conducted to determine QoE in video streaming services. The studies conducted in [13], [14], [15], [16], [17], [18], [19] consider an subjective approach. Authors in [15] conduct subjective tests through crowdsourcing while authors in [13], [14], [16], [17], [18], [19] use MOS to conduct subjective test. Further, studies conducted in [12], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31] consider an objective approach to determine QoE. On the other hand, authors in [32] have correlated both subjective and objective quality assessment methods. Peak Signal to Noise Ratio (PSNR) is one of the most widely used measure for determining the QoE to evaluate user perceved quality of video and is used in [20], [21], [32]. Other popular QoE metrics are Video Quality Metric (VQM) and Structural Similarity (SSIM), which are used in [32]. VQM is further analyzed in measurement studies conducted in [12], [13], [14], [15], [17], [20], [33], [34], [35], [36], [37], [38], [39]. In these studies, QoE is influenced by video quality metrics, for example, stalling patterns (i.e., number and duration of video stalling events) [15], rebuffereing events [13], [14], [33], [34], [38], [39], download time [17], video buffer status [12], [17], average bit rate [14], [17], startup delay [14], [20], [37], [39], file size [35], TCP connection [36] and available bandwidth [39]. ML has been applied in various studies to build QoE evaluation models for example, in studies conducted in [18], [28], [29], [30] [31] authors have developed an QoE evaluation model based on logistic function, Radial Basis Function Networks (RBFN), decision trees, Arificial Neural Network (ANN) with backpropagation algorithm and KNN respectively. The studies conducted in [12], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31] develop predictive models to estimate QoE for video streaming services. Further, authors in [40] have given a comprehensive review of evolution of QoE, based on video quality assessment methods.

Another approach is to monitor the Access Point (AP) queue as all Internet traffic passes through an AP. Video streaming applications resides on a UE and when a user wants to view a certain video, the request goes through an AP. AP queue capacity greatly influences packet loss, latency and the flow's achievable throughput and thus it is a critical factor in wireless Local Area Network (LAN) performance. Although with immense popularity of Wireless LAN (WLAN), queue capacities of an AP's are not known. It can vary among different vendors and different classes i.e., residential and commercial. Authors in [41] have concluded that AP queues are packet based and can contain between 50 to over 350 packets.

A further approach is to identify QoS metrics that effect QoE i.e., to establish a relationship between QoS and QoE. The study conducted in [16] catalogs QoS factors that affect QoE at each level of the protocol stack to form a conceptual relationship between QoS and QoE. Similarly, the studies conducted in [13], [15], [20], [32] investigate the relationship between QoS metrics and KPIs at different layers in the protocol stack with QoE. Authors in [13] have a two step approach to investigate the relationship between network layer QoS and QoE. In the first step they have studied the relationship between network layer and application layer QoS and in the second step they have evaluated the relationship between application layer

QoS and QoE. Similarly, authors in [15] have studied the relationship between network layer and application layer QoS with QoE and authors in [16] have studied the relationship between application level QoS and QoE. Further, authors in [32] have studied the relationship between network level QoS and QoE and authors in [20] have studied the relationship between network layer and transport layer QoS with QoE.

Network impairments influence video quality metrics that can cause rebuffereing events and lead to jerky playback as well as deformation of videos' temporal structure, which lead to degradation in QoE. Effects of network layer QoS on video QoE have been addressed in multiple works [12], [13], [15], [18], [19], [20], [21], [23], [24], [25], [26], [28], [29], [33], [42], [43]. In these studies, the effect of average packet loss, loss patterns, Round Trip Time (RTT) and delay variance on QoE are analyzed. Studies in [13], [15], [39] show how impairment of RTT can affect QoE of HTTP video streaming services. Authors in [12], [13], [19], [20], [21], [23], [28], [29], [33] have studied the impact of packet losses on streaming video performance. The significance of loss patterns on multimedia applications are discussed in [18], [24], [25], [42], [43]. Video sensitivity due to delay variation is discussed in [19]. Adittionally, authors in [26] use the frame loss pattern instead of the packet loss pattern to estimate QoE.

The studies conducted in [44], [45] describe network measurement methods. Authors in [44], have focused on network measurements and have discussed active, passive, hybrid measurement techniques, monitoring methods and given a comparison of both active and passive techniques. Active Measurements are further analysed and existing active measurement tools, network parameters and techniques are discussed [45]. Most of the resarchers have considered passive measurements as it is good for analysis. Studies conducted in [15], [33], [42], [43] have considered passive measurements. Authors in these studies have stored huge amount of data and then analyzed it. On the other hand active measurements are good for end to end measurements as active measurements consist of light weight probes that can measure round trip metrics and can give real time measurements. Studies conducted in [46], [47], [48] have used the active measurement protocol TWAMP to measure two way metrics. Authors in [46] have conducted measurements on Wi-Fi and 3G network of various network QoS with TWAMP like RTT, packet loss, duplicate packets, re-ordering, loss distance, loss period. Further, the study conducted in [47] compares four active end-to-end network measurement tools in an emulated environment and concluded that TWAMP can be a competitive alternative for measuring network performance metrics. Finally, in [48], authors have used TWAMP for 4G measurements in real time.

The study conducted in [49] gives a brief review of Machine Learning (ML) and its applications.

A subjective approach is time consuming while an objective approach captures user perception by conducting measurements at the UE. Objective methods give a very accurate measure of QoE but network operators cannot take congestion based decisions by these methods as reporting QoE value of each user back to them can consume significant bandwidth in terms of user scalability. For example, in [27], authors have proposed a way to monitor QoE in real time. They have extended the International Telecommunication Union (ITU) standardized multimedia quality assessment model G.1070 [50] that was originally designed as a video quality planning for video quality monitoring. But this monitoring application is implemented at the UE and there is no way to report monitoring results to the network operator.

Monitoring AP queues can be a very efficient way to monitor network congestion but due to uncertainty in AP queue sizes, between different vendors and classes, monitoring solutions at AP based on AP queues will be different for different AP's based on their queue sizes.

In various studies, researchers preferred to use simulators and emulators while performing their studies. For example in [13], [15], [18], [19], [20], [21], [23], [24], [25], [33], [47] authors have used network emulators to emulate network conditions to analyze the effect of various network parameters on QoE while in study conducted in [28], [42], authors have performed simulations to conduct their studies. Simulations can provide full control for researchers to investigate network behaviors, but not always reflect real-world scenarios especially wireless network behaviors due to the limitation of radio propagation models in simulators.

In order for network operators and service providers to predict subjective QoE, rather than to access physical properties of their access network (for example, bandwidth, delay, or loss rate), a QoE assessment approach is required, which can predict QoE by monitoring of objective network performance indicators.

3

Proposed Solution

QoE is subjective in nature. Therefore, it is difficult for network operators and service providers to predict QoE. There can be several factors that can affect QoE. For example, user equipment, human factors, network factors, type of encoding, type of decoding and usage context. QoE can be affected either by one of these factor or a combination of several factors. This chapter describes how an end user observes network congestion, the approach adapted in this thesis to effectively monitor a network based on QoE, video QoE metrics that determine end user QoE and software tools used in this master thesis.

3.1 Definition of Congestion

A network can be defined as *congested* for video applications if most of video end users in that network are experiencing video stalling, jerkiness and blurring of video. When the total bit rate is higher than the available download bandwidth then there is a high chance of video stalling, which will lead to degradation in user perception of service.

User perception is measured by QoE. There can be several issues that can affect QoE as shown in Section 2.2. The end user can experience degradation in QoE due to low throughput. This low throughput can be due to two reasons, congestion and the end user's location. In the former case, end users can perceive low QoE since many end users are connected to the same AP due to bandwidth sharing. Thereby each end user gets lower throughput than the desired bit rate, which leads to degradation of QoE. In the latter case, an end user can have low throughput due to low bandwidth allocation due to the wireless rate adaptation algorithm in IEEE 802.11 standard based on the location of the UE from the AP. Due to this, there are more chances that end users that are further away from the AP can perceive low QoE as less bandwidth is allocated to them due to their location. Further, if there are more end users then due to bandwidth sharing even less bandwidth will be allocated to each user. Thus, end users that are far away from the AP have a higher probability to have low QoE as their location will lead to congestion.

Video requires higher bit rates compared to other applications. These rates can vary depending on how the video is encoded. For instance, video applications distributed over the Internet range from 100 kbps for low-quality to over 3 Mbps for streaming High Definition (HD) videos, which lead to bit rates of more than 10 times higher than that of popular applications like Facebook[1] and music streaming applications [51].

HTTP streaming videos uses the TCP protocol, which always try to maximize the achievable throughput as allowed by the TCP congestion control and TCP flow-control mechanisms. Thus, leading to capacity-constrained networks. It is possible that video end users face low QoE while web users and users of low data rate applications are experiencing good services. If demands of resource exceed on capacity-constrained networks, the video end users will further have more degradation in QoE, due to increased contention as Wi-Fi has a shared-link. This will increase the queue at the AP, which will eventually fill up resulting in large number of packet losses. In this case, all video end users in the coverage area of the AP will experience degradation in QoE as YouTube QoE is highly sensitive to throughput bottlenecks.

In terms of service quality, video streaming services are more sensitive to quality issues as compared to services such as e-mail or FTP applications. For example, end users will not complain if they experience repeatedly interrupted downloads, but would become irritated as soon as a small amount of sound or video disturbance occurs during something being watched over the Internet [21].

This highlights the need for video quality monitoring method to effectively monitor end user satisfaction. This video quality monitoring method is to be designed based on the end user's perspective (QoE).

[1] `www.facebook.com`

3.2 Approach

In this Thesis, a fourfold approach will be adapted to effectively monitor a network based on QoE. Initially, network monitoring probes will be developed. Subsequently, a direct link will be established between subjective QoE and objective network measurements. Then, a classifier will be built, which translates network level measurements into QoE levels. Ultimately, the performance of the proposed method will be evaluated in different test cases.

First, network monitoring probes will be developed by identifying QoS parameters that affect QoE and developing a QoE assessment model in which QoE is a function of network level measurements, and implementing an active measurement protocol for network level measurements. A direct link will then be established between the subjective QoE and objective network measurements by designing various test cases and observing YouTube video manually during each network measurement, and recording the subjective QoE. Next, a ML approach will be used to build a classifier, which translates network level measurements into QoE levels. This will include data collection by performing network measurements in a Wi-Fi testbed for supervised learning and building a classification model using three ML algorithms: SVM, KNN and logistic regression. Finally, the performance of these classification models is evaluated using an evaluation metric called 'classification accuracy'. This thesis does not impose any particular ML algorithm.

3.3 Video QoE Metrics

Video QoE can be evaluated by different metrics as discussed in Section 2.6. In this thesis, YouTube QoE will be evaluated by stalling event patterns as experienced by the end user and two metrics, *stalling events* and *stalling duration* will be manually observed to determine the subjective QoE. Stalling events describe the number of stalling events and stalling duration describes the total duration of stalling. Impact of stalling events on QoE has been studied in [13], [17], [33], [52], [53] while impact of stalling duration on QoE has been studied in [14], [17], [54] and a combination of both has been studied in [12], [15].

3.4 Analysis Tools

Various support softwares are used within this Master thesis. Their specific roles are briefly described below

3.4.1 Octave/MATLAB

Octave[2]/MATLAB[3] is a high-level language primarily intended for numerical computations. In this Thesis, MATLAB was used used for graphic visualizations.

3.4.2 Scikit-learn

Scikit-learn[4] is an open source *Python* module integrating a wide range of stae-of-the-art machine learning algorithms for medium-scale supervised and unsupervised problems. In this Thesis, *scikit-learn* is used to implement selected supervised learning classification algorithms and for graphic visualization.

3.4.3 C

C[5] is general purpose computer programming language. In this Thesis, C is used to implement Network Application in Linux Ubuntu[6].

3.4.4 BASH

BASH is a UNIX shell and command language deployed under GNU open source licence. BASH has been distributed widely as a default shell on OS X[7] and major LINUX distributions. In this Thesis, BASH is used to post process data.

3.4.5 Iperf

Iperf[8] v2.0.5 is a tool to measure TCP and User Datagram Protocol (UDP) bandwidth performance. It can measure UDP characters like time transfer,

[2] `http://www.gnu.org/software/octave/`

[3] `http://se.mathworks.com/products/matlab/`

[4] `http://scikit-learn.org/stable/`

[5] Kernighan, Brain W.; Ritchie, Dennis M. (February 1978). The C Programming Language (1st ed.). Englewood Cliffs, NJ: Prentice Hall. ISBN 0-13-110163-3

[6] `http://www.ubuntu.com`

[7] `https://developer.apple.com/library/mac/documentation/Darwin/Reference/ManPages/man1/bash.1.html`

[8] `https://iperf.fr`

transfer rate , bandwidth, jitter and packet loss. In this Thesis, *iperf* is used to generate Constant Bit Rate (CBR) UDP cross traffic, contention traffic and to determine actual path capacity.

3.4.6 Wireshark

Wireshark[9] v1.12.3 is network protocol analyzer. Wireshark is used to check what is happening at a microscopic level at network. In this Thesis, *Wireshark* is used to test the Network application and estimate the bit rate of the YouTube video stream.

3.4.7 Iwconfig

Iwconfig[10]is a wireless tool for Linux that is dedicated to wireless interfaces. It is used to set the parameters of the network interface that are specific to the wireless operation. It is also used to display the parameters and wireless statistics. In this Thesis, iwconfig is used to view wireless parameters for determining the position to place wireless hosts in the network. It is used to check *Bit Rate* and *Recieved Signal Strength Indicator (RSSI)*. Based on these two things the wireless hosts are placed for each case as discussed in Section 5.2.

3.4.8 Wifi Analyser

Wifi Analyzer[11] is an Android application that turns an Android phone into a Wi-Fi analyzer. Wi-Fi channels can be seen and the tool helps to find a less crowded channel. Also, the signal strength can be seen using this application. In this thesis, *WiFi Analyser* is used to check neighboring channels and check the RSSI.

[9] `https://www.wireshark.org/`

[10] `http://linux.die.net/man/8/iwconfig`

[11] `https://play.google.com/store/apps/details?id=com.farproc.wifi.analyzer&hl=sv`

4

QoE Assessment Method

IN ORDER TO predict QoE, this study proposes a QoE assessment method. The proposed method is based on ML, which allows network operators and service providers to predict QoE from network level measurements. This chapter describes the proposed QoE assessment method and the steps leading to development of it.

4.1 Proposed QoE Assessment Method

There are two QoE assessment methods (discussed in Section 2.2). The most widely used method for subjective assessment is *user subjective tests*, which are conducted in a test environment, where test environment parameters used in the experiments can be controlled and the effect of each parameters on user experience can be assessed and modeled as a function of each parameter. However, temporal aspects of user experience cannot be assessed in subjective assessment tests as these tests are planned to be of short duration so that participants do not get bored during the tasks. Subjective measurements are expensive and time-consuming tasks as discussed in Section 2.2.1.

Due to this *objective assessments* will be used in this thesis. Objective assessments are implemented in two phases as discussed in Section 2.2.2. Figure 4.1 shows the objective assessment model.

The first step of an objective assessment is to select a computational model that can monitor QoE in real time. There is a lot work done on quality estimation and predicting of video streaming applications but still there is

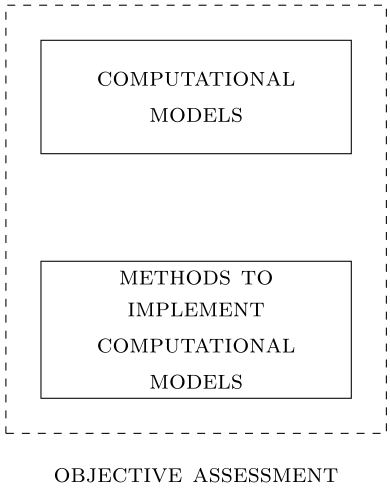


Figure 4.1: Objective QoE Assessment

no final standardized model available for streaming video applications. The current objective computational methods cannot be implemented because these methods are not practical for network monitoring. For example, Full reference models require a source video and as the network operator often receives an encoded video from the content provider, the source video wil be unavailable, so full reference models cannot be applied. For example, the video quality estimation model of International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) G.1070 [50], which is a computational model for video-telephony applications, uses encoded bit rate and frame rate of the compressed video along with the expected packet loss rate of the channel to predict the subjective video quality. It can be very difficult to have reliable values of certain channel parameters (e.g., expected packet loss in this case). As IP is traditionally designed for best effort services, due to congestion in links, traffic can be routed through another path and traffic from a large number of applications passes through a single network, which can cause additional tail-dropping in router queues due to which it is difficult to predict or estimate a realistic packet loss. Also, challenges posed by the wireless nature of communication links will add more difficulty in order to estimate realistic packet loss. Reduced reference models can be applied at UE but an end user cannot obtain partial information about the source video. Therefore, reduced reference models cannot be applied for QoE monitoring. No reference models are useful for monitoring QoE at the UE.

Implementation at the UE can have certain drawbacks. For example, the application will use the processing power of the UE causing overhead at the UE. Also, transmitting the results of the monitoring back to the network operator will create extra traffic depending on the number of UEs in the network which will consume the available bandwidth. Further, scalability can be an issue (scalability, both in terms of number of users and applications as each user can use several applications and if each application is doing its QoE assessment separately this will create a lot of additional traffic and overhead at UE). However, by implementing these models the QoE monitoring will be very accurate.

Due to these reasons, a more simplified QoE assessment model along with an efficient method to implement that assessment model are needed to be designed in order to monitor network congestion in real-time.

4.2 QoE Assessment Model

QoS [55] is usually described in terms of the KPIs such as latency, jitter, throughput and packet loss and ITU-T Recommendation G.1010 [56] introduces delay, jitter and packet loss as QoS parameters that can impact QoE. So, by improving these KPIs, QoE can be improved.

In this Thesis, a simplified QoE assessment model will replace QoE computational models used for objective assessment in Figure 4.1. This model will be designed by identifying the KPIs from NPMs (as described in Section 2.3), which effect end users' perceived performance taking into account the measurements of those KPIs in real-time in order to make a quality assessment. KPIs that effect the user's perceived performance are discussed as below.

4.2.1 Packet Loss

Packet loss is the dominating parameter that affects video quality as discussed in [12], [13], [19], [20], [21], [23], [33], [57]. Shared environments such as WLAN are error prone. Typical communications over WLAN involves a high BER due to the dynamic nature of wireless communication links and packet loss either happens due to collision or due to a weak signal. Wireless packet losses are often location and time varying.

Even with a high BER, from the test bed in [20] it is observed that video QoE remains stable in WLAN even with high packet loss because Wi-Fi communication is limited by interference and shows more stable behavior as compared to 3GPP. The Majority of these losses in video may be recovered during TCP fast re-transmission, even for a high loss rate and there are less Retransmission Time Outs (RTOs) in Wi-Fi, and therefore, Wi-Fi can maintain the throughput.

In [58], a singleton metric is defined for *round-trip packet loss*. The metric defines an IP network's ability to transfer packets in both direction from one host to another host. Two way communication is always needed, thus failure to transfer a packet in either direction results in a round-trip packet loss. The metric is described as a boolean. It is one if there is a packet loss and zero if there is a successful packet transmission.

4.2.2 Loss Patterns

Loss pattern or loss distribution [59] is a key parameter that determines the performance observed by users. Loss pattern is an important factor when measuring performance of both real-time and non real-time applications. Internet exhibits bursty packet loss that can effect QoE. For the same loss rate, two different loss distributions could potentially produce widely different perceptions of performance [42]. The significance of loss patterns on multimedia applications are discussed in [18], [24], [25], [42], [43]. In [59], two derived metrics are defined which are as follows.

4.2.2.1 Loss Period

Loss period captures the frequency and the *burstiness* (length) of loss once it occurs. The Loss period allows the study of loss burstiness for each occurrence of loss. A single loss period of length 'x' can account for a significant portion of the overall loss rate.

4.2.2.2 Loss Distance

Loss distance captures the separation between the loss periods or packet losses. The Loss distance is useful in determining the *spread factor* associated with the packet loss rate. The greater the value of loss distance the greater the spread factor.

A wireless link involves unpredictable burst errors, which can be uncorrelated with instantaneous available bandwidth. Burst errors can happen due to an UE being in a fading dip and due to time varying characteristics of the wireless channel.

In this Thesis, loss distance equal to one is considered as important, since it signifies consecutive packet losses. The distribution of loss distance is measured and considered as an important KPI. By this, bursty packet losses in video are expected to be measured.

4.2.3 Inter Packet Delay Variance (IPDV)

Inter Packet Delay Variance (IPDV) [60] describes the difference in the one-way delay of two selected packets within a stream. The IPDV is important for measuring the performance of a network. An increase in delay variation can indicate presence of congestion in the network i.e., the maximum delay variation within a high percentile can be of interest. This increase in delay variation can be an indication of an increasing congestion and if higher values of IPDV persists, the network can lead to congestion. In case for media streams that require delivery of packets at regular intervals, regular measurements of IPDV can be very useful in determining buffer sizes. Video sensitivity due to delay variation is discussed in [19].

IPDV measurement is performed by sending a stream of equally sized packets between the measuring node and then the IPDV is measured between selected packets. All packets are timestamped at source and destination hosts, an the one-way delay is computed. The one-way delays of the two packets are then subtracted, and the IPDV can be measured. If one or both of the chosen packet are lost or do not arrive within the specified time, the IPDV is undefined.

In this Thesis, the sum of IPDV is measured i.e., the sum of the difference in one-way delay from the source host to the destination host and the difference in one-way delay from the destination host to the source host.

4.2.4 Round Trip Time (RTT)

RTT [61] is the time it takes for a packet to be transmitted from the source host to its destination, and then back again. At transmission, the source host timetamps the packet. When the destination host receives the packet, it immediately responds and sends another packet back to the source host. When this response packet is received, the source host records received timestamp. The value of RTT is then computed by taking the difference in timestamps. If the packet is lost, RTT is undefined (as RTT is computed by first comparing the sequence number and if the sequence number matches, then computing the difference between the two timestamps). No distinction is made between in what direction the packet was lost. There is no need of synchronization between two host as this measurement is performed at the source host.

Studies in [13], [15], [39] shows how impairment of RTT can affect the QoE of HTTP video streaming services.

In this Thesis, the proposed QoE assessment model consists of QoE as a function of packet loss, loss period, loss distance, RTT and IPDV.

4.3 Method of Implementation of QoE Assessment Model

The next step is the implementation of QoE assessment model. Initially, it is to be decided where the measurements can be taken. In order to make this QoE assessment model scalable and obtain measurements in real-time, network-level measurement can be a good option. Followed by, implementation of monitoring method and an ML approach to predict QoE.

4.3.1 Network-based methods for QoE Assessment

As discussed in Section 3.1, all the users will suffer from degradation in QoE either due to contention in the medium or due to overflow of AP queues. Also, in order to deal with the user scalability issue in the network, a monitoring method is needed that can monitor QoE in real-time and that is not affected by the number of end users in the access network. This can be achieved by making network-level measurements. By network-level measurements, the access network can be monitored as shown in Figure 4.2, irrespective of the number of users in the network. This implementation can later be useful also for congestion based decision making by network operators.

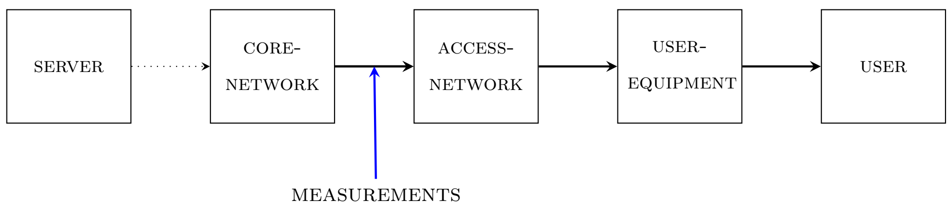


Figure 4.2: Network-level measurements for assessing QoE

Therefore, there is a need for a method that is capable of network-level measurements to extract user perceived performance indicators and then compute the QoE based on these extracted user perceived performance indicators in real-time.

4.3.2 Monitoring Method

Active and passive measurements (discussed in Section 2.3.2) produce different kinds of information depending on the measurements and the results do not necessarily correlate. Both methods have advantages and are recommended in certain kind of measurements.

In this Thesis, active measurements are chosen to extract user perceived performance indicators on the network level.

Active measurements inject additional traffic in the network, which can consume available bandwidth of the network that could have been used for end-user applications. Therefore, an effective measurement method based on active measurements has been designed, which takes into account

1. Real-time extraction of user perceived indicators.
2. Consumption of minimum available bandwidth.
3. The method should be able to retrieve all perceived indicators required by the assessment model.
4. The method should be able to communicate the results of the assessment to network operators so that some action can be taken to manage the congestion.

Active measurements consist of tools and protocols. Each existing measurement tool is often dedicated to the measurement of a single parameter. Therefore, it is difficult to measure several parameters with a single tool. IPPM is working on specifications of active measurement protocols as defined in Section 2.3.3.

By using TWAMP, user perceived indicators can be extracted in real-time. In other studies, authors have used different tools to measure different parameters but TWAMP can measure several parameters and can also communicate the results back to network operators for further assessment. Therefore, TWAMP is selected in this thesis as the method to implement the assessment model defined in Section 4.2.1. The QoE assessment used in this thesis can be seen in Figure 4.3.

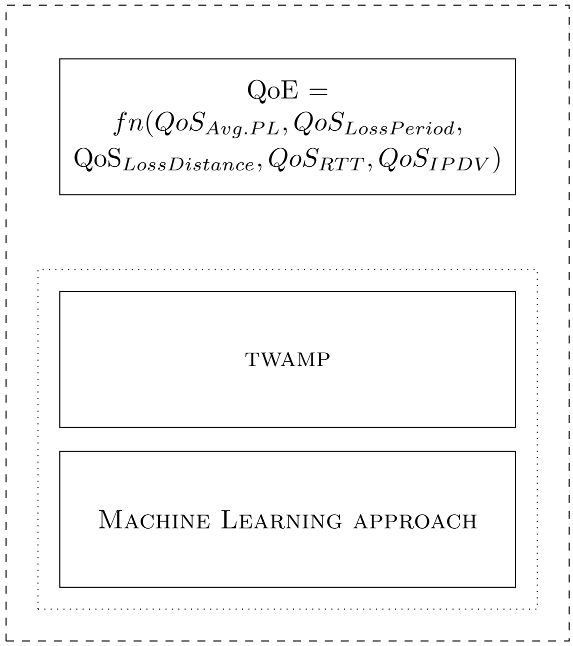


Figure 4.3: Objective QoE Assessment used in this Thesis

4.3.2.1 TWAMP Implementation

TWAMP-Light is a simpler implementation of TWAMP having simpler architecture and is implemented in this thesis to monitor network congestion. It is implemented as a software tool, which measures certain KPIs (eg. packet loss, RTT, etc.) and on basis of these parameters, TWAMP can indicate if there is a degradation in QoE.

4.3.2.1.1 TWAMP-Light

TWAMP light is implemented in a two-host scenario, where the client side is known as *Controller* and the server side is known as *Responder*. *Session-Sender* and *Control-Client* are implemented at the controller while *Session-Reflector* is implemented at the responder. *Controller* establishes a test-session with *Responder* by sending probe packets. *Responder's* role is simply to listen for the incoming probes on a specified port and reflect them back to *Controller* upon arrival of each probe while copying necessary information and generating sequence number and time-stamp values as in [7]. *Controller* receives the packets reflected back by *Responder* and extracts round-trip metrics.

However, the logical role of the Server is omitted, as well as the TWAMP-control protocol are omitted. The TWAMP-Light implementation can be seen in Figure 4.4.

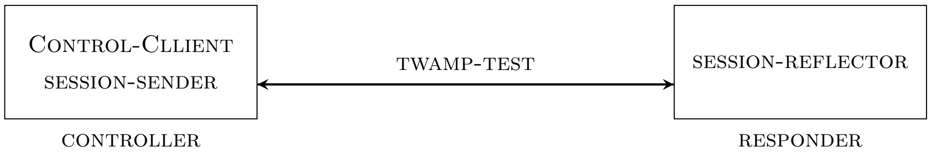


Figure 4.4: TWAMP-Light Implementation

The implemented TWAMP-Light components are described below:

4.3.2.1.1.1 Controller

Controller is a network application implemented in C and targeted towards the Linux platform installed on a host. Controller sends a UDP test-session of probe packets of fixed size 1500 Bytes to mimic video packets to Responder on UDP port 25000. After the packets are reflected by Responder, Controller receives the reflected packets and Control-client computes two way metrics and process them.

4.3.2.1.1.2 Responder

Responder is also a network application implemented in C in the Linux platform installed on a laptop. Responder is always running and waiting for test packets from Controller. It reflects packets as soon they arrive as by the packet format defined in [7].

4.3.3 Machine Learning Approach

In this Section, an overview of a proposed machine learning approach for congestion detection is described and then each step of the approach is discussed in detail.

4.3.3.1 Overview

The key idea proposed in this thesis is to use an ML approach to classify the state of the network i.e., weather the network is congested or not. In this approach, KPI's i.e., packet loss, consecutive probe packet losses (loss distance=1), loss period, RTT and IPDV that are extracted during network measurement (discussed in Section 4.2) act as *features*. The subjective QoE is manually recorded for each network measurement and Youtube QoE metrics (discussed in Section 3.3) are recorded and each measurement is then labeled.

In a nutshell, the proposed approach consists of five (5) major steps. The first step consists of labeling each measurement during data collection. The second step involves preprocessing data i.e., feature scaling. The third step consists of feature selection. The fourth step consists of model building from the labeled training data and selected features from the feature selection step. The fifth step consists of evaluating the performance of model. Two different approaches will be used to evaluate the performance of the model. First, the stability of the classifier will be accessed by performing *cross validation* and then, the model's performance (i.e., *classification accuracy*) will be evaluated using an independent hold-out set.

4.3.3.2 Labeling Training Set

In this Thesis, an approach is adopted to manually observe and record the two YouTube QoE metrics i.e., stalling events and stalling duration (discussed in Section 3.1) for each measurement during data collection (discussed in the next chapter in Section 5.2) as *congested* and *not congested*. The label variable, congested, is then encoded to take value 0 when QoE is good i.e., the network is not congested and 1 when QoE is low i.e., the network is congested. As the label variables are categorical, *supervised learning* (discussed in Section 2.5.1) is used. Further, as the label variables are discrete, *classification* (discussed in Section 2.5.1.1) methods are used.

4.3.3.3 Feature Scaling or Normalization

Feature scaling is also known as data normalization and is a common requirement for many machine learning algorithms. Feature scaling is performed at the data prepossessing step and is used to standardize the range of features in such a way that the resulting normalized data is better suited for classification. Some ML algorithms may not work properly if the individual features do not more or less look like normally distributed data i.e., *Gaussian with zero mean and unit variance*, so in this case it becomes essential to perform feature scaling on data during the classification model building in order to extract reliable classification.

In this Thesis, since the values in raw dataset varies a lot, a *Standard Scalar* normalization technique is used to scale all features. This technique standardizes features by removing the mean and scaling to unit variance.

Centering and scaling occur independently on each feature by computing the relevant statistics on each of the samples in the training set.

4.3.3.4 Feature Selection

Feature selection is also known as attribute selection. Not all features are important and it is not always beneficial to use all the features of a dataset because not all features contribute to the outcome and in that case using all features can result in decreasing the accuracy of the models. Feature selection is a process that selects those features in a dataset that contributes most to the outcome. Feature selection can be defined as a process that chooses a minimum subset of M features from the original set of N features so that the feature space is optimally reduced according to a certain evaluation criterion. The resulting feature space contains the most important features that contributes in predicting the outcome. The benefits of feature selection are threefold:

- *Reducing over-fitting* as after feature selection there are only those features left that contributes most to the outcome. Therefore, the remaining features provide better understanding of the underlying process that generates the outcome. Feature selection leads to less chances to make decisions based on noise. Thus, creating accurate predictive models.
- *Improves accuracy* i.e., prediction performance, since having the features that contribute more leads to having less misleading data.
- *Reduces training time* as less data after feature selection lead to faster training time and achieve cost effective classifiers.

Recursive Feature Elimination (RFE) is a popular feature selection method. RFE works by recursively removing attributes and building a model on those attributes that remain. RFE uses model accuracy to identify the attributes (and combination of attributes) that contribute the most to predicting the label attribute. In this Thesis, RFE with cross validation is used as a feature selection method that performs RFE in a cross-validation loop to find the optimal number of features.

4.3.3.5 Training, Testing and Cross Validation

In order to test and evaluate the classifiers, *k-fold stratified cross validation* is adapted in this thesis. The data is splitted into k subsets of equal sizes and in each iteration, one of k subset is used as test set while other *k-1* subsets are used as training set. This process is repeated for k iterations and the mean statistic of k iterations is calculated. In this thesis, k=*10* is used to compute testing accuracy in order to measure the effectiveness of the classifier. 10-fold cross validation limits problems like over-fitting and gives estimate of out of sample accuracy i.e., it will give an insight on how the classifier will generalize to out of sample data.

4.3.3.6 Evaluation Metric

In order to evaluate and compare different classifiers, *classification accuracy* is considered as an evaluation metric. Classification accuracy indicates the accuracy of each classifier. Classification accuracy is determined after building the classification model and by testing the classification model with the *hold-out* set and checking the classification accuracy. Equation 4.1 defines the fraction of correct predictions over n_s.

$$\text{classification accuracy}(y,\hat{y}) = \frac{1}{n_s}\sum_{i=0}^{n_s-1} I(\hat{y}_i = y_i) \tag{4.1}$$

where, $\hat{y}_i$ is the predicted value of i^{th} sample, y_i is the corresponding true value and $I(x)$ is the Indicator function[1] such that

$$I(x) = \begin{cases} 1 & \text{when } \hat{y}_i = y_i \\ 0 & \text{when } \hat{y}_i \neq y_i \end{cases}$$

A good classifier is defined as where the classification accuracy (determined by eq. 4.1) is the greatest.

4.3.3.7 Classifiers Evaluated

In this Thesis, three different classifiers are proposed to be evaluated on the dataset. These algorithms consist of both linear and non-linear classifiers. These classifiers are:

1. Support Vector Machine (SVM)
2. K-Nearest Neighbor (KNN)
3. Logistic Regression

The idea in this work is to apply an ML approach to detect network congestion i.e., low QoE. The features for training the network monitoring system

[1] The indicator function of an event is a random variable that takes value 1 when the event happens and value 0 when the event doesn't happen.

system were selected in a way to offer a more in-depth description of the characteristics of the network.

A supervised ML approach is used to train the network monitoring system in order to classify user experience. A set of ML algorithms: SVM, KNN and Logistic Regression were tested and evaluated to build a classification model in order to be used in the network monitoring system module within the network management system. Section 6.2.3 shows how these classifiers perform on the data set. Further, this thesis doesn't impose any particular ML algorithm. The three classifiers are described below.

4.3.3.7.1 Support Vector Machine (SVM)
Support Vector Machine (SVM) [62] is a powerful learning and pattern recognization [63] technique for data classification. The basic idea behind SVM is to use training data to create an optimal classification line i.e., *hyperplane*, which classifies two classes. First, the data points in each class that lie closest to hyperplane are identified. These data points are called *support vectors*. Then, an optimal hyperplane is computed using these support vectors. An SVM classifier can be written using linear equation

$$y = \boldsymbol{w} \cdot \boldsymbol{x} + b \tag{4.2}$$

where, $\boldsymbol{w}$ is the weight vector (i.e., the normal vector of the hyperplane), $\boldsymbol{x}$ is the input vector and b is the contribution from the bias weight. The hyperplane is at $y = 0$, and the support vectors are at $y = \pm 1$. The smallest distance between the decision boundary and any of the support vectors is called margin (m), which is defined as

$$m = \frac{2}{\|w\|} \tag{4.3}$$

where, $\|w\|$ is the norm (or length) of $\boldsymbol{w}$.

SVM minimizes the miss-classification probability by maximizing the *margin* around the separating hyperplane. This can be done by minimizing $\|w\|$, with the condition that there should be no data points within margin i.e.,

$$\boldsymbol{x_i} \cdot \boldsymbol{w} + b \geq 1 \text{ when } y_i = 1 \tag{4.4}$$

$$\boldsymbol{x_i} \cdot \boldsymbol{w} + b \leq 1 \text{ when } y_i = 1 \tag{4.5}$$

Equation 4.4 and 4.5 can be combined to

$$y_i(\boldsymbol{x_i} \cdot \boldsymbol{w} + b) \geq 1 \tag{4.6}$$

The problem is to minimize $\|w\|$, such that the discrimination boundary is obeyed i.e., *min* $f(x)$ such that $g(x) = 0$. This is a *constrained optimization problem* and can be expressed mathematically as:

$$\min f : \frac{1}{2}\|w\|^2 \text{ subject to g: } y_i(\boldsymbol{x_i} \cdot \boldsymbol{w} - b \geq 1) \tag{4.7}$$

where, $\boldsymbol{x}_i$ is the *ith* training sample and y_i is the correct output of SVM for that sample. For positive samples in class, the value of y_i is $+1$ and for negatives samples, the value is -1. Also, $f(x)$ is a *quadratic function.* This is an example of a *quadratic programming problem* in which the aim is to minimize a quadratic function subject to a set of linear inequality constraints. This quadratic programming problem can be solved with a *Lagrangian Multiplier* method. Another modification to the problem is to change the original optimization problem into a *maximization problem* by finding its *dual fuction* that produces a version of the problem that is more efficient for *quadratic optimization* and thus, leading to Equation 4.8. Dual functions are constructed using *Karush-Kuhn-Tucker construction* instead of Lagrange in case of non-linear behavior. The above mentioned optimization techniques are described in textbooks on optimization such as [64].

$$L(\alpha) = \max \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x_i}, \boldsymbol{x_j}) \tag{4.8}$$

subject to the constraints,

$$\alpha_i \geq 0, \forall i$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

where, $K(\boldsymbol{x_i}, \boldsymbol{x_j})$ is a kernel function.

Data may not always be linearly separable. Kernels provide transformation of data by mapping data into higher dimensions and finding a linear decision boundary that separates the data into classes. In this thesis, four (4) different kernels are tested, with two (2) implementations of linear models and two (2) non-linear kernel implementations to check the evaluation metric.

- SVC with linear kernel
- LinearSVC (Linear kernel)
- SVC with RBF kernel

- SVC with polynomial (degree 3) kernel

Equations 4.9-4.11 define decision functions of these kernels.

$$sgn\Big(\sum_{i=1}^{n} y_i\alpha_i K(\boldsymbol{x_i}, \boldsymbol{x_j}) + \rho\Big) \tag{4.9}$$

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = exp(-\gamma \parallel x_i - x_j \parallel^2) \tag{4.10}$$

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = (x_i \cdot x_j + 1)^s \tag{4.11}$$

where, Equation 4.9 describes the decision function of *Support Vector Classification (SVC) with linear kernel. LinearSVC* is another implementation of SVC for the case of a linear kernel. Equation 4.10 describes the decision function of *SVC with Radial Basis Function (RBF) kernel*, where γ defines how much influence a single training sample has and $\gamma > 0$. Equation 4.11 describes the decision function of *SVC with polynomial kernel*, where s specifies the *degree* of the polynomial.

4.3.3.7.2 K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) assumes that all instances correspond to points in the n-dimensional space $\Re^n$, where the distance between instances is in terms of Euclidean distance. For example, let an arbitrary instance 'X' be defined by the feature vector

$$\{x_1, x_2, ..., x_n\}$$

where $x_i, x_j, ..., x_n$ are corresponding features. The Euclidean distance between two instances X_i and X_j is defined by $d(X_i,X_j)$, where

$$d(X_i,X_j) = \sqrt{\sum_{k=l}^{n}(x_k^i - x_k^j)^2}$$

where, x_k^j represents the value of the k^{th} feature of instance X_i. In KNN, the discrete valued label function $f : \Re^n \longrightarrow L$, where L is defined as

$$L = \{l_j \mid 1 \leq j \leq m,\ \text{m is postive integer}\}$$

In KNN, training algorithms train each training sample $(x,f(x))$ and add them to a list **training_examples** and then the classification algorithm performs classification by finding K instances from the training_examples with the smallest Euclidean distance with query point x_q and then taking the most popular value. This is shown in Equation 4.12 by $\hat{f}(x_q)$

$$\hat{f}(X_q) \longleftarrow \underset{t \in T}{\operatorname{argmax}} \sum_{i=1}^{k} \delta(t, f(X_i)) \tag{4.12}$$

where,

$$\delta(a, b) = \begin{cases} 1 & \text{when } a = b \\ 0 & \text{when } a \neq b \end{cases}$$

There is another variant of KNN, called *weighted nearest neighbor algorithm* that eliminates the impact of noise and outliers in the sample space. The algorithm gives weights to neighbors according to the distance and thereby gives larger weight to nearest neighbor. This is shown in Equation 4.13

$$\hat{f}(X_q) \longleftarrow \underset{t \in T}{\operatorname{argmax}} \sum_{i=1}^{k} w_i \delta(t, f(X_i)) \tag{4.13}$$

where,

$$w_i = \frac{1}{d(X_q, X_i)^2}$$

denotes the distance weight of the nearest neighbor X_i.

4.3.3.7.3 Logistic Regression

Logistic regression is also known as logit regression in literature and is a linear model for classification. Logistic regression uses a logistic function or logit function (shown in Equation 4.14) to model the probabilities describing the possible outcomes of a single trail (input) based on one or more features. Logistic regression measures the relationship between categorical labels and one or more features by estimating probabilities using a logit function, which is a cumulative logistic distribution.

$$f(x) = \frac{L}{1 + e^{-k(x - x_0)}} \tag{4.14}$$

The model relates the probability of event i.e., (p_i) to the features $x_{1,i}, x_{2,i}, ..., x_{k,i}$ through:

$$logit(p_i) = log_e\left(\frac{p_i}{1 - p_i}\right) = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + ... + \beta_k xk,i \tag{4.15}$$

where,

$$p = P(Y_i = 1)$$

Solving Equation 4.15 for p_i,

$$p_i = \frac{e^{\beta_0+\beta_1 x_{1,i}+\beta_2 x_{2,i}+\ldots+\beta_k xk,i}}{1+e^{\beta_0+\beta_1 x_{1,i}+\beta_2 x_{2,i}+\ldots+\beta_k xk,i}} \tag{4.16}$$

In Equation 4.16, the parameters α and β are usually estimated using the Maximum Likelihood method. Description of the Maximum Likelihood method can be found in [65].

5

Data Collection

THIS CHAPTER DESCRIBES the methodology that has been used for data collection and development of experimental cases. Initially, a direct link is established between subjective QoE and objective network measurements by performing network measurements on a Wi-Fi testbed to study the impact of wireless rate adaptation and link utilization on QoE by loading WLAN with cross traffic on downlink or bi-directional paths along with YouTube video. Followed by, designing various test cases to check the performance of QoE assessment model.

5.1 Experimental Testbed

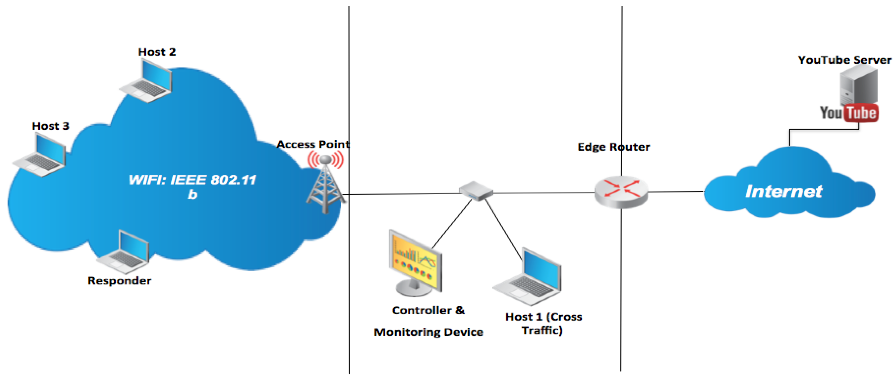


Figure 5.1: Testbed

The testbed is assembled as in Figure 5.1 to monitor and estimate congestion in the access network (WLAN). The Network Application is implemented in Linux and is evaluated by varying network conditions in an IEEE 802.11 b WLAN testbed. This testbed consists of *Controller*, *Host 1*, *Wireless router* and three other hosts i.e., *Responder*, *Host 2* and *Host 3*. Controller and Host 1 are connected to the wireless router by 100 Mbps LAN. Controller (Network Application Client) performs estimation and Host 1 serves as a cross traffic generator for all measurements. Host 1 also serves as a contending traffic receiver for scenario 2 (as described in Section 5.2) measurements. Responder, Host 2 and Host 3 are connected to the wireless router by IEEE 802.11 b WLAN using Intel Corporation PRO/Wireless 5100 AGN [Shiloh] Network Connection[1] with Intel corporation Mobile 4 series chipset[2], Qualcomm Atheros AR9285 Wireless Network Adapter (PCI-Express)[3] with Intel Corporation 5 Series/3400 Series chipset[4] and Intel Corporation PRO/Wireless 3945ABG [Golan] Network connection[5] respectively. Responder (Network Application Server) reflects incoming packets back to Controller as soon as it receives them. Host 2 serves cross traffic server for all measurements and it also serves as contending traffic generator for scenario 2 measurements. Host 3 serves as YouTube UE. All the testbed laptops run Ubuntu 14.04 LTS[6] with Linux kernel version 3.13.0-34. Table 5.1 shows the specification of equipment used in the test bed.

In order to minimize interference, an Android application *Wifi Analyzer*[8] is used to check the Wi-Fi channels used by neighbouring Wi-Fi networks and then, accordingly channel 13 is selected for this testbed as no other neighbouring network are using that channel.

Experiments are conducted in an indoor environment at Bredbandsgruppens forskningslaboratorium, Elektro och Informationsteknik (EIT), Lunds Tekniska Högskola (LTH).

[1] `http://www.intel.com/content/www/us/en/wireless-products/wifi-link-5100-brief.html`

[2] `http://www.intel.com/Assets/PDF/datasheet/320122.pdf`

[3] `http://www.qca.qualcomm.com/wp-content/uploads/2013/11/AR9285.pdf`

[4] `http://www.intel.com/content/www/us/en/chipsets/5-chipset-3400-chipset-datasheet.html`

[5] `http://www.intel.com/products/wireless/prowireless_mobile.htm`

[6] `http://www.ubuntu.com`

[7] `http://laureldsl.net/pdf/D-Link%202640B%20UserManual.pdf`

[8] `https://play.google.com/store/apps/details?id=com.farproc.wifi.analyzer&hl=sv`

TABLE 5.1
SPECIFICATION OF THE EQUIPMENT USED IN THE TESTBED

Device	Specification
Wireless Router	D-Link DSL-2640B[7], IEEE 802.11 b
Controller	Levano ThinkPad Laptop, AMD Dual-Core Processor, 4 GB RAM
Responder	HP-Compaq Laptop, Intel Core 2 Duo Processor, 2 GB RAM
Host 1	Desktop Computer, Intel Pantium 4 CPU 2.4 GHz, 1 GB RAM
Host 2	Sony-Vaio Laptop, Intel Core i3 Processor, 4 GB RAM
Host 3	HP-Compaq Laptop, Intel Core 2 Duo Processor, 2 GB RAM

5.2 Experimental Cases

A YouTube video titled *UNSOLVED MYSTERIES: The Secret of Easter Island*[9] of 480p quality is used to test the network application.

Two test scenerios are designed that are as follows

1. YouTube video and single CBR UDP cross traffic sharing the wireless downlink channel.

2. YouTube video and bi-directional CBR UDP cross traffic (single flow in each direction).

In each of these test scenarios, measurements are designed to study the impact of wireless rate adaptation and link utilization on QoE. To study the impact of rate adaptation, measurements were conducted at three bandwidth levels: 2, 5.5 and 11 Mbps, depending on the distance of the wireless devices from the AP and in each bandwidth level, measurements were designed to study the impact of link utilization on QoE. There are no experimental measurements designed for bandwidth level of 1 Mbps as there will not be sufficient available bandwidth for 480p video at bandwidth level of 1 Mbps and wireless link will be congested.

[9]https://www.youtube.com/watch?v=mH0sIjAHBVY

Table 5.2 shows the eleven measurements. Measurements 1-3 include base configurations, in which no cross traffic and contending traffic is introduced for each data rate.

Measurements 4-6 include a variety of cross traffic rates, modeling applications with download traffic, but there is no stalling event for each data rate.

Measurements 7-9 consist of measurements corresponding to a congested network for each data rate that include a variety of cross traffic rates in order to make the wireless link congested, modeling applications with significant download traffic.

Measurements 10 and 11 consist of measurements corresponding to a congested network for data rates 11 Mbps and 5.5 Mbps respectively with both cross and contending traffic modeling users applications with both upload and download traffic and more than one user using Wi-Fi.

Table 5.2
Measurements

Measurement. #	Data Rate (Mbps)	Cross Traffic	Contending Traffic
1	11	None	None
2	5.5	None	None
3	2	None	None
4	11	Yes	None
5	5.5	Yes	None
6	2	Yes	None
7	11	Yes	None
8	5.5	Yes	None
9	2	Yes	None
10	11	Yes	Yes
11	5.5	Yes	Yes

The wireless radio channel consists of fluctuations in the link throughput and sometimes an error-prone communication environment due to characteristics such as multipath propagation, interference from other sources, decrease signal strength, shadowing, etc. For each bandwidth level the RSSI range is fixed. The actual path capacity is measured using *Iperf* by measuring UDP throughput of the path and checking the packet loss at a present UDP injection rate and then repeating this process until a suitable injection rate is found that allows the highest rate at which the packet loss is negligible. This is done by injecting a single saturated CBR UDP flow with Maximum Transmission Unit (MTU) of 1460 Bytes for 300 seconds. The cross traffic rate is then determined based on actual path capacity and the bit rate of the video stream. If cross traffic is sent at actual path capacity there might be no degradation on QoE due to TCP congestion control.

For each measurement 4-9, cross traffic rate is determined by injecting a single saturated CBR UDP flow with MTU of 1460 Bytes. After 15 seconds, letting the flow to stabilize, the video is played and observed.

For measurements 4-6, the highest injection rate is determined at which there is no stalling event.

For measurements 7-9, a similar procedure is repeated, but here the lowest injection rate is determined at which video starts to experience stalling events.

For measurements 10 and 11, the contention traffic is kept fixed at 2 Mbps and 1 Mbps respectively while the cross traffic rate is then determined similar to measurements 7-9. There is no experimental case for 2 Mbps for this category as observed from measurements 6, the wireless link easily gets congested with very little cross traffic.

Cross traffic rates may vary a bit. Cross traffic for each measurements 4-11 are determining my running the above mentioned tests for ten times to determine the cross traffic range. Table 5.3 shows the RSSI range, actual path capacity and cross traffic range for each scenario.

Table 5.3
RSSI & Cross Traffic Rates Ranges

Data Rate (Mbps)	RSSI Range (dBm)	Actual Path Capacity (Mbps)	Cross-Traffic Rate (Mbps)		
			Measurements 4-6	Measurements 7-9	Measurements 10-11
11	< -79	3.4	3.9 - 4.1	4.60 - 4.75	3.0 - 4.0
5.5	-79 - -84	2.2	1.80 - 1.90	2.1 - 2.2	1.35 - 1.6
2	-84 - -90	1.0	0.16 - 2.0	0.38 - 0.45	-

Each evaluation consists of loading the network with traffic depending on the measurement number (as specified in Table 5.2) for 15 seconds and letting the system stabilize and then playing the above mentioned YouTube video. The network application starts 10 seconds after the video. Cross and Contending traffic are generated with *Iperf* by a single CBR UDP flow with packet size of the MTU of 1460 Bytes with the determined rate.

Each of the eleven measurements were repeated 39 times and desired metrics were extracted. The sample size is 429. When experiment population size is not definite, sample sizes greater than 380 can be considered relaible for any further analysis [66].

Furthermore, all measurements were conducted after midnight till early morning when most wireless networks in the building were assumed to be in idle state.

From these measurements, five cases were designed, which are shown in Table 5.4.

Table 5.4
Evaluation cases for experiments

Case. #	Scenario. #	Max Data Rate (Mbps)
1	1	11.0
2	1	5.5
3	1	2.0
4	2	11.0
5	2	5.5

6

Experimental Results and Analysis

THIS CHAPTER FOCUSES on building classification models and performance evaluation of proposed QoE assessment model. Also, this chapter illustrates the relationship of the selected KPIs i.e., average packet loss, loss period, loss distance, RTT and IPDV with subjective QoE.

6.1 Effect of KPIs on QoE

The effect of selected KPIs i.e., average packet loss, loss period, loss distance, IPDV and RTT (discussed in Section 4.2) on YouTube QoE (discussed in 3.3) are visualized in this section. It is observed in all designed measurements corresponding to congested network (measurement 7-11), an increase in all these KPIs, results in an increase in YouTube QoE metrics i.e., *number of stalling events* and *total stalling duration*, resulting in a decrease in user QoE. As an example, visualizations from measurement 8 are shown in Figure 6.1-6.6, where relationship between two variables can be seen by Least square line[1]. It can be seen from Figure 6.1 that there is direct relation between YouTube QoE metrics i.e., number of stalling events and total stalling duration. Figure 6.2-6.6 shows effect of KPIs on YouTube QoE metric 'total stalling duration'. It can be seen from these figures that an increase in all these KPIs results in an increase in YouTube QoE metrics. Thus, resulting in a decrease in user QoE. As Figure 6.1 showed direct re-

[1] http://mathworld.wolfram.com/LeastSquaresFitting.html

lation between the two YouTube QoE metrics, the effect of KPIs on stalling events will be similar of that of total stalling duration.

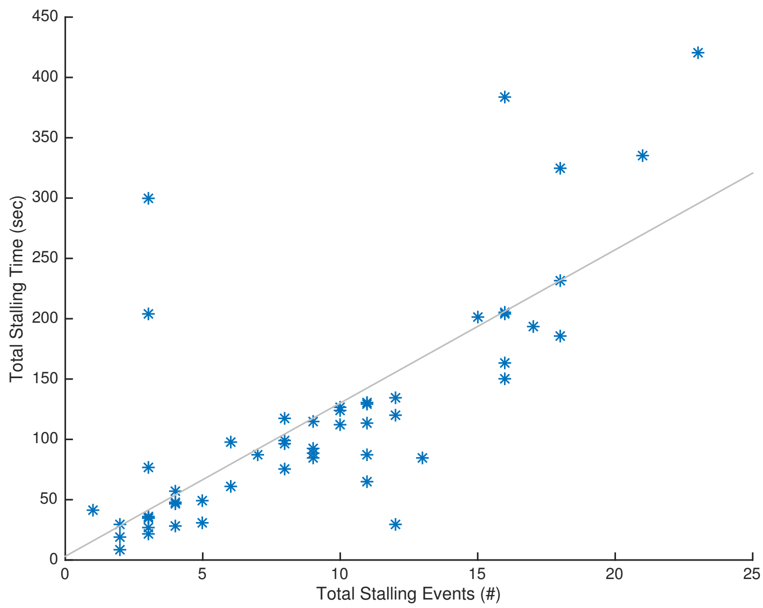


Figure 6.1: Relationship between Stalling Duration and Stalling Events

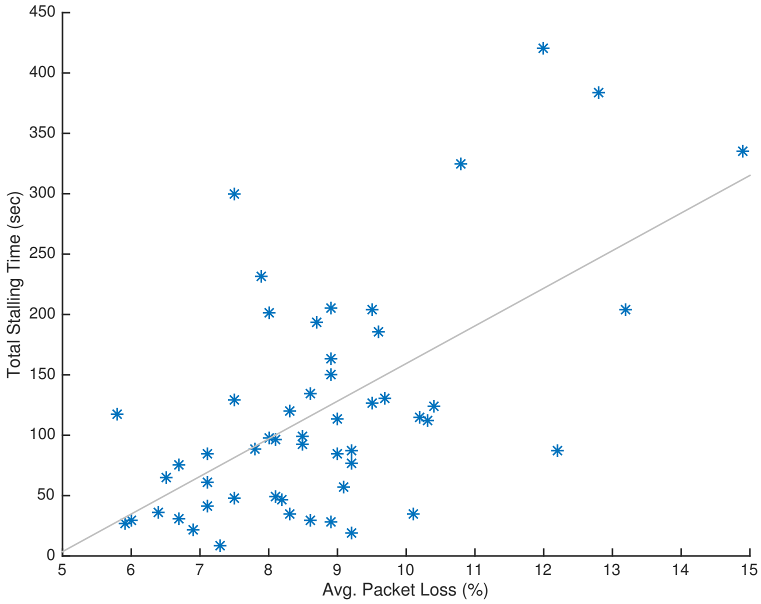


Figure 6.2: Effect of Avg. Packet Loss on QoE

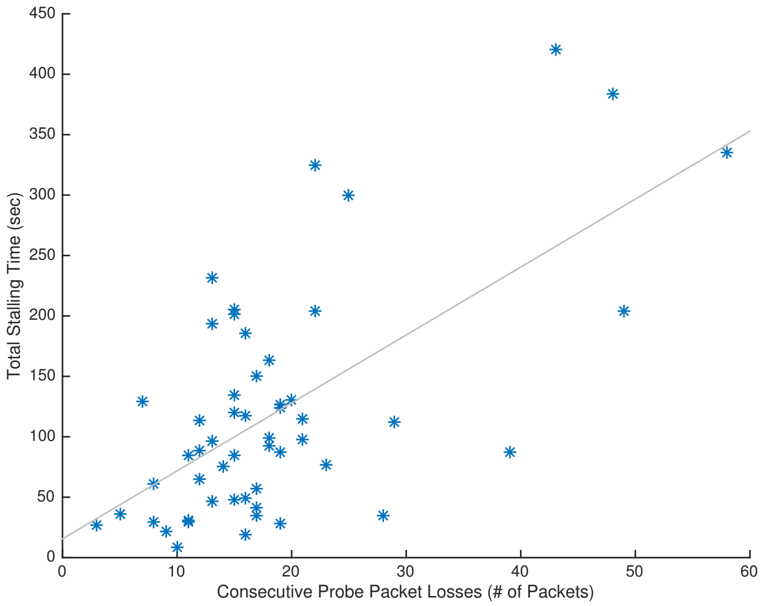


Figure 6.3: Effect of Consecutive Packet Losses on QoE

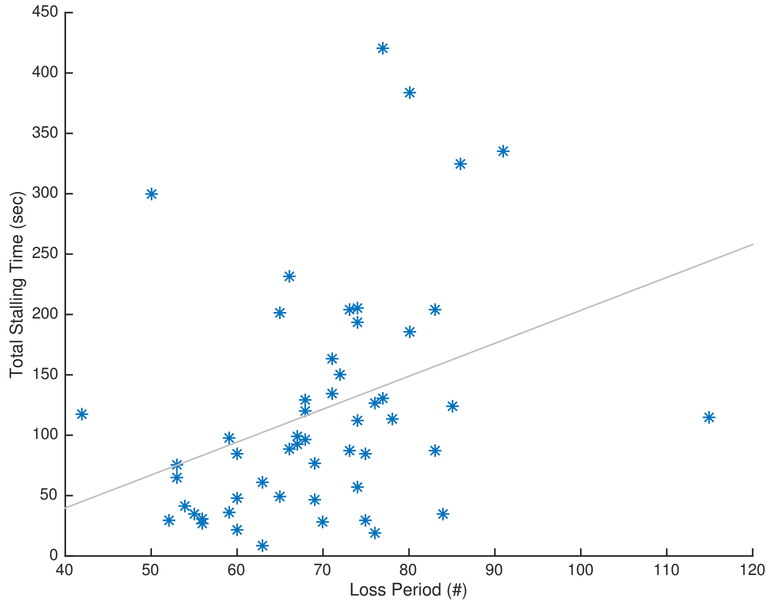


Figure 6.4: Effect of Loss Period on QoE

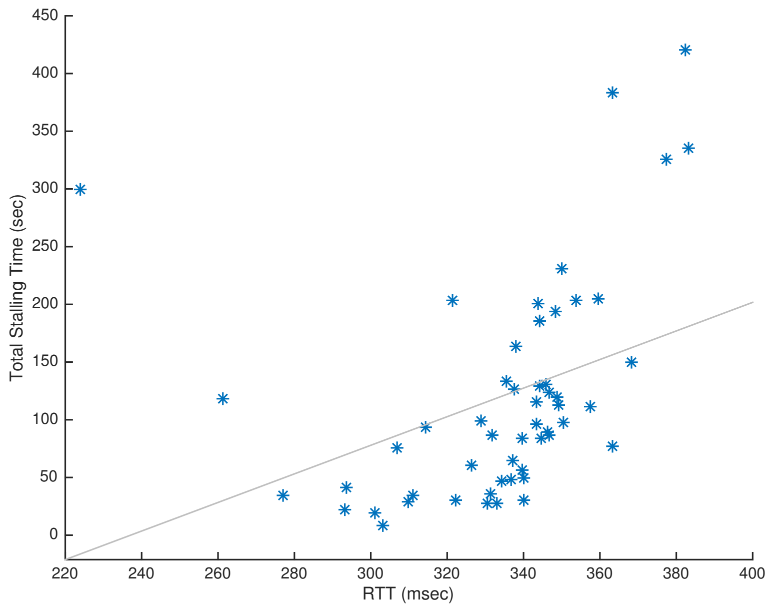


Figure 6.5: Effect of RTT on QoE

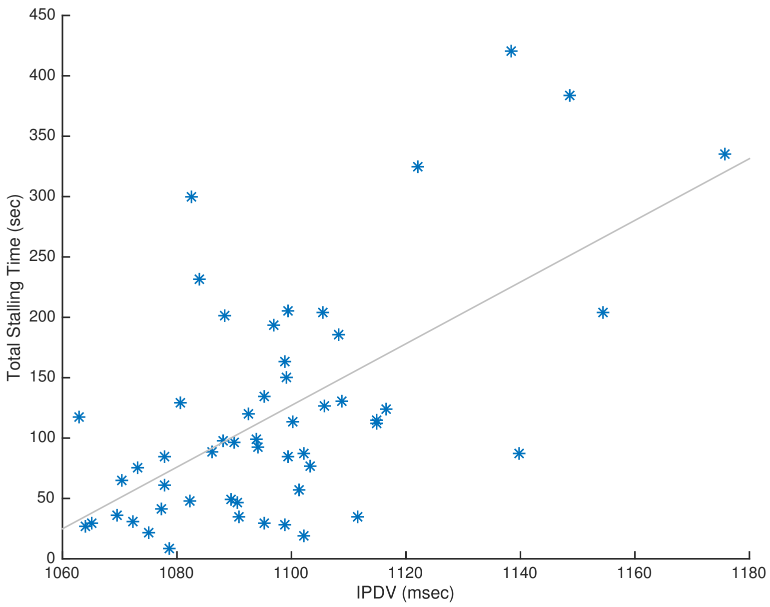


Figure 6.6: Effect of IPDV on QoE

6.2 Building a Classification Model

The dataset was divided into two parts. 10% of data was sampled out randomly from the data set and a *hold-out* set was created. The remaining portion of data is used for model building. Three different ML algorithms: SVM, KNN and logistic regression were tested and evaluated to build a classification model in order to be used in the network monitoring system module within the network management system. The classification model was then evaluated using a hold-out set.

As hold-out data is separated from the dataset before model building, the data of the hold-out set is out-of-sample data for the built classification model. Thus, the accuracy from validating the model with the hold-out set gives a more reliable estimate of out of sample performance.

The steps involved in building a classification model from each classifier are described in the following subsections. For model building and analysis, Case 3 is chosen as an example for illustrations throughout this chapter.

6.2.1 Support Vector Machine (SVM)

In each case (1-5), 10-fold cross validation was applied to the remaining dataset and four different kernels functions: SVC, Linear SVC, RBF and polynomial (degree 3) were tested. Figure 6.7-6.9 shows one fold of cross validation for these four kernels with 2D plots, where the relationship between features are shown as an example. It can be seen in Figure 6.7, that SVC with polynomial (degree 3) kernel has two miss-classifications while the other three kernels have none. Similarly, in Figure 6.8, SVC with RBF kernel has one while SVC with polynomial (degree 3) kernel have 2 miss-classifications while both linear kernels have none. Also, it is observed that both linear kernels have minimum miss-classifications as compared to the RBF and polynomial (degree 3) kernels. In order to choose the best kernel, cross validation accuracy of each of these kernels were determined for all cases as shown in Table 6.1 and the kernel with the highest accuracy was chosen. It can be seen from Table 6.1 that SVC with linear kernel and Linear SVC have the same accuracy. Therefore, *SVC with Linear Kernel* was selected for model building.

A 10-fold cross validation was then applied to the remaining dataset, which can be seen in Figure 6.10 and cross validation accuracy was determined that is an estimate of the out of sample accuracy. In the end, the model was evaluated using a hold-out set, which can be seen in Figure 6.11 and detailed

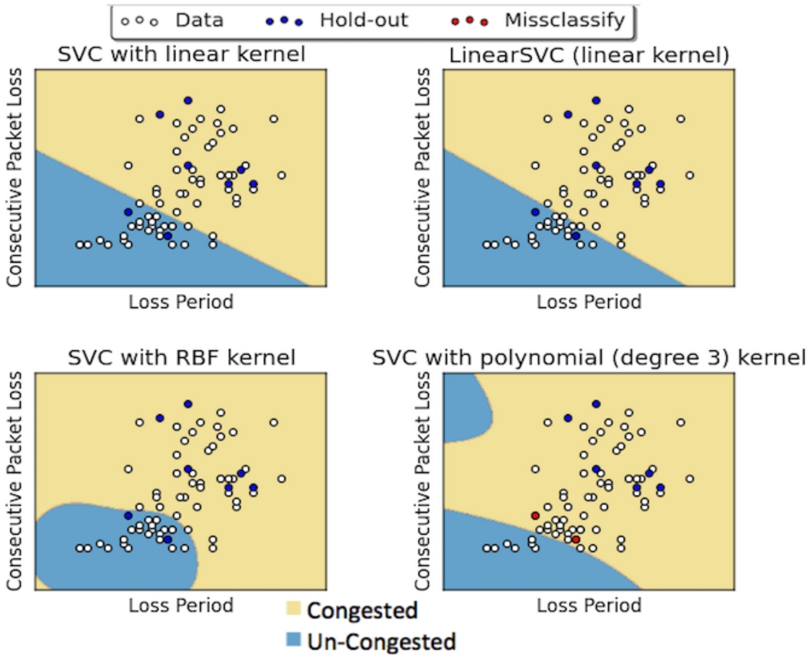


Figure 6.7: SVM: Comparision between Kernels functions; Loss Period vs Consec. PL

2D figure with decision boundaries showing the relationship between two features, as an example, can be seen from Figure 6.12. Cross validation accuracy and final testing accuracy can be seen in Table 6.2.

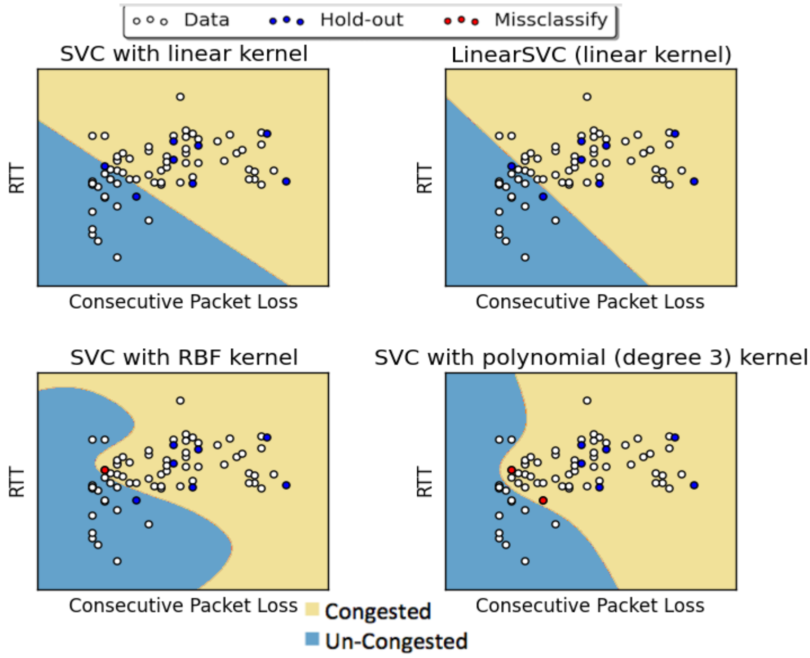


Figure 6.8: SVM: Comparision between Kernels functions; Consecutive PL vs RTT

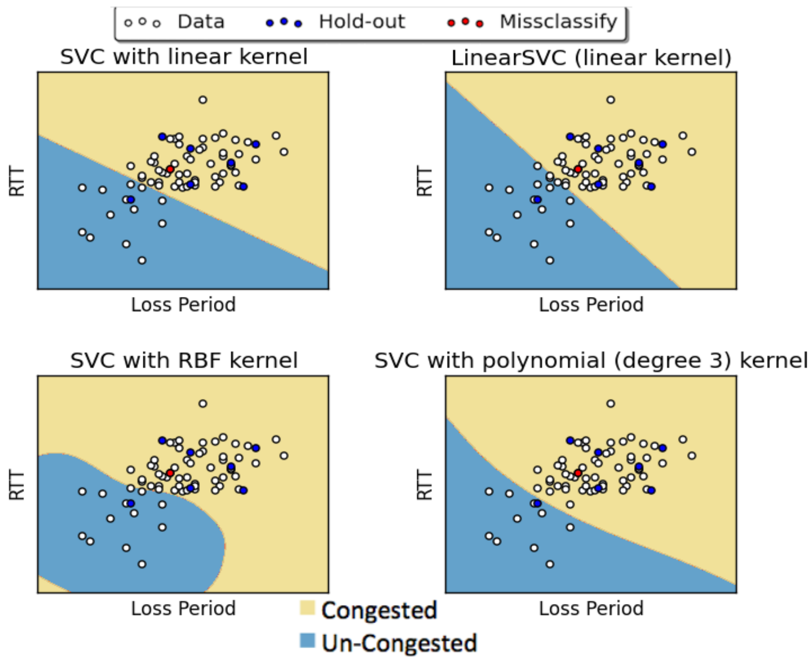


Figure 6.9: SVM: Comparision between Kernels functions; Loss Period vs RTT

Table 6.1
SVM: Choosing Kernel

Case. #	Kernel Function	Cross Validation Accuracy Score
1	SVC with Linear Kernel	0.949603174603
	SVC with RBF Kernel	0.938492063492
	SVC with polynomial (degree 3) Kernel	0.867658730159
	LinearSVC (linear kernel)	0.949603174603
2	SVC with Linear Kernel	0.93
	SVC with RBF Kernel	0.913333333334
	SVC with polynomial (degree 3) Kernel	0.963333333334
	LinearSVC (linear kernel)	0.93
3	SVC with Linear Kernel	0.8
	SVC with RBF Kernel	0.796428571429
	SVC with polynomial (degree 3) Kernel	0.7875
	LinearSVC (linear kernel)	0.814285714286
4	SVC with Linear Kernel	1.0
	SVC with RBF Kernel	1.0
	SVC with polynomial (degree 3) Kernel	0.965277777778
	LinearSVC (linear kernel)	1.0
5	SVC with Linear Kernel	1.0
	SVC with RBF Kernel	1.0
	SVC with polynomial (degree 3) Kernel	1.0
	LinearSVC (linear kernel)	1.0

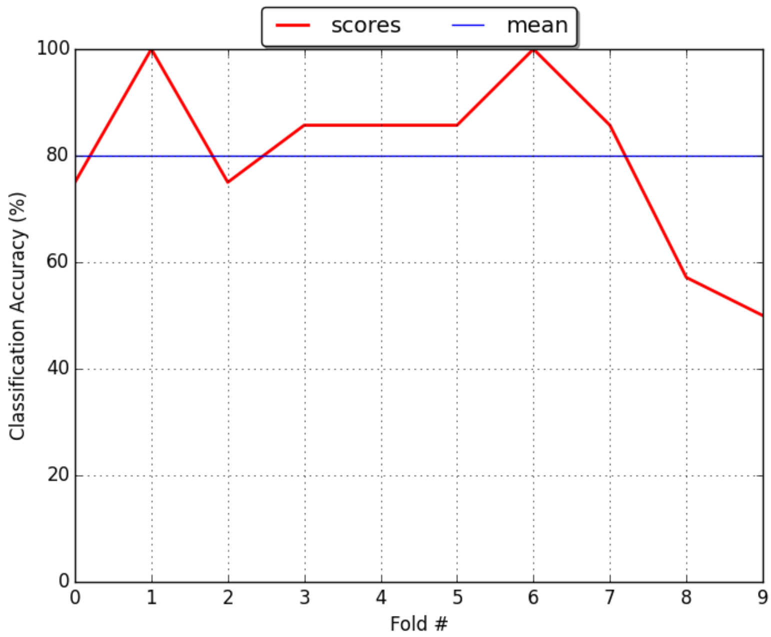


Figure 6.10: SVM: Cross Validation Accuracy

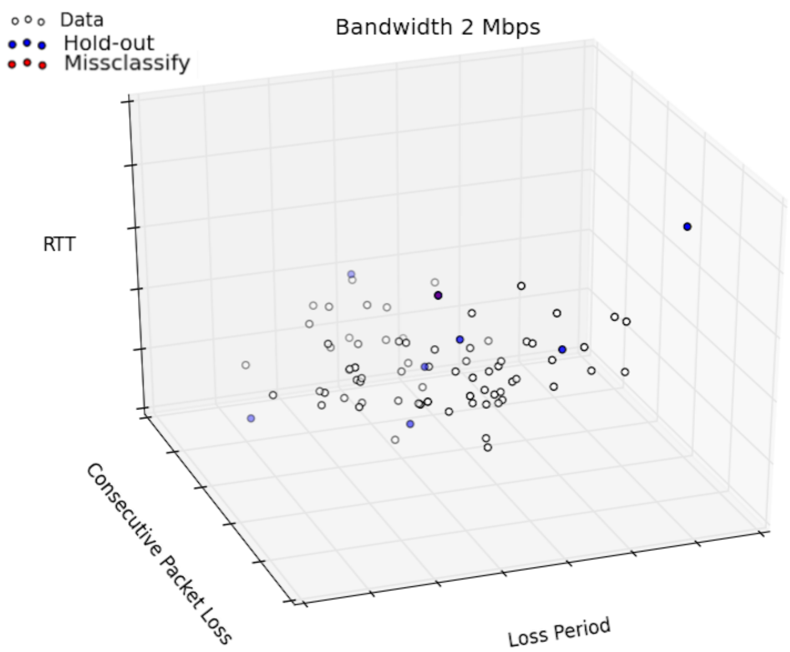


Figure 6.11: SVM: 3D-Model testing with hold-out dataset

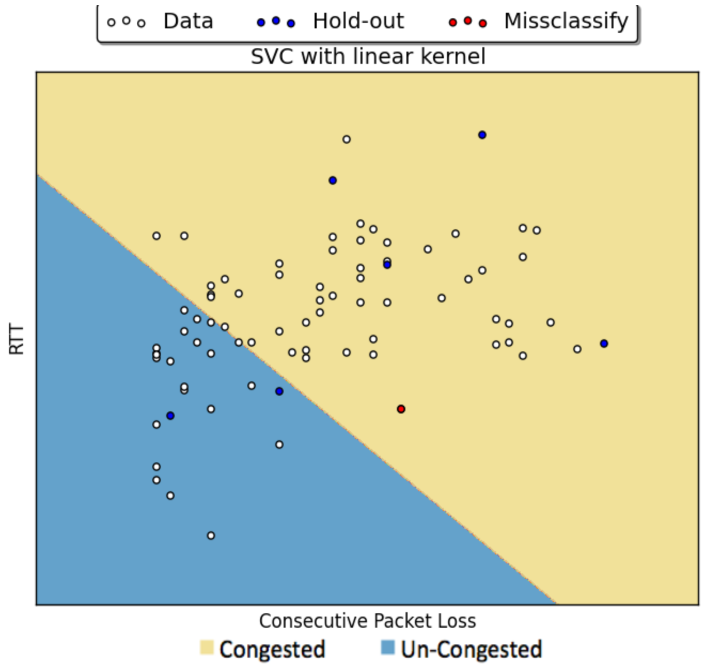


Figure 6.12: SVM: 2D-Model testing with hold-out dataset; Consecutive PL vs RTT

Table 6.2
SVM: Accuracy Scores

Case.	Cross Validation Accuracy Score	Testing Accuracy
1	0.949603174603	1.0
2	0.93	0.833333333333
3	0.8	0.875
4	1.0	1.0
5	1.0	1.0

6.2.2 K-Nearest Neighbor (KNN)

In each case (1-5), 10-fold cross validation was applied to the remaining dataset. This dataset was tested with both *uniform weights* and *weighted neighbors*, and after checking the 2D feature relationship between uniform weights and weighted neighbors, it was determined that weighted neighbors will miss-classify more as compared to uniform weights. For example, Figure 6.13 shows one set of features in one fold plot in which it can be seen that when there are uniform weights in Figure 6.13(a), there is one miss-classification while in case of weighted neighbors in Figure 6.13(b), there are two miss-classifications. In order to choose whether to use uniform weights or weighted neighbours, cross validation accuracy of each is determined for all cases as shown in Table 6.3 and the method with the highest accuracy was chosen. It can be seen from Table 6.3 that uniform weights give better cross validation accuracy so *uniform weighs* are further used in model building.

TABLE 6.3
KNN: COMPARISON BETWEEN UNIFORM AND WEIGHTED NEIGHBORS

Case.	Weights	Cross Validation Accuracy Score
1	Uniform weights	0.962103174603
	Weighted neighbors	0.949603174603
2	Uniform weights	0.963333333333
	Weighted neighbors	0.963333333333
3	Uniform weights	0.828571428571
	Weighted neighbors	0.8125
4	Uniform weights	1.0
	Weighted neighbors	1.0
5	Uniform weights	1.0
	Weighted neighbors	1.0

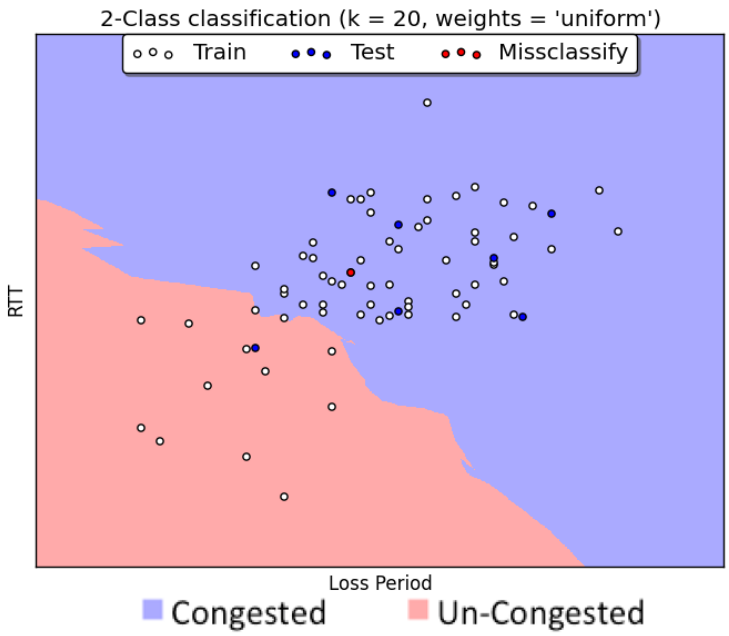
During model building, cross validation accuracy was checked for different values of K from 1-30 for each case and then, whichever value of K that gave the highest cross validation accuracy, that value of K was selected for model building for that case. For example, cross validation or case 3 is shown in Figure 6.14. The higher the value of K, the lower the complexity of model. The selected values of K for each case can be seen in Table 6.4. A 10-fold cross validation was then applied to the remaining dataset, which can be seen in Figure 6.15 and cross validation accuracy was determined, which is an estimate of out of sample accuracy. In the end, the model was evaluated using a hold-out set, which can be seen in Figure 6.16 and detailed 2D figure, with decision boundaries showing relationship between two features, as an example, can be seen in Figure 6.17. Cross validation accuracy and final testing accuracy can be seen in Table 6.5.

Table 6.4
KNN: Values of K for each Case

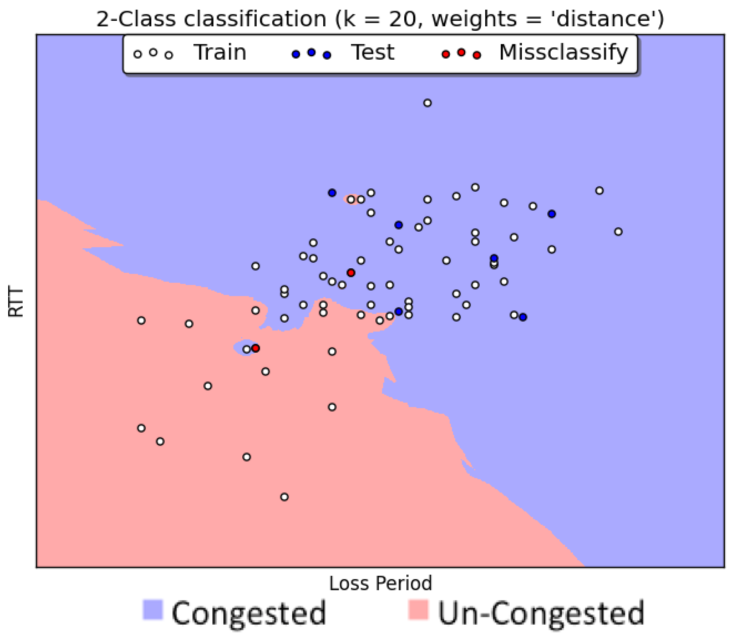
Case.	Value of K
1	13
2	4
3	20
4	20
5	8

Table 6.5
KNN: Accuracy Scores

Case.	Cross Validation Accuracy Score	Testing Accuracy
1	0.962103174603	1.0
2	0.963333333333	1.0
3	0.828571428571	0.875
4	1.0	1.0
5	1.0	1.0



(a) Loss Period vs RTT



(b) Loss Period vs RTT

Figure 6.13: KNN: Comparison between Uniform weights and weighted neighbors

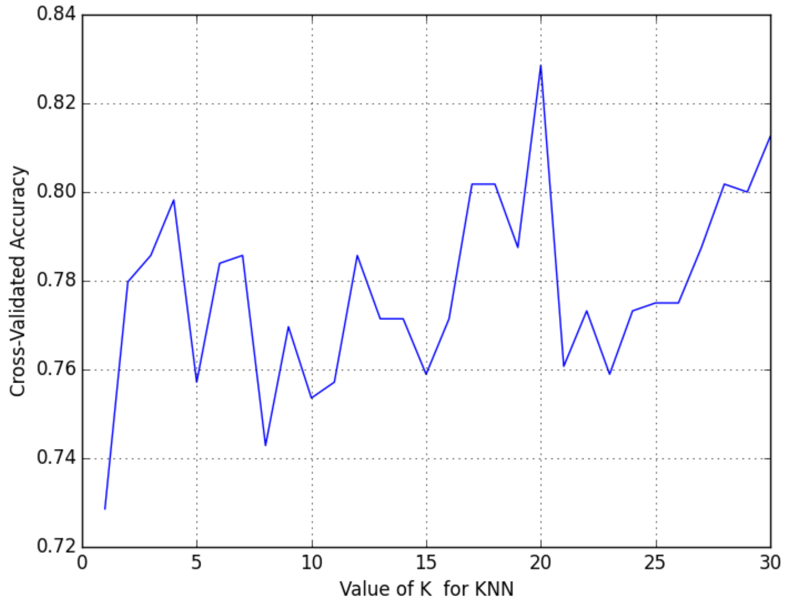


Figure 6.14: KNN: Value of K

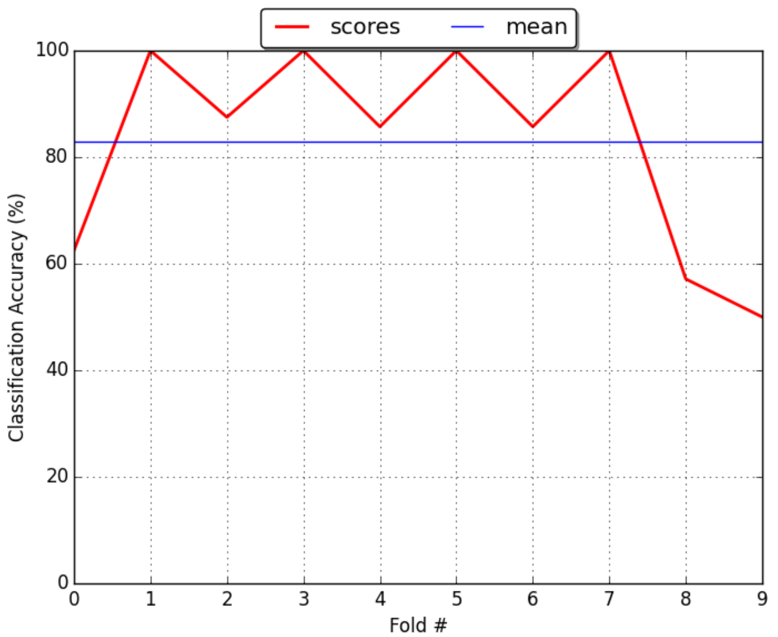


Figure 6.15: KNN: Cross Validation Accuracy

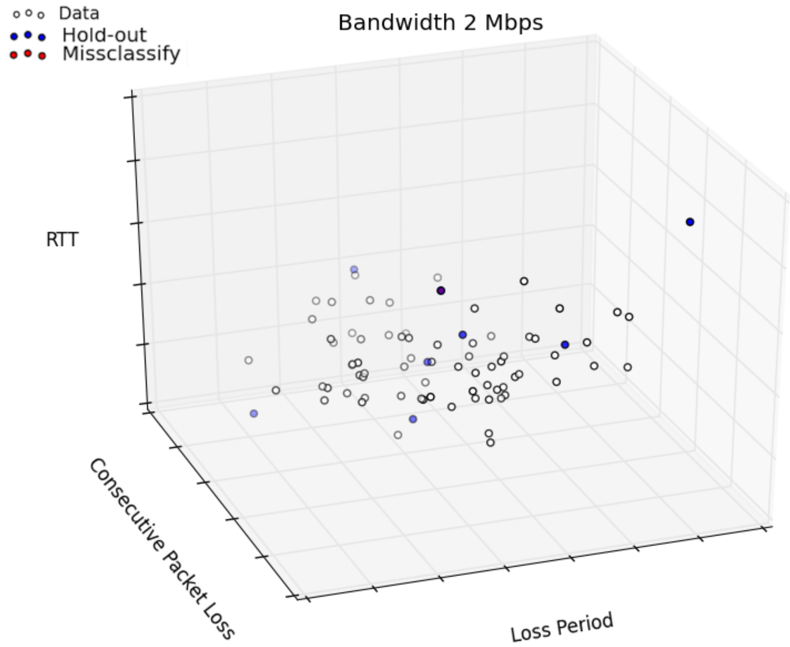


Figure 6.16: KNN: 3D-Model testing with hold-out dataset

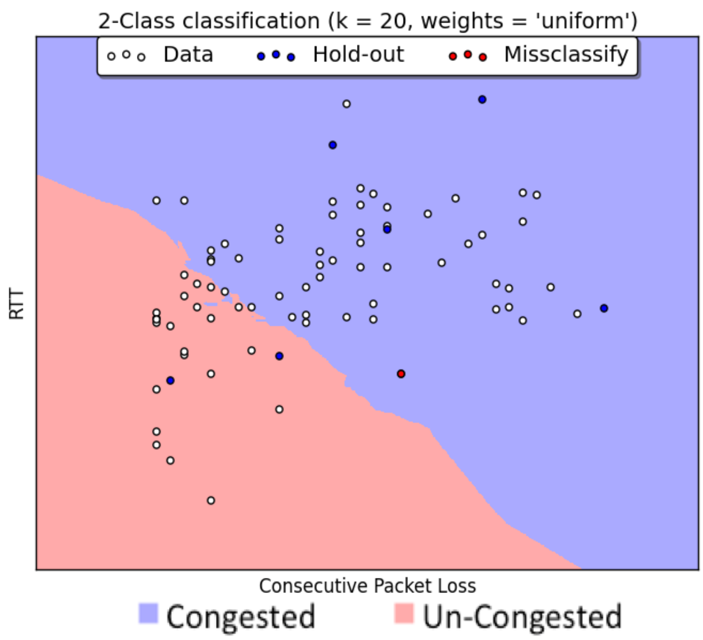


Figure 6.17: KNN: 2D-Model testing with hold-out dataset; Consecutive PL vs RTT

6.2.3 Logistic Regression

In each case (1-5), 10-fold cross validation was applied to the remaining dataset. Cross validation for case 3 can be seen in Figure 6.18 and cross validation accuracy was determined which is an estimate of out of sample accuracy. In the end, the model was evaluated using a hold-out set, which can be seen in Figure 6.19 and detailed 2D figure, with decision boundaries showing the relationship between two features, as an example, can be seen from Figure 6.20. Cross validation accuracy and final testing accuracy can be seen in Table 6.6.

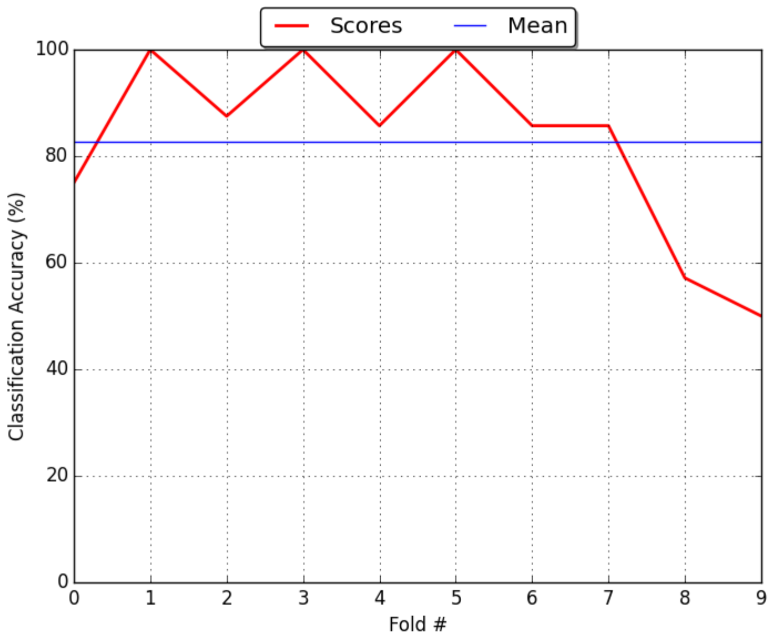


Figure 6.18: Logistic Regression: Cross Validation Accuracy

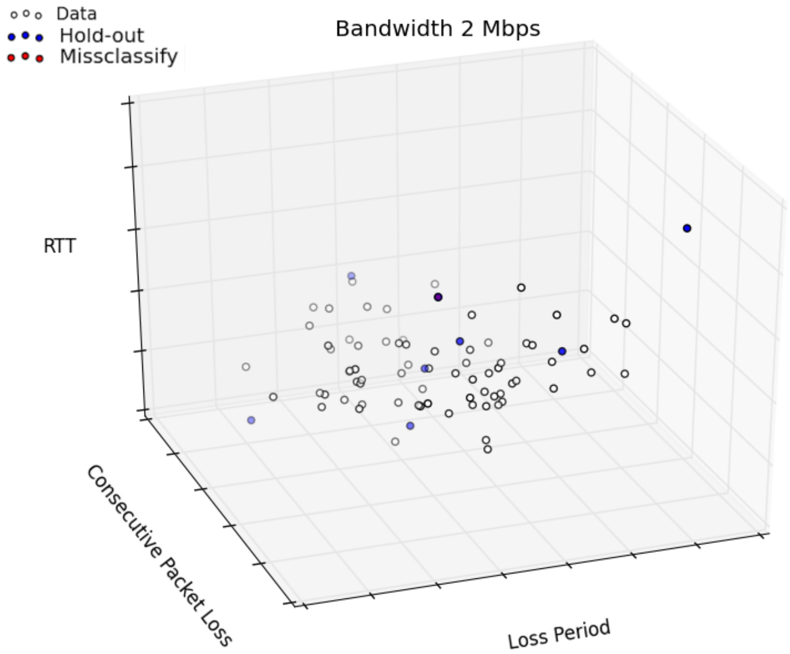


Figure 6.19: Logistic Regression: 3D-Model testing with hold-out dataset

Table 6.6
Logistic Regression: Accuracy Scores

Case.	Cross Validation Accuracy Score	Testing Accuracy
1	0.949603174603	1.0
2	0.913333333333	0.833333333333
3	0.826785714286	0.875
4	1.0	1.0
5	1.0	1.0

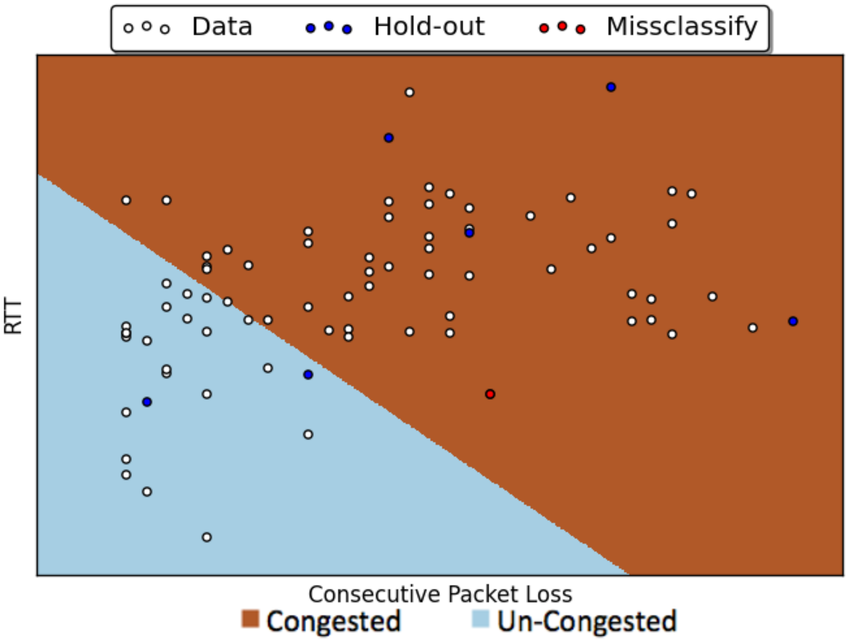


Figure 6.20: Logistic Regression: 2D-Model testing with hold-out dataset; Consecutive PL vs RTT

6.3 Summary of Results

The performances of these three classification algorithms with respect to classification accuracy was shown in Table 6.7, where it can be observed that all classifiers gave high classification accuracy for all cases. It is also observed that these classification algorithms give similar accuracies in all cases, leading to the conclusion that these classifiers performed equally well. However, KNN performs slightly better than the others in Case 2.

Table 6.7
Evaluation Results

Classifiers	Case 1	Case 2	Case 3	Case 4	Case 5
SVM	1.0	0.834	0.875	1.0	1.0
KNN	1.0	1.0	0.875	1.0	1.0
Logistic Regression	1.0	0.834	0.875	1.0	1.0

7

Conclusion and Future Work

THIS CHAPTER REPORTS the main conclusions that can be drawn from this study and discusses possible improvements and future work proposals. Further, it stresses on improving the ML predictive models by keeping on collecting more data and training model with that data to improve the accuracy of the classifiers.

7.1 Conclusion

HTTP video streaming services are sensitive to network impairments. Exponential growth in the popularity of HTTP video streaming services have lead service providers and network operators to develop reliable QoE aware management of networks to ensure the end users' ever increasing demands and to gain competitive edge.

In this study, a network monitoring method was developed for monitoring network congestion in Wi-Fi, based on QoE. For this, a QoE assessment method based on ML was proposed, which allows network operators and service providers to predict QoE from network level measurements. A four step approach was adopted for development of this method: development of network monitoring probes, establishing direct link between subjective QoE and objective network measurements, building a classifier to classify QoE levels and performance evaluation.

In the first step, the relationship between QoE and QoS was studied and it was determined that packet loss and RTT are dominating parameters, which affected end user experience. Communication over Wi-Fi is typically characterized by high bit error rates due to the dynamic nature of wireless links and often wireless packet losses are location and time varying. Consequently, loss pattern was considered as an important parameter and two derived loss metrics 'loss distance' and 'loss period' were considered as important parameters for measurements. Thus, a QoE assessment model was developed, which expresses QoE as a function of network QoS and an active measurement protocol TWAMP was implemented to extract KPIs by performing network-level measurements. In the second step, a direct link was established between subjective QoE and objective network measurements by designing various test cases. In the third step, ML was used to classify network congestion. Three ML algorithms: SVM, KNN and Logistic Regression were tested and evaluated to build a classification model, which translates network level measurements into QoE levels using the ML approach. Finally, this method was evaluated in a Wi-Fi testbed by varying different network conditions and the method's performance was tested with network traffic in both downlink and bi-directional paths. The results showed that this method performed well and gave high classification accuracy in all cases.

Network monitoring is a first step in a QoE aware network management system. With insights obtained from network monitoring, network operators and service providers can manage their network and services more effectively by making QoE aware congestion management decisions to bring back QoE to satisfactory level.

7.2 Future Work

There are several small and large scale upgrades that can further improve the proposed network monitoring method.

Small scale upgrades include expanding the QoE assessment model by introducing more QoS metrics. In this study, five QoS metrics are considered, which are further limited to three with ML feature selection. It would be interesting to further research for more QoS metrics that influence QoE and upgrade the QoE assessment model. Further, more data should be collected to train ML predictive models, in order to improve the accuracy of the classifiers. Also, an interesting upgrade would be to develop more test cases to determine more QoE quality levels. Moreover, test were performed only on laptops in this study. Experiments should be extended to smart phones and tablets, since with the ubiquitous availability of Wi-Fi, end users extensively use smart phones and tablets along with laptops to view video content. Furthermore, experiments can be extended for other video qualities, for example 720p, 1080p and 2160p.

Large scale upgrades should involve further development of the proposed monitoring method and extending it for multiple video qualities and other services for example, IP TV, web, gaming, VOIP, live video streaming. Another interesting case would be to develop test cases for dynamic network conditions. Finally, with the popularity of heterogeneous networks and current multimedia applications usage trends, this network monitoring method can also be extended to detect congestion for 3GPP networks.

A

Appendix

A.1 MOS

MOS is the most popular subjected assessment method. It uses opinionated scores that are mathematically averaged to obtain a quantitative indicator of system performance. The quality of service (e.g., voice, video) is rated by individual testers according to the Table A.1 and then, the arithmetic mean of all the individual scores is the final MOS score.

Table A.1
Mean Opinion Score (MOS) Scale

MOS	Quality	Impairment
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

References

[1] "Cisco Visual Networking Index: Forecast and Methodology, 2015–2020," Jun 2016, White Paper, Accessed: 2016-06-28. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html

[2] S. Taylor, "The Next Generation of the Internet - Revolutionizing the Way We Work, Live, Play, and Learn," April 2013.

[3] *Vocabulary for Performance and Quality of Service, Amendment 1: New Appendix I – Definition of Quality of Experience (QoE)*, ITU-T Rec. P.10/G.100, 2006.

[4] *Subjective Video Quality Assessment Methods for Multimedia Applications*, ITU-T Rec. P.910, 2008.

[5] *Methods for Subjective Determination of Transmission Quality*, ITU-T Rec. P.800, 1996.

[6] *Mean Opinion Score (MOS) Terminology*, ITU-T Rec. P.800.1, 2006.

[7] K. Hedayat, R. Krzanowski, A. Morton, K. Yum, and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP), RFC 5357," Aug 2008.

[8] S. Shalunov, B. Teitelbaum, A. Karp, J. Boote, and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)," RFC 4656 (Proposed Standard), Internet Engineering Task Force, September 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4656.txt

[9] P. Casas, P. Fiadino, A. Bar, A. D'Alconzo, A. Finamore, and M. Mellia, "YouTube All Around: Characterizing YouTube from Mobile and

Fixed-Line Network Vantage Points," in *Networks and Communications (EuCNC), 2014 European Conference on*, June 2014, pp. 1–5.

[10] J.-P. Laulajainen, A. Arvidsson, T. Ojala, J. Seppanen, and M. Du, "Study of YouTube Demand Patterns in Mixed Public and Campus WiFi Network," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2014 International*, Aug 2014, pp. 635–641.

[11] J. Li, J. Wu, G. Dan, A. Arvidsson, and M. Kihl, "Performance Analysis of Local Caching Replacement Policies for Internet Video Streaming Services," in *Software, Telecommunications and Computer Networks (SoftCOM), 2014 22nd International Conference on*, Sept 2014, pp. 341–348.

[12] J. Gustafsson, G. Heikkila, and M. Pettersson, "Measuring Multimedia Quality in Mobile Networks with an Objective Parametric Model," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, Oct 2008, pp. 405–408.

[13] R. K. P. Mok, E. W. W. Chan, and R. K. C. Chang, "Measuring the Quality of Experience of HTTP Video Streaming," in *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*, May 2011, pp. 485–492.

[14] M. N. Garcia, D. Dytko, and A. Raake, "Quality Impact Due to Initial Loading, Stalling, and Video Bitrate in Progressive Download Video Services," in *Quality of Multimedia Experience (QoMEX), 2014 Sixth International Workshop on*, Sept 2014, pp. 129–134.

[15] P. Casas, A. Sackl, S. Egger, and R. Schatz, "YouTube & Facebook Quality of Experience in Mobile Broadband Networks," in *Globecom Workshops (GC Wkshps), 2012 IEEE*, Dec 2012, pp. 1269–1274.

[16] S. Tasaka and Y. Ishibashi, "Mutually Compensatory Property of Multimedia QoS," in *Communications, 2002. ICC 2002. IEEE International Conference on*, vol. 2, 2002, pp. 1105–1111 vol.2.

[17] R. Schatz, T. Hoßfeld, and P. Casas, "Passive YouTube QoE Monitoring for ISPs," in *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, July 2012, pp. 358–364.

[18] Q. Dai and R. Lehnert, "Impact of Packet Loss on the Perceived Video Quality," in *Evolving Internet (INTERNET), 2010 Second International Conference on*, Sept 2010, pp. 206–209.

[19] T. N. Minhas, O. Gonzalez Lagunas, P. Arlos, and M. Fiedler, "Mobile Video Sensitivity to Packet Loss and Packet Delay Variation in Terms of QoE," in *Packet Video Workshop (PV), 2012 19th International*, May 2012, pp. 83–88.

[20] M. A. Mehmood, C. Sengul, N. Sarrar, and A. Feldmann, "Understanding Cross-Layer Effects on Quality of Experience for Video over NGMN," in *Communications (ICC), 2011 IEEE International Conference on*, June 2011, pp. 1–5.

[21] S. Khorsandroo, R. M. Noor, and S. Khorsandroo, "A Generic Quantitative Relationship Between Quality of Experience and Packet Loss in Video Streaming Services," in *Ubiquitous and Future Networks (ICUFN), 2012 Fourth International Conference on*, July 2012, pp. 352–356.

[22] T. Kawano, K. Yamagishi, K. Watanabe, and J. Okamoto, "No Reference Video-Quality-Assessment Model for Video Streaming Services," in *Packet Video Workshop (PV), 2010 18th International*, Dec 2010, pp. 158–164.

[23] M. B. Myers, "Predicting and Measuring Quality of Service for Mobile Multimedia," in *Personal, Indoor and Mobile Radio Communications, 2000. PIMRC 2000. The 11th IEEE International Symposium on*, vol. 2, 2000, pp. 1032–1036 vol.2.

[24] S. Qiu, H. Rui, and L. Zhang, "No-Reference Perceptual Quality Assessment for Streaming Video Based on Simple End-to-End Network Measures," in *Networking and Services, 2006. ICNS '06. International conference on*, July 2006, pp. 53–53.

[25] F. You, W. Zhang, and J. Xiao, "Packet Loss Pattern and Parametric Video Quality Model for IPTV," in *Computer and Information Science, 2009. ICIS 2009. Eighth IEEE/ACIS International Conference on*, June 2009, pp. 824–828.

[26] K. Singh and G. Rubino, "Quality of Experience Estimation Using Frame Loss Pattern and Video Encoding Characteristics in DVB-H Networks," in *Packet Video Workshop (PV), 2010 18th International*, Dec 2010, pp. 150–157.

[27] N. D. Narvekar, T. Liu, D. Zou, and J. A. Bloom, "Extending G.1070 for Video Quality Monitoring," in *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, July 2011, pp. 1–4.

[28] Y. Kang, H. Chen, and L. Xie, "An Artificial-Neural-Network-Based QoE Estimation Model for Video Streaming Over Wireless Networks," in *Communications in China (ICCC), 2013 IEEE/CIC International Conference on*, Aug 2013, pp. 264–269.

[29] N. Staelens, G. V. Wallendael, K. Crombecq, N. Vercammen, J. D. Cock, B. Vermeulen, R. V. de Walle, T. Dhaene, and P. Demeester, "No-Reference Bitstream-Based Visual Quality Impairment Detection for High Definition H.264/AVC Encoded Video Sequences," *IEEE Transactions on Broadcasting*, vol. 58, no. 2, pp. 187–199, June 2012.

[30] C. Wang, X. Jiang, F. Meng, and Y. Wang, "Quality Assessment for MPEG-2 Video Streams Using a Neural Network Model," in *Communication Technology (ICCT), 2011 IEEE 13th International Conference on*, Sept 2011, pp. 868–872.

[31] M. Venkataraman, M. Chatterjee, and S. Chattopadhyay, "Evaluating Quality of Experience for Streaming Video in Real Time," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, Nov 2009, pp. 1–6.

[32] P. Orosz, T. Skopkó, Z. Nagy, P. Varga, and L. Gyimóthi, "A Case Study on Correlating Video QoS and QoE," in *Network Operations and Management Symposium (NOMS), 2014 IEEE*, May 2014, pp. 1–5.

[33] F. Metzger, A. Rafetseder, D. Stezenbach, and K. Tutschku, "Analysis of Web-Based Video Delivery," in *FITCE Congress (FITCE), 2011 50th*, Aug 2011, pp. 1–6.

[34] M. Vilas, X. G. Paneda, R. Garcia, D. Melendi, and V. G. Garcia, "User Behavior Analysis of a Video-on-Demand Service with a Wide Variety of Subjects and Lengths," in *Software Engineering and Advanced Applications, 2005. 31st EUROMICRO Conference on*, Aug 2005, pp. 330–337.

[35] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "YouTube Traffic Characterization: A View From the Edge," Oct 2007.

[36] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, "A Measurement Study of a Large-Scale P2P IPTV System," *IEEE Transactions on Multimedia*, vol. 9, no. 8, pp. 1672–1687, Dec 2007.

[37] A. Finamore, M. Mellia, M. M. Munafo, R. Torres, and S. G. Rao, "YouTube Everywhere: Impact of Device and Infrastructure Synergies on User Experience," Nov 2011.

[38] L. Chen, Y. Zhou, and D. M. Chiu, "Video Browsing - A Study of User Behavior in Online VoD Services," in *Computer Communications and Networks (ICCCN), 2013 22nd International Conference on*, July 2013, pp. 1–7.

[39] T. D. Pessemier, K. D. Moor, W. Joseph, L. D. Marez, and L. Martens, "Quantifying the Influence of Rebuffering Interruptions on the User's Quality of Experience During Mobile Video Watching," *IEEE Transactions on Broadcasting*, vol. 59, no. 1, pp. 47–61, March 2013.

[40] Y. Chen, K. Wu, and Q. Zhang, "From QoS to QoE: A Tutorial on Video Quality Assessment," *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, pp. 1126–1165, Secondquarter 2015.

[41] F. Li, M. Li, R. Lu, H. Wu, M. Claypool, and R. Kinicki, "Measuring Queue Capacities of IEEE 802.11 Wireless Access Points," in *Broadband Communications, Networks and Systems, 2007. BROADNETS 2007. Fourth International Conference on*, Sept 2007, pp. 846–853.

[42] A. Nafaa, Y. Hadjadj-Aoul, and A. Mehaoua, "On Interaction Between Loss Characterization and Forward Error Correction in Wireless Multimedia Communication," in *Communications, 2005. ICC 2005. 2005 IEEE International Conference on*, vol. 2, May 2005, pp. 1390–1394 Vol. 2.

[43] M. Borella, D. Swider, S. Uludag, and G. Brewster, "Internet Packet Loss: Measurement and Implications for End-to-End QoS," in *Architectural and OS Support for Multimedia Applications/Flexible Communication Systems/Wireless Networks and Mobile Computing., 1998 Proceedings of the 1998 ICPP Workshops on*, Aug 1998, pp. 3–12.

[44] V. Mohan, Y. R. J. Reddy, and K. Kalpana, "Active and Passive Network Measurements: A Survey," *(IJCSIT) International Journal of Computer Science and Information Techonologies*, vol. 2, no. 4, pp. 1372–1385, 2011.

[45] F. Michaut and F. Lepage, "Application-Oriented Network Metrology: Metrics and Active Measurement Tools," *Communications Surveys Tutorials, IEEE*, vol. 7, no. 2, pp. 2–24, Second 2005.

[46] N. Soumyalatha, R. K. Ambhati, and M. R. Kounte, "Performance Evaluation of IP Wireless Networks Using Two Way Active Measurement Protocol," in *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on*, Aug 2013, pp. 1896–1901.

[47] I. Bäckström, "Performance Measuremnt of IP Networks using Two-Way Active Measurent Protocol," 2009.

[48] H. S. Saqlain Haider, "Available Bandwidth Measurement in 4G Networks," 2013.

[49] H. Wang, C. Ma, and L. Zhou, "A Brief Review of Machine Learning and Its Application," in *Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference on*, Dec 2009, pp. 1–4.

[50] *Opinion Model for Video-Telephony Applications*, ITU-T Rec. G.1070, 07 2012.

[51] J. F. Kurose and K. Ross, *Computer Networking: A Top-Down Approach*, 6th ed. Essex, England: Pearson Education Limited, 2013.

[52] L. Plissonneau, E. Biersack, and P. Juluri, "Analyzing the Impact of YouTube Delivery Policies on User Experience," in *Teletraffic Congress (ITC 24), 2012 24th International*, Sept 2012, pp. 1–8.

[53] T. Hoßfeld, M. Seufert, and R. Schatz, "Quantification of YouTube QoE via Crowdsourcing," in *IEEE International Symposium on Multimedia (ISM)*, Dec 2011.

[54] "App Coverage; Rethinking Network Performance For Smartphones," Sept 2013, White Paper.

[55] *Definitions of Terms Related to Quality of Service*, ITU-T Rec. E.800, 2008.

[56] *End-User Multimedia QoS Categories*, ITU-T Rec. G.1010, 2001.

[57] X. Zhang, Y. Xu, H. Hu, Y. Liu, Z. Guo, and Y. Wang, "Profiling Skype Video Calls: Rate Control and Video Quality," in *INFOCOM, 2012 Proceedings IEEE*, March 2012, pp. 621–629.

[58] A. Morton, "Round-Trip Packet Loss Metrics, RFC 6673," Aug 2012.

[59] R. Koodli and R. Ravikanth, "One-way Loss Pattern Sample Metrics, RFC 3357," Aug 2002.

[60] C. Demichelis and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM), RFC 3393," Nov 2002.

[61] G. Almes, K. Kalidindi, and M.Zekauskas, "A Round-trip Delay Metric for IIPM, RFC 2681," Sept 1999.

[62] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods.* New York, NY, USA: Cambridge University Press, 2000.

[63] V. N. Vapnik, *The Nature of Statistical Learning Theory.* New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[64] L.-C. Böiers, *Mathematical Methods of Optimization.* Lund, Sweden: Studentlitteratur AB, 2010.

[65] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*, ser. Springer Texts in Statistics. Springer, 2013.

[66] R. V. Krejcie and D. W. Morgan, "Determining Sample Size for Research Activities," in *Educational and Psychological Measurement*, vol. 30, 1970, pp. 607–610.

LUND UNIVERSITY

Series of Master's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2016-533

http://www.eit.lth.se