MACHINE LEARNING QOE PREDICTION FOR VIDEO STREAMING OVER HTTP

by

MARY ABDELMALEK

A thesis submitted to the

School of Computing

in conformity with the requirements for

the degree of Master of Science

Queen's University

Kingston, Ontario, Canada

July 2020

Abstract

Streaming video services have been growing rapidly in the past decade due to the wide adoption of more capable mobile devices such as smartphones and tablets, together with the deployment of higher capacity mobile networks and more efficient video compression and streaming techniques. Dynamic adaptive streaming over hypertext transfer protocol (DASH) can adapt bitrates to fluctuating network bandwidth. This makes measurement of perceived video quality a significant task. Therefore, there has been a strong demand for Quality of Experience (QoE) measurements and prediction models. The work in this thesis focuses on improving the performance of existing QoE prediction models that still have a few limitations. We propose a feature enhancement approach for QoE prediction using machine learning (ML) methods. Given the need for an accurate and reliable data set to do our analysis, train and evaluate our machine learning models, we utilize the Waterloo Streaming QoE Database III (SQoE-III) that combines the effects of video compression, initial buffering, and stalling events along with the subjective user responses to them. Additionally, the database contains the results of implementing various objective video quality assessment (VQA) models.

First, we perform an extensive data and correlation analysis to study the effects of the different QoE key influence factors (KIF) which account for the video quality degradation. We further study the effect of quality switching that includes switching up and down, switch magnitude, ratio/time spent on the highest quality level/layer, in addition to the effect of stalling expressed as the number of stalls and their duration and initial buffering time. We analyze the effect of various objective video quality assessment (VQA) models on the QoE. The results of the analysis reveal the dependency between objective VQA models features, playback stalling, and quality switching features that affect the QoE. Towards effectively predicting user QoE and based on the results of our analysis, we identify/introduce a new set of enhanced features that combine objective VQA features, playback stalling features, and quality switching features, as input to various machine learning models to predict the values of QoE. Experimental results show that the proposed models are in close agreement with subjective opinions and significantly outperforms existing

QoE prediction models and can also provide a highly effective and efficient future for QoE prediction in video streaming services.

Acknowledgements

I would like to express my deep gratitude and respect for my supervisor Professor Hossam Hassanein for giving me the opportunity to work under his supervision, and for his endless guidance and continuous feedback since the start of my research to the final phase. What I have accomplished would not have been possible without his constant support, patience, and encouragement.

My deep gratitude to Professor Hazem Abbas for his beneficial comments, valuable suggestions, and sharing his vision during the planning and development of the work in this research. He was always there whenever I need help.

I would like to thank my amazing husband and my wonderful daughter for always being by my side and helping me survive during my difficult times.

I dedicate my Thesis to the soul of my father, who will remain in my heart and mind forever, and I would like to express my deep love to my mother and my sister for always being there for me and giving me endless love and support.

Last but not least, I would like to thank all my colleagues in the TRL lab for their great company, and a special thanks goes to Basia for her advice and willingness to help all the time.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

QoE	Quality of Experience
HTTP	Hypertext Transfer Protocol
DASH	Dynamic Adaptive Streaming over HTTP
HAS	HTTP Adaptive Streaming
KIF	Key Influence Factors
VQA	Video Quality Assessment
SQoE	Streaming Quality of Experience
QoS	Quality of Service
ISP	Internet Service Provider
MOS	Mean Opinion Score
ACR	Absolute Category Rating
PSNR	Peak Signal to Noise Ratio
SSIM	Structural Similarity Index Metric
VQM	Video Quality Metric
VMAF	Video Multimedia Assessment Fusion
STRRED	Spatio-Temporal Reduced Reference Entropic Differences
VIIDEO	Video Intrinsic Integrity and Distortion Evaluation Oracle Model
ABR	Adaptive Bitrate Streaming
MSE	Mean Square Error
IFs	Influence Factors
VQEG	Video Quality Expert Group
PLCC	Pearson Linear Correlation Coefficient
SRCC	Spearman’s Rank Correlation Coefficient
OR	Outlier Ratio

OTT	Over the Top
HD	High Definition
MS-SSIM	Multi-Scale Structural Similarity Index
SI	Spatial Information
TI	Temporal Information
DMOS	Difference Mean Opinion Score
MICE	Multivariate Imputation by Chained Equations
SVM	Support Vector Machines
SVR	Support Vector Regression
SVC	Support Vector Classification
KNN	K-Nearest Neighbors
MAE	Mean Absolute Error
RMSE	Root Mean Square Error

Chapter 1

Introduction

Given the expanding use of mobile video devices and the huge network bandwidth demands of streaming users, the biggest challenge in video content delivery is to enhance end user's quality of experience (QoE) by creating better network-aware strategies. To this end, DHAS is being used by content providers as a way of dealing with network fluctuations. Dynamic adaptive streaming over hypertext (DASH) transfer protocol, is an international standard for adaptive video streaming, that provides video streams in a variety of bitrates (rate-adaptive). However, DASH has introduced an additional level of difficulty for measuring video quality since it varies the video quality during streaming to match available network bandwidth, and this causes fluctuation and results in QoE degradation. Predicating user's quality of experience (QoE) for adaptive video streaming is needed to help service providers in delivering smooth, high-quality content to their customers in a cost-effective manner and better network resource allocation strategies**.** Predicting user QoE is a non-trivial task. Various QoE prediction models were proposed in the literature, but these models have limitations. In this chapter, we will shed light on some of these limitations and present our proposed solutions that will solve these problems and overcome a few of the limitations of these methods.

1.1 Problem Statement

There have been tremendous developments in the number of available mobile devices due to the large available network capacity and the evolving streaming technologies to transmit media service to devices, in addition to the popular protocol standards that are rate adaptive. However, these streaming strategies usually add more complexity to measure the perceptual video quality. While streaming content providers such as YouTube and Netflix are both increasingly deploying Dynamic HTTP adaptive streaming (HAS) strategies, they are still looking for efficient ways to fulfil user experience expectations to have smooth and high-quality video content.

Predicting user's quality of experience (QoE) for adaptive video streaming is essential for the service providers in delivering this smooth and high-quality content to clients in a cost-effective manner and better network resource allocation methods**.** However, predicting user QoE is a non-trivial task as there are technical and perceptual factors that influence the quality of user experience, while adaptively streaming a video, and must be taken into consideration. Based on the results of previous studies [1] [2] [3], that initial delay, stalling and quality adaptation are the important influence factors, and each of these factors can include multiple dimensions. Various QoE prediction models were proposed in the literature, that studied the effects of these KIF on QoE. These studies have employed different approaches to tackle this issue, but they still face some limitations and shortcomings. Some of these include most of existing subjective video QoE datasets are limited in size, or their design is not suitable for streaming applications. In addition, they can't be used to develop general QoE models as they include only either quality change data or rebuffering data not both. In addition, these datasets are hand-crafted so it does not reflect the real-time impairments that user may perceive. Consequently, the usage of QoE quality prediction techniques was limited, as most mechanisms require large amounts of data. In addition, most of the proposed prediction models studied the interaction between one or two influence factors that may ignore the effect of others. Moreover, the effect of an objective VQA model on a subjective user QoE and its interaction with these KIF was not widely studied or investigated; although various methods on VQA were designed for specific applications such as quality assessment for static video or progressive video streaming. Furthermore, little work has been done to predict QoE based on objective VQA models that measure the perceptual effect of several distortions, such as compression, blurring, noise, or combine them with video impairments like initial buffering, rebuffering, the quality switch that happens during video streaming session and then compare them with subjective data comprising a wide variety of video sequences.

In this work, we investigate the effect of multiple (multidimensional) KIF on user QoE using the SQoE-III database, which combines different factors combined with the results of implementations of various objective video quality assessment models (VQA) models measurements. This data simulates realistic

network conditions in a typical video streaming scenario. It is publicly available and considered as the second largest database with 450 streaming video sessions evaluated by 34 subjects.

The streaming videos are impaired with initial buffering, playback buffering event and quality switching, and the various video quality assessment model measurements can be easily calculated from source videos. We performed a comprehensive correlation analysis to study the effect of quality variation, stalling (frame freezing), initial buffering (delay). We explore different issues that can have different effects on perceived quality. The impact of quality level variation can be represented by the number of switches, switch magnitude, time spent on the highest quality layer. Stalling can be represented by the time of stalling and its frequency/number. VQA measures need to be included in the QoE assessment. These factors are investigated in this work. Based on the results of this analysis, we found correlations between VQA measurement features, stalling features and quality variation features. These features are combined as input to various machine learning models to provide better prediction capabilities of the QoE measure for the tested dataset.

1.2 Motivation and Objectives

The motivation for this research is to improve the QoE prediction for HAS service by proposing a feature enhancement QoE prediction model using machine learning methods.

To do this, we carried out data analysis on the SQoE-III database. The dataset is publicly available and is used here to extract features that reflect different KIFs. In addition, the availability of source videos made it possible to measure the VQA model metric scores on different video frames. The dataset has a large number of real streaming sessions that include the outcomes of an extensive subjective study that reflects the effect of rebuffering and other quality changes. This has helped in doing a detailed analysis of the extracted features and, combined with the VQA measures, to further select the best set of enhanced features to perform QoE prediction to improve the predictive power.

The work presented in this thesis will follow the methodology below.

1. Use a publicly available dataset that offers real-time streaming video sequence, integrates many KIF that reflects many real impairments in the video session and being evaluated by a large user base.
2. Perform an extensive correlation analysis to investigate the effect of different factors on user QoE
3. Integrate VQA models in the correlation analysis, investigate its interaction with KIF and study their combined effect on QoE, something that is not widely studied in the past.
4. Propose a new set of features that improve the prediction power for the QoE prediction model using machine learning.
5. Perform detailed correlation and data analysis on the existing and the new proposed features.
6. Integrate the Video Quality assessment models in the prediction of QoE models.

1.3 Thesis Contributions

The major contributions of this research are outlined below.

1. We identify the most important features affecting the subjectively perceived quality of users while streaming a video by performing detailed correlation and data analysis
2. We introduce a model with improved predictive power, where we propose a new feature set that takes into consideration objective video quality measures and adaptation-related parameters. Specifically, the amplitude and frequency of switch fluctuation, the ratio of time spent on the highest quality layer, and the time and frequency of stalls are considered.
3. We evaluate the model performance on test data using multiple evaluation metrics.
4. Additionally, we perform a comparative analysis of the various machine learning models and recommended the best ones for predicting the QoE for the used dataset.

1.4 Thesis Outline

The thesis is organized into six chapters. Chapter 1 provides an introduction. Chapter 2 presents the literature review for existing QoE prediction models. In Chapter 3, we present the proposed approach (methodology) for QoE prediction; an analysis of the data is carried out on the used database. This

correlation and data analysis is applied to extract important key influence factors. In Chapter 4, we explain the preprocessing steps applied to the data and introduce the machine learning models that are used for QoE prediction. Chapter 5 discusses and compares the experimental results of the different models used for QoE prediction. Finally, Chapter 6 summarizes our findings and presents insight regarding potential future research directions.

Chapter 2

Background and Related Work

In this chapter, we present background information about the topics that will be discussed throughout this thesis. We will start with an introduction in Section 2.1. In Section 2.2, a brief definition of QoE is presented. Section 2.3 presents Video Quality of Experience Assessment (VQA) measurement and methodologies. In Section 2.4, we discuss the various influence factors which need to be taken into consideration for QoE model design. In section 2.5, performance evaluation metrics for QoE models are discussed. In Section 2.6, the HAS technology is presented. Section 2.7 reviews existing work in the field of HAS QoE models. In Section 2.8, an overview of the different types of machine learning algorithms is presented.

2.1 Introduction

According to Cisco Visual Networking Index and global mobile data traffic forecast that by 2021, the internet traffic will increase by 717 Terabyte, with video alone consuming 82% of the net consumption [4]. This is due to the significant advancement of network and user handheld technologies, especially the unrivaled progress of multimedia streaming services like Vimeo, Netflix, and YouTube. This has led to a switch in video delivery service from conventional Quality of Service (QoS) based assessments [5] to Quality of Experience (QoE) based assessment [6] , [7].

Various Video Quality assessment (VQA) models that measure the perceptual effects of several distortions such as compression, blurring, noise, jerkiness has been proposed aiming to predict the video quality as the end user perceives. In addition to the work done by multimedia service providers to improve video delivery service by increasingly adopting the use of HTTP Adaptive Streaming (HAS) technology.

However, there is a need to develop a reliable and accurate QoE prediction models that consider various network and application level factors (including several QoS factors) and aim at predicting the QoE as experienced by the end user.

2.2 QoE Definition

Quality of Experience (QoE) is a concept that describes the subjectively perceived quality of end users with a service [6], and it complements the QoS measure. QoE is defined as "the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of their expectations with respect to the utility and /or enjoyment of the application or service in the light of the user's personality and current state" [8] [9]. Additionally, "In the context of communication services, QoE is influenced by service, content, network, device, application, and context of use" [7]. This definition includes the end user's experience and their level of satisfaction with the service offered. By understanding the end user's experience and expectations, this will help in improving existing technologies and developing better future services. However, QoS does not perfectly reflect user perception for many services; it also fails to take user related factors into account. For example, in a video streaming scenario, the user does not explicitly notice delays in arriving packets because they are absorbed by the application buffer of the video player. The only effect the user may perceive is the playback interruption when the buffer is empty, which is considered a propagated network issue in the application layer. In short, QoS only describes the service delivery by objective network parameters such as jitter, delay, packet loss or throughput, which are considered technical measurements.

On the other hand, QoE is application-dependent and requires an understanding of all influence factors. To do this, extensive subjective studies need to be conducted. The results of these studies measure the QoE in terms of the Mean Opinion Score (MOS) on an ordinal scale ranging from 1 (bad) to 5 (excellent) , known as Absolute Category Rating (ACR) scale [10]), or in terms of acceptability scores.

In this work, we focus on predicting the overall QoE of a video session expressed in terms of MOS scores.

2.3 QoE Assessment methods

QoE assessment is the method of measuring or evaluating the QoE for a set of clients using an application or service with a dedicated procedure whereas taking under consideration the impacting factors that are simply collected, measured, controlled and reported using the procedures in [9]. The main objective of QoE

assessment is the design of a system that can identify the various factors and their influence on the end user QoE.

Multimedia data uses lossy compression for transportation over the internet to decrease the transport cost and the required bandwidth. As a result, some information will be lost, and higher compression ratios will result in a higher amount of information loss, and this will lead to artifacts which is unpleasant to the user. In most real-world application, it is hard to provide a service without any artifact. Hence a proper QoE model/metric is required to help quantify the amount and type of distortions that might occur and measure the magnitude of their effects on the end user QoE [11].

There are two main categories for the VQA models subjective and objective. On the one hand, subjective VQA depends on user feedback. Users can give feedback in the form of rates or scores based on their perception of the video quality. Such subjective assessment scores are typically expressed as Mean Opinion Score (MOS), which is the average of the scores collected from subjects/users doing the assessment. The common guidelines for performing these subjective tests are issued in ITU-R Rec BT.500 and ITU-T Rec P.910 [10], [11]. In these guidelines, the methods, procedures, and test settings that need to be done are described.

On the other hand, objective VQA methods are mathematical methods that aim at providing a quality score that closely reflects the perceived image/video quality. Objective VQA methods employ different metrics such as Peak Signal to Noise Ratio (PSNR) [12], Structural Similarity (SSIM) index [13], and VQM [14]. Theses metrics are calculated on a frame-by-frame basis and a final score is found as the average/mean of each individual score on the full duration of the video sequence. There are different pooling techniques/methods that combine different objective scores, viz. the exponential weighting method, the averaging method and the Minkowski method [15].

The popular approach to assess an objective quality metric's performance is to calculate the mean square error (MSE) and coefficient of correlation values between the estimated MOS scores from the objective VQA measurements and the real MOS scores from the subjective assessment. This must be done on the same set of test sequences [16].

In addition, most of the traditional objective VQA metrics were designed for quality estimation of compressed videos that are degraded due to packet losses that happen during the transmission process. They do not consider impairments such as quality switches, rebuffering, among others, which are present in HAS applications. This requires a new QoE estimation/prediction model that is designed for HAS applications and can take into consideration the Influence Factors (IFs) such as quality switching and rebuffering, for example, in combination with other impairments due to the lossy encoding operation.

2.4 QoE Influence Factors

A QoE IF is "any characteristic of a user, system, service, application, or context whose actual state or setting may have an influence on the Quality of Experience for the user" [16]. Influence factors on QoE can be organized into four categories as depicted in [16]:

1- **System IFs:** enclose a wide range of aspects that are related to media such as (quality fluctuation events) or network related such as (wired/wireless/mobile, bandwidth, delay, jitter, packet loss, etc.).

2- **Human IFs:** incorporate individual characteristics of a user such as memory and recency effect, his usage history of the application (e.g., browsing history), his expectations from the service and demographic and socio-economic background.

3- **Context IFs:** include location, viewing environment, time of the day, type of usage, and time of service consumption (peak time, etc.)

4- **Content IFs:** explains the characteristics of the contents such as type of video, its duration and content aspects related to complexity including (temporal and spatial complexity).

Managing QoE is a complex task and requires as a key step the design of an efficient QoE model, which is defined as “An algorithm with the purpose of estimating the subjective (perceived) quality of a media sequence,” in ITU-T Recommendations P.1201 [6]. This definition considers various influence factors and tries to estimate the end user's QoE.

2.5 Performance Evaluation of Objective QoE Models

There are criteria that are aforementioned in Video Quality Experts Group (VQEG) FRTV Phase I and next in VQEG FRTV Phase II [17], [18], that are used to evaluate the performance of an objective video QoE model which are:

- **Prediction Accuracy:** refers to the ability of the model to predict the subjective rating scores with a small error [11] [18].
- **Prediction Monotonicity:** refers to the degree of the model's prediction agreement with the relative magnitude of subjective rating scores [11] [19].
- **Predication Consistency:** refers to the ability of a model to maintain prediction accuracy over a wide range of test sequences with a variety of video impairments [11] [19].

There are many metrics to evaluate the performance of a model, such as the Pearson Linear Correlation Coefficient (PLCC) [20], which is used to evaluate the model prediction accuracy by calculating the correlation between the predicted and actual subjective rating scores. In the same way, the Spearman's Rank Correlation Coefficient (SRCC) [20] is used to evaluate the prediction monotonicity of a model between predicted and actual subjective scores. One can also evaluate the prediction consistency of the model using other measurements, such as the outlier Ratio (OR) [19]. QoE models that provide insight into how the IFs affect the QoE of the end user are of greater value against those models that fail to provide such insight.

In this work, we use machine learning methods to build QoE prediction models for the HTTP Adaptive Video Streaming (HAS) application. In the following sections, we discuss the basic concepts of HAS and the existing HAS QoE models, followed by a brief introduction to the machine learning approaches.

2.6 HTTP Adaptive Video Streaming

The basic idea of HAS is that the video file is encoded at different representation levels and then divided into chunks (also referred to as segments) of equal durations (often 2, 4, or 10 seconds, depending on the

standard/implementation) which are then stored on a server. When a first request for the video file is made by the client, the server sends the corresponding manifest file (e.g., mpd for DASH, .m3u8 for HLS). These examples corresponds to the manifest files in a different implementation of HAS named Dynamic adaptive streaming over HTTP (DASH) [20]. DASH is an open source international standard developed by MPEG [21], while HLS stands for Apple HTTP Live Streaming which is a proprietary and vendor specific HAS implementation [22]. Each manifest file has the details about the video file such as video duration, available representation levels, codec, segment size, etc. The client then requests for video chunks based on its rate adaptation logic. The rate adaptation strategy/approach used at the client can be widely classified into buffer-based, throughput-based and hybrid approach. For a extensive survey of the rate adaptation methods for HAS, the reader can refer to the survey paper of Kua et al. [23]. Figure 2-1 illustrates the concept of streaming assuming a throughput-based rate adaptation method. It can be observed that the client, based on its network condition, adapts the quality of the video to provide a smooth streaming experience to the end user.

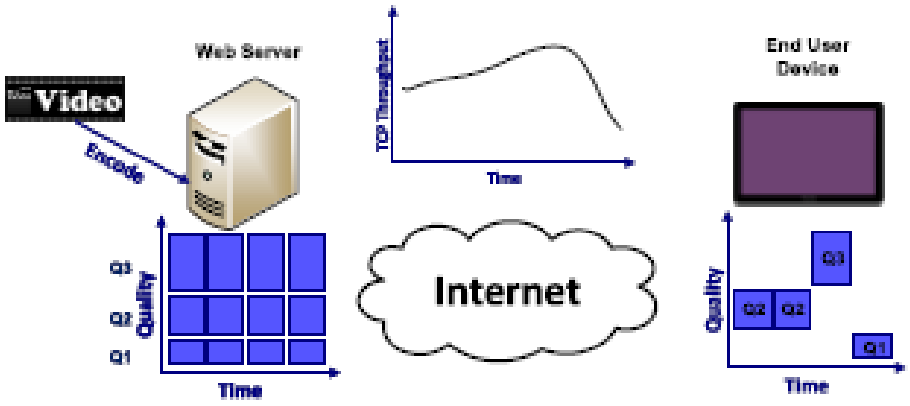


Figure 2-1:HAS Schematic(Q3,Q2,Q1 denote high, medium and low quality levels respectively [50]

The client starts to play out the video when the buffer contains enough data. This waiting time between the video request and the start of playback is referred to as initial delay and influences the QoE of HAS. When the buffer is depleted, the video playback stalls and can only be resumed after sufficient data is received. Streaming systems try to avoid stalling by adaptation of video bitrate to the current network conditions. To

change the video bitrate, some aspects of the transmitted video must be changed, e.g., frame rate, resolution or compression.

HAS is one of the most popular streaming technologies for video delivery over the Internet, currently used by the primary Over the Top (OTT) providers such as YouTube and Netflix. Both, contribute more than 50% of the total peak internet traffic for fixed access networks in North and Latin America [24]. The increasingly adoption of HAS can be credited to its scalability, since no special streaming server infrastructure is needed for the reuse of existing infrastructure. In addition to the existence of HTTP based progressive download solutions. It uses reliable transport protocols TCP with guaranteed packet delivery and congestion control mechanism. Therefore, network impairments such as packet loss do not cause any artifact as blurring, motion jerkiness, etc. In addition, it runs on HTTP, which is a firewall friendly, and considered a stateless protocol where server does not store any information related to the client and the requests. This is useful from the network point of view (e.g., load balancing) as each request is treated individually and can be handled by any of the servers without keeping track of which server is serving which request.

There are some challenges in using HAS, including quality switching as it adapts the bitrate to fluctuating bandwidth and it depends on the network conditions and buffer status. Since quality switching is an important feature of HAS, which helps in minimizing the number of stalling events, frequent quality switching will result in an increased user annoyance. In addition, for most of the HAS applications, full segment download is required before playback can start. Such requirements can lead to increased stalling events during playback. It needs higher storage due to the creation of multiple quality representations for the same video/audio content.

2.7 Related works on HAS QoE models

In this section, we will show the QoE models particular to HTTP Adaptive video streaming. Mok et al. [25] presented one of the most punctual works towards building a QoE model for HAS applications. Their model quantifies the QoE for HAS applications as a simple linear equation using application and network layer

QoS parameters, considering (low, medium and high levels) of initial startup delay, rebuffering frequency and rebuffering duration, respectively. They found that rebuffering frequency is the main IF. Inspite their work proposes a simple linear equation that maps application QoS metrics to QoE, the subjective assessment used to perform the regression analysis to obtain the proposed model was limited to only a single video with a single resolution and rated by 10 users, which is not realistic for most HAS applications. Also, they assume constant network bandwidth, packet loss and round trip time which is not always true for the real networks. The important issue they have left out is considered one of the major IFs of HAS, quality switching [11].

Rodriguez et al. [26] introduce a model that takes temporal interruptions initial startup delay, (location, number and length of the rebuffering events) into account during a video session, and quality switching (location and number) to propose a new metric $VsQM_{DASH}$ The model is shown to be of low complexity and suitable for devices such as tablets and mobile phones which have limited power and processing capabilities. The model design is done using similar types of playout patterns and it also considers a fixed number (four) of segments. This raises an open question considering the performance of the model on unknown datasets employing different playout patterns and of different video length.

Hoÿfeld et al. [27] investigate the effect of these five IFs: last quality level, quality switching amplitude, recency time for the different number of switches, time on the highest quality and the frequency of quality switching. The finding of the paper is that quality switching dimness the effect of recency and recency time (total duration of high-quality playback after last quality switch) and does not affect the QoE, it was observed that the time on highest quality level has more significant impact than that of the frequency of rebuffering. The authors do not consider other IFs (that are based on statistical analysis) and proposed a simple QoE model, that only consider two IFs and it takes into account the effect of amplitude (which is the difference between the two quality levels) only and the time on the highest level using an exponential relationship. In their subjective assessment, they used three full High definition (HD) video sequences and various rebuffering conditions and encoding (not described in the paper). An increase of MOS with an increase in the resolution or bitrate was observed [11].

Yamagashi and Hayashi [28] display a quality model consisting of a video quality estimation module and an audio quality estimation module, and they integrate the short term (per-second) audio-visual coding quality with other IFs factors such as the total length of rebuffering events and the number of rebuffering events. The authors conclusion is that the model performs quite well for video sequences without rebuffering and also with some specific sequences with rebuffering (where rebuffering occurs at the point where the compression quality is worse) [11]. This leads to the observation that the amount of QoE degradation due to rebuffering is dependent on the quality of the video frame where rebuffering occurs. The main limitation of this work is they did not include other IFs such as the initial loading delay, the performance evaluation of individual quality estimation modules, and so on/to name a few [11].

The work in Wang et al. [29] presents two QoE models based on regression and classification, using video quality metric Peak signal to Noise Ratio (PSNR). They propose an evolved PSNR (*ePSNR*) model based on standard deviation, average, minimum and maximum of differential PSNR (*dPSNR*). In their classification model, they use weighted K-nearest neighbor (*WkNN*) based on segment bitrate and video segment position to predict QoE. The subjective tests for both models use only two videos using real-world LTE network testbed[[11].

Duanmu et al. [30] present Streaming Quality Index (SQI) as a QoE model that considers the integrated effect of rebuffering, encoding quality and initial loading delay. The overall quality is computed from the instantaneous quality in a moving average fashion. At each time slot, this instantaneous quality is a linear combination of instantaneous video presentation quality that is estimated at the server side using a frame-level VQA model and the impact of rebuffering at individual frames. The authors assume that each rebuffering event is additive and independent, then they model the decline in memory of memory retention due to the rebuffering happening based on Ebbinghaus forgetting curve [31]. After that they use this in a piecewise model to get the combined effect of rebuffering on QoE degradation. An evaluation of existing models (SSIM, MS-SSIM ,PSNR, SSIMplus) [32], FTW [33] and Mok et al. [25] and their proposed Streaming Quality Index (SQI) using PSNR, SSIM, SSIM, SSIMplus , MS-SSIM on designed database shows that the proposed SQI model, when used with SSIMplus as VQA model, has highest performance,

while other SQI models (SQI with PSNR, SSIM and MS-SSIM as VQA) perform better than other compared models. Their work is considered as a big step forwards towards QoE modeling by considering both rebuffering related information and encoded video quality, with reasonable performance on given dataset. The limitations of their work is that the database and IFs considered are limited due to the short duration of sequences (only 10 seconds videos, with fixed duration rebuffering and using only two rebuffering events at start and middle) which is not realistic.

Bampis and Bovik [34] propose Video ATLAS which is machine learning-based framework, which integrates objective VQA metrics and QoE related features such as rebuffering-related and memory-related functions, to predict user QoE. The video quality is evaluated using well-known image and video quality metrics and other IFs. Then they combine the calculated features to be used with various learning-based algorithms. The authors conclusion was that IFs such as bitrate changes and rebuffering should be considered together and not independently, in spite of this negates with the approach of many other models discussed (e.g., [28], [35]). Based on the results, it is observed that the video quality model used for the prediction of compressed video quality plays a very important role in the QoE prediction quality [11].

Ghadiyaram et al. [36] perform a subjective study to understand the impact of dynamic network impairments such as the stalling events on QoE of users watching videos on mobile devices. They construct a database named LIVEMSV that contains 176 distorted videos generated from 24 reference videos with 26 hand-crafted stalling events. The authors used the single stimulus continuous quality evaluation procedure. In this procedure the reference videos are also evaluated to obtain a difference mean opinion score (DMOS) for each distorted video sequence. This study lacks the presence of video compression and quality switching that reduces the relevance of the database to real-world HAS scenarios.

In Waterloo Streaming QoE Database I (SQoE-I) [30] the interaction between video presentation quality and playback stalling experiences was investigated. This database consists of 20 reference videos of size 1920×1080 pixels of diverse content. They encoded each reference video into 3 bitrates with x264 encoder, next a 5-second stalling event is simulated and added at either the beginning or the middle point of the encoded sequences. In total there are 200 video sequences, this include 20 source videos, 60 compressed

videos, 60 initial buffering videos, and 60 mid-stalling videos. The most important finding of this study is that the video presentation quality of the stalled frame constitutes strong correlation with the dissatisfaction level of the stalling event with statistical analysis [11].

In Waterloo Streaming QoE Database II (SQoE-II) [37] short and long video clips of size 168 and 588 respectively was used. These are varied in spatial resolution, compression level and frame-rate. A path-analytical experiments to address the confounding factors and better explore the space of quality adaptations was carried out by the authors. In spite the interesting analysis, the database may not serve as a benchmark database due to the lack of stalling events [11].

2.8 Machine Learning

Machine learning is a tool for turning information into knowledge, using techniques and algorithms to build a mathematical model to automatically make predication or decisions without being explicitly programmed to perform the task [38]. Machine learning has been shown to be successful in various fields, such as computer vision, speech recognition and natural language processing. The general categories of machine learning algorithms are supervised learning, unsupervised learning, and reinforcement learning

2.8.1 Supervised Learning

Supervised learning is the task of learning a function that maps the input features to desired output values [39]. Supervised learning algorithms receive a labeled dataset, where each sample in the dataset has a corresponding label or ground truth. Supervised learning can be further divided into classification and regression applications

- **Classification:** Classification models approximate a mapping from the input variables to a discrete output variable [40]. A classification model classifies the inputs into one of two or more classes. The performance of the model is often measured by the classification accuracy.
- **Regression**: Regression models approximate a mapping from the input variables to a continuous output variable [41]. The performance of a regression model is measured in terms of errors made in the model's predictions.

2.8.2 Unsupervised Learning

Unsupervised learning algorithms receive a set of input variables with no output variable or label [42]. The objective of unsupervised learning is to learn the underlying structure of the data and find an efficient representation for it. Two common unsupervised learning tasks are clustering and dimensionality reduction:

- **Clustering:** Clustering groups samples into clusters based on their similarities [43]. Samples which are similar to each other are grouped into the same cluster.
- **Dimensionality Reduction:** The idea of dimensionality reduction is to project samples from a high-dimensional space onto a lower dimensional space without losing much information [44]. This reduces the complexity of the data while retaining its structure.

2.8.3 Reinforcement Learning

Reinforcement learning is a class of machine learning where an agent interacts with the environment [45]. The goal of reinforcement learning is to train the agent in such a way that for a given environmental state, it chooses the optimum action that yields the highest reward.

2.9 Summary

In this chapter, we presented background information regarding the work discussed in this thesis. We provided an overview about the work done in the literature for QoE modeling for HAS service. In the next chapter, we give detailed description on the dataset we used, and we will present our proposed methodology and analysis done on dataset.

Chapter 3

Proposed Methodology

In this chapter, a description of Waterloo SQoE-III database and the detailed correlation and data analysis that was carried out is presented. Section 3.1 shows the description of the dataset and its characterization. Section 3.2 explores the various features/factors of the data and explains how they are computed. In Section 3.3, we will show the results of the carried-out analysis to understand the correlation between the different features and the user perceived quality (MOS). In addition, the results of feature visualization and exploration is presented. Section 3.4 explains the objective video quality assessment models used and explores results of correlation analysis to compare them with the subjective perceived quality. Section 3.5 presents and explains our feature enhancement approach.

3.1 Dataset Description and characterization

In this work, the publicly available database the Waterloo Streaming Quality of Experience III (SQoE-III) [46] is used. The SQoE-III database, which is the most realistic, and to date is the second largest database, consisting of 450 streaming videos created from diverse source contents and distortion patterns, with six adaptation algorithms of diverse characteristics under 13 representative network conditions. These Adaptation Algorithms include the following. Rate-based [47]: the rate-based ABR algorithm, which is the default ABR strategy in the DHAS standard. It works by selecting the maximum available bitrate less than predicted throughput using arithmetic mean of previous 5 chunks. The BBA [48]: is a buffer-based adaptation work by selecting the bitrate as a piecewise linear function of buffer occupancy. AIMD [49]: This algorithm selects the bitrate based on the bandwidth estimation using the previous downloaded segment in an additive increase and multiplicative decrease manner. ELASTIC [50]: This algorithm uses PI controller to maintain a constant duration of video in the buffer using 5 seconds. QDASH [51]: picks an

intermediate bitrate when there is a bandwidth drop to avoid the negative impact of abrupt quality degradation. FESTIVE [52]: This algorithm balances both efficiency and stability, and incorporates fairness.

The database contains in total 1560 video sessions produced from 20 source videos and 6 Adaptive bitrate algorithms and 13 bandwidth profiles. Almost around 25% of them were found to be duplications of each other and were removed from subjective experiments. The result was 1164 unique streaming videos. They selected 10 randomly streaming sessions from resulting streaming video pool for 15 content due to the limited duration of subjective experiment then reconstructed all streaming session for the other 5 contents. In summary waterloo database contains 20 source videos and 450 simulated streaming videos [39, 48].

All streaming videos are assessed by a total of 34 subjects, of which 19 were males and 15 were females. The test videos included in the SQoE III database was generated from conducting a set of Dynamic Adaptive Video Streaming (DASH) video streaming experiments, and then recorded the relevant streaming activities, and reconstructed the streaming session using video processing tools. A log file was generated including the selected bitrates, duration of initial buffering, start time, and end time of each stalling event after each video streaming session, From the recorded logs, each streaming session was reconstructed by concatenating streamed bitrate representations, appending blank frames to the test video to simulate initial buffering, and inserting identical frames at the buffering time instance to simulate stalling event [39]. The database consists of 20 high-quality source videos of size 1920 x 1080 pixels that cover diverse content, and of an average duration range between 13 and 21 seconds (after adding these impairments). Details about distribution (characterization) of the impairments included

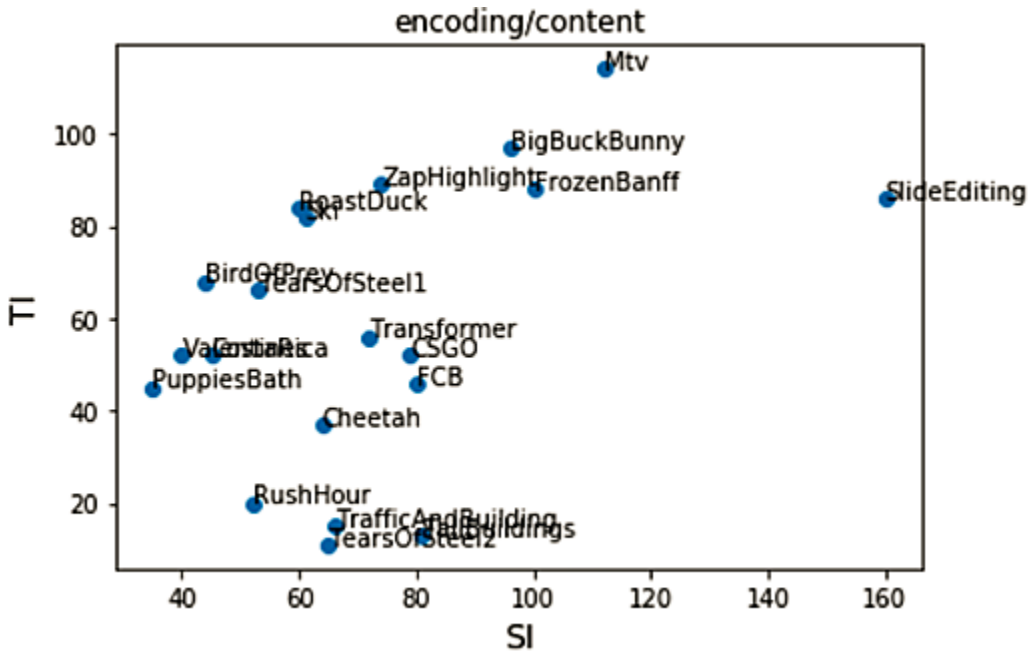


Figure 3-1:SI/TI for videos in Waterloo Database

in these videos are presented and explained next [48]. The detailed specification of these videos' contents is presented in Table 3-1. The spatial information (SI) and temporal information (TI) that reflect the complexity of video content are also shown in [46]**.** For example, in videos with static scenes and smooth motion such as CostaRica, PuppiesBath the SI and TI are both low. Meanwhile in high motion like sports videos as Ski movie, the SI is low while TI is high. Additionally, in Animation videos with high action the SI and TI are both high as BigBuckBunny. The video sequences are of diverse spatio-temporal complexity and widely span the SI-TI space. Using the sequences as the source, each video was encoded with an x264 encoder into eleven representations using the encoding ladder shown in Table 3-2 to cover different quality levels. The test sequences were segmented with a segment length of 2 seconds [48].

Table 3-1: Specification of the videos in Waterloo III Database

Name	FPS	SI	TI	Description
BigBuckBuny	30	96	97	Animation, high motion
BirdOfPery	30	44	68	Natural scene, smooth motion
Cheetah	25	64	37	Animal, camera motion
CostaRica	25	45	52	Natural scene, smooth motion
CSGO	60	70	52	Game, average motion
FCB	30	80	46	Sports, average motion
FrozenBanff	24	100	88	Natural scene, smooth motion
MTv	25	112	114	Human, average motion
PuppiesBath	24	35	45	Animal, smooth motion
RoastDuck	30	60	84	Food, smooth motion
RushHour	30	52	20	Human, smooth motion
Ski	30	61	82	Sport, high motion
SlidingEditing	25	160	86	Screen content, smooth motion
TallBuildings	30	81	13	Architecture, static
TearofSteel1	24	53	66	Movie, smooth motion
TearofSteel2	24	56	11	Movie, static
TrafficAndBuilding	30	66	15	Architecture, static
Transformer	24	72	56	Movie, Average motion
Valentines	24	40	52	Human, smooth motion
ZapHighlight	25	97	89	Animation, high motion

A total of 34 subjects assessed the videos. In the subjective testing (full details about subjective testing in [47]), the single-stimulus methodology was selected in which the reference/source videos are also evaluated in the same experimental session with the test streaming videos. A 100-point continuous scale was chosen because of its expanded range, and its fine distinctions between rating and its prior efficiency as opposed to a discrete 5-point ITU-R Absolute Category Scale (ACR). The raw subjective scores are converted to Z-scores per session to account for any differences in the use of the quality scale between sessions. After outlier removal, Z-scores are linearly rescaled to lie in the range of [0, 100]. The final quality score for each individual video is computed as the average of rescaled Z-scores, namely the mean opinion score (MOS),

from all valid subjects [48]. Figure 3-2 plots the histogram of MOS scores across distorted videos for the subjective study. The figure demonstrates that the distorted videos covers most of the quality range. The average standard deviation of opinion score in the MOS were 19.

Table 3-2: Encoding Ladder of Video sequences

Representation index	Resolution	Bitrate(kbps)
1	320x240	235
2	384x288	375
3	512x38	560
4	512x3844	750
5	640x480	1050
6	720x480	1750
7	1280x720	2350
8	1280x720	3000
9	1920x1080	4300
10	1920x1080	5800
11	1920x1080	7000

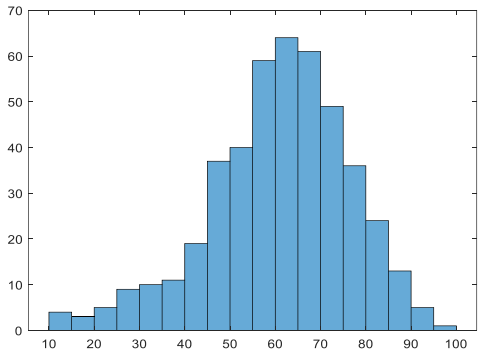


Figure 3-2: MOS statistics of Waterloo SQoE-III database

Figure 3-3(a)-(f), illustrates the 13 network traces used in the study. Figure 3-3 (a) shows the scenario where network bandwidth is stable with no change and in (b) the bandwidth is continuously increasing or Ramp up, while in (c) the bandwidth is continuously decreasing or Ramp down, and in (d), (e), the bandwidth is

changing one step either up or down. Finally, in (f) the bandwidth keeps fluctuating up and down. They are of wide-range and represent stationary as well as different scenarios indexed from the lowest to the highest average bandwidth. Where I and XIII represent the cases with the lowest and highest average bandwidth of the network traces that varies between 200Kbps and 7.2Mbps, covering all range of bitrates in the encoding bitrate ladder [48].

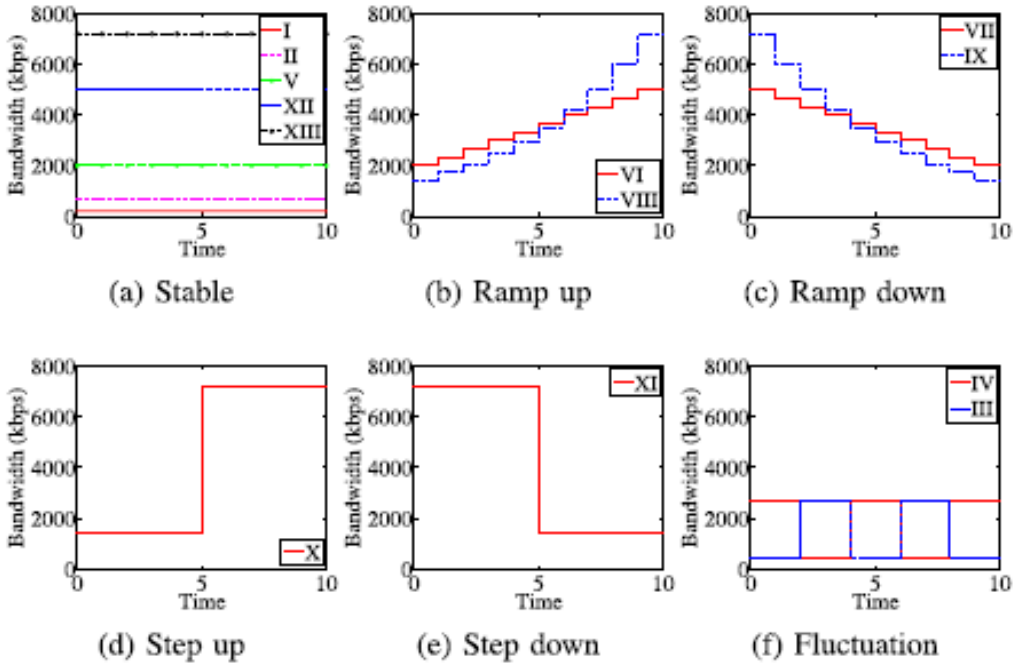
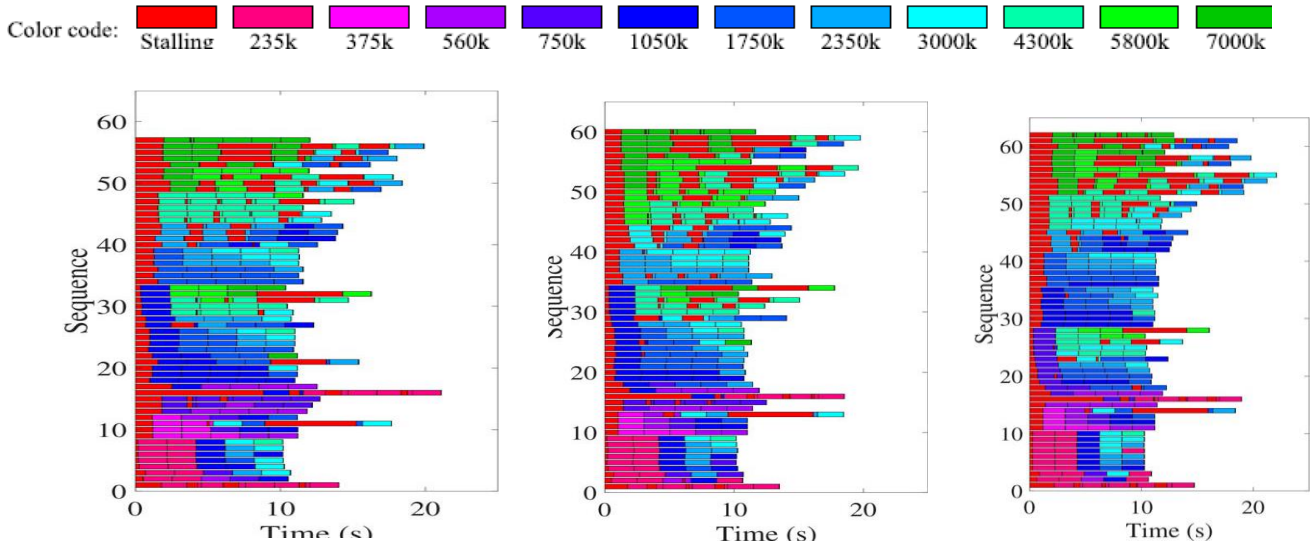


Figure 3-3:Network Traces used in the study

Figure 3-4 shows an example of the distortion profiles of the streaming video sequences used in the subjective study. Each subfigure represents a streaming video which was generated by one or multiple bitrate adaptation algorithms under one network trace [48].



(a) BigBukBunny **(b) Birdofprey** **(c) FCB**

Figure 3-4: Distortion profiles of the streaming video sequences in the subjective study

Dataset Distribution

In this section, we present the data analysis of the dataset to gain some insight about the relationship between the different features (startup delay, stalls, quality switches) and the MOS. As existing databases suffer from including patterns that reflect realistic scenarios as mentioned in Chapter 2, specifically the Waterloo Streaming QoE-I database [30] that comprises videos with only one stalling event either at the beginning or the middle point, such simplification makes the analysis of human QoE behavior easier. However, these hand-crafted distortion patterns can hardly represent distortions in the real adaptive streaming scenario that are dependent on the behavior of the adaptive bitrate algorithms (ABR) algorithms. In addition, the distortion types of video sequences are isolated, such that spatial resolution adaptation which is very important and commonly used in practice is not presented; In addition, the network information and bitstream [48], which are valuable to develop an ABR algorithms and objective QoE models, are not available. Along this direction, we make some statistical analysis on the distribution of the attributes/factors that this database reflects to make sure that it contains realistic scenarios.

Figures 3-5 to 3-8 present a brief characterization of the studied dataset. Figure 3-5 presents the results of data interpolation to convert MOS from (1-100 to 1-5) range. The MOS scores range from 1.5 to 4.9, with about 20% of the videos rated as of poor quality (i.e., MOS < 3), 40% rated as good quality (i.e., 3.5 < MOS < 4), and 20% rated as excellent quality (i.e., MOS > 4.5).

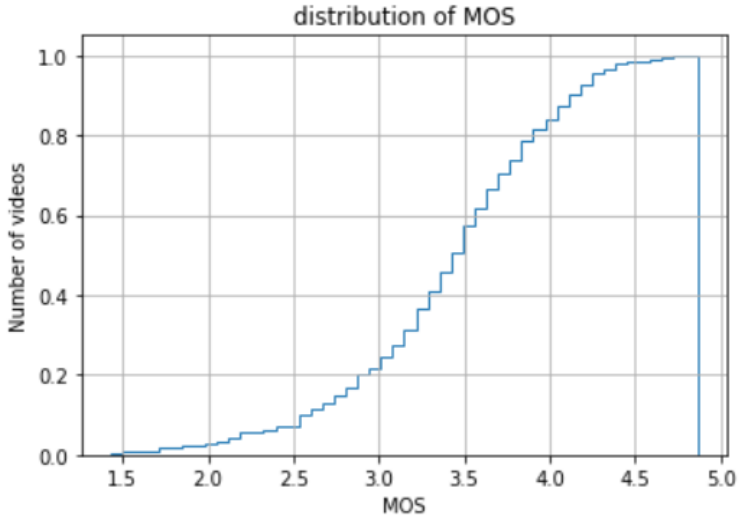


Figure 3-5:Distribution of MOS

As shown in Figure 3-6, more than 30% of the tested conditions have four stalling/rebuffering events and approximately 50% of the video conditions consider perfect quality, with one stalling or without stalling. Roughly 15% are having three stalling events, and 10% with two stalling events are clearly visible, with 55% of the videos having 0s stall duration. Almost 45% of the videos having an average stalling duration between 0.1 and 5s, and 5% between 5 to 7s as shown in Figure 3-7.

We use the terms stalling and rebuffering interchangeably in this thesis to refer to the event when there is no data in the buffer, hence the video playback is stalled (frame freeze occurs) [48]. Stall/rebuffer duration refers to the combined length of all stall events in a single video session.

A video session indicates the audiovisual playback from start till the end of the video and includes the effects of initial loading time, rebuffering events and quality switching if any. Hence, in presence of any of these events the video session length will be longer than that of total video playback length [48].

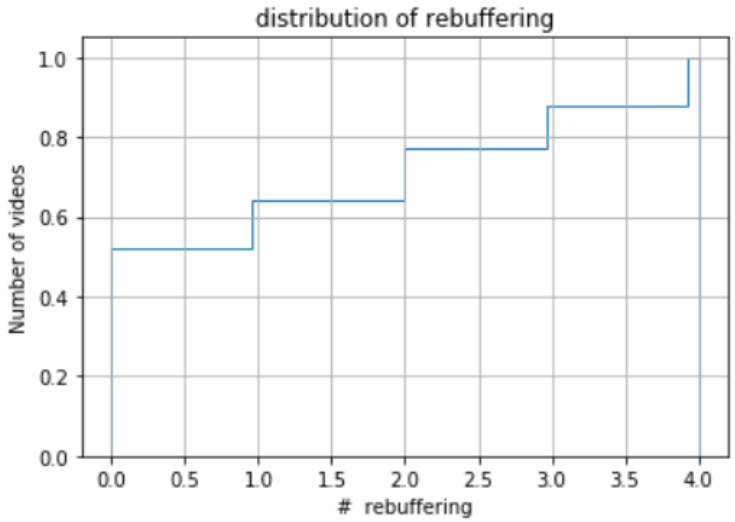


Figure 3-6:Distribution of number of rebuffering events

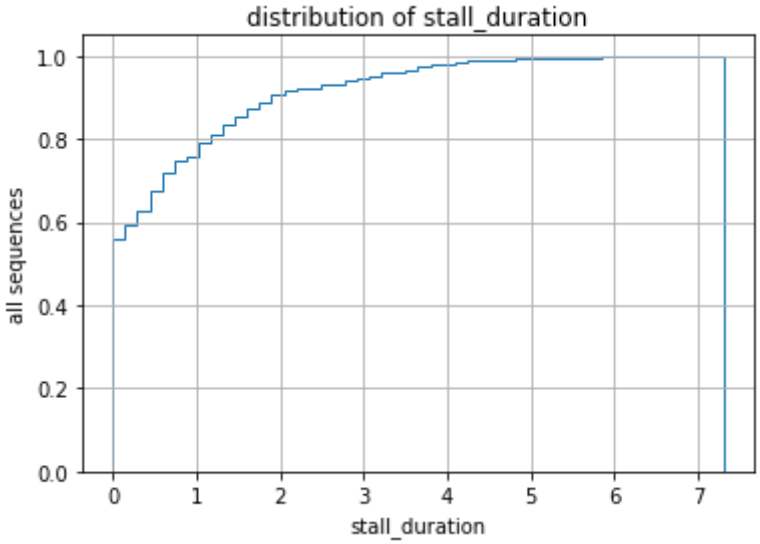


Figure 3-7:Distribution of stall duration

More than 30% of the tested conditions have four switches, and about 10% of the video conditions consider perfect quality, with one or without switches. About 20% are having two switches and about 40% of the videos having three switches, as shown in Figure 3-8.

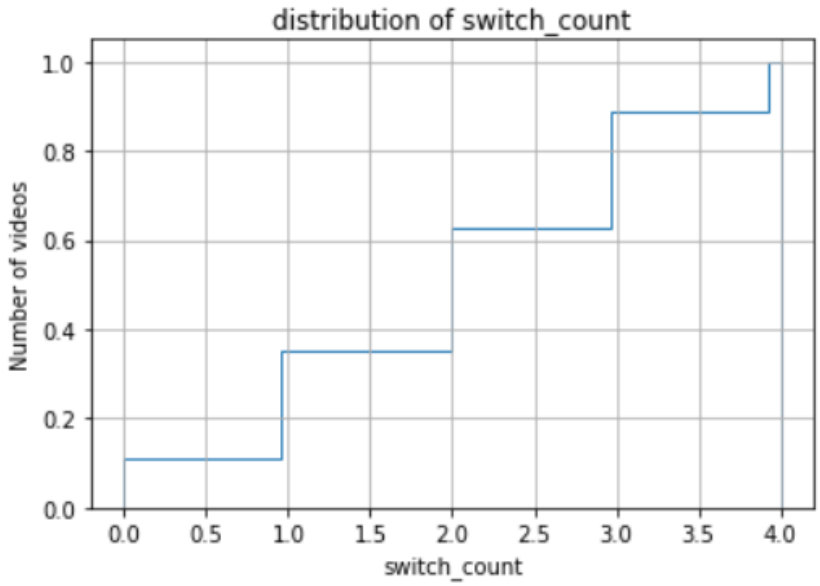


Figure 3-8:Distribution of number of switches

Quality switching is referred to as the rate or quality or bitrate adaptation, and it is the change of quality over the duration of the video playback [48].

3.2 Quality Influence Features

The following are the features of the dataset and how they are computed:

- ***Initial buffer time/delay (Ti):*** this metric represents the elapsed time starting from the time that the player establishes a connection to a video server until the time that enough player video buffer is filled up and the player starts receives video frames. It is measured in seconds. It is the waiting times in the beginning to pre-download the video before watching [48] [46].
- ***Rebuffer percentage (P_r):*** This metric is an aggregate metric that can reflect duration of a long video "stalls" observed by the user. This metric is the fraction of the total session time (i.e., playing plus rebuffer time) spent in buffering. It is computed as [46]:

$$\sum_i \frac{duration\ of\ rebuffer\ event\ i}{session\ duration} \qquad (3.1)$$

- ***Rebuffer count (C_r):***. In a video session that experiences "video stall" where each stall time is small but the total number of stalls is high, may not have a high buffering ratio but may be just as

annoying to a user [46]. *As* Rebuffer percentage metric does not capture the number of induced interruptions observed by a user. We need to capture frequency of stall using this metric.

- ***Stall duration/total duration of rebuffering:*** Length of all rebuffering events in a single media session [46] [48].
- ***Quality switch count (C_s):*** The number of quality switches is a metric that describes the variance of the session. High values indicate very frequent switching which can lead to a decreased QoE. This is calculated as follows [46] [48] (we will be using bitrate and quality interchangeably throughout this thesis):

$$\mathbf{g}(s_i) = \begin{cases} \mathbf{1} \;\; if \; i = 0 \\ \mathbf{1} \;\; if \; f(s_{i-1}) \neq f(s_i) \\ \mathbf{0} \;\; else \end{cases} \qquad \textbf{(3.2)}$$

- ***Average bitrate/quality switch magnitude ($\overline{B}_s$):*** This metric was also identified as an influencing factor of flicker effect. people likes multiple switches with smaller bitrate differences to abrupt quality variation. It is Measured in kilobytes per switch, It is computed as [46] [48]

$$\sum_{i=2}^{n} \frac{|bitrate_i - bitrate_{i-1}|}{\#\ of\ switches} \qquad \textbf{(3.3)}$$

where n is the number of segments.

- ***Ratio on the highest video quality layer (P_h):*** This metric measures the percentage of time spent on the highest quality layer. Previous studies have argued that the effect of quality switch count is negligible compared to the percent of time on the highest quality layer [49], [50].
- ***Frame per second (fps):*** video frames per second

Table 3-3 summarizes the features we have used in QoE prediction that are computed from Waterloo database.

Table 3-3 Features used in QoE prediction

Feature	Description
Initial buffer time/Delay (Ti):	Time duration between the request for video playback by the client and the actual start of the video playback.
Rebuffer percentage (P_r):	Rebuffering refers to the event when there is no data in buffer.
Rebuffer count (C_r):	Frequency of rebuffering refers to the number of rebuffering events per unit of time
quality switch count (C_s):	Refers to the change of bitrate/quality over the duration of the video playback.
Average quality switch magnitude ($\bar{B}_s$):	Refers to the "gap" between the levels of quality switching.
Ratio on the highest video quality layer (P_h):	Time on the highest layer indicates the percentage of time the media playback is at the highest quality.
FPS	Frame per second of all videos.
Stall duration	The length of all rebuffering events in a single media session.
MOS	The average video MOS score.

3.3 **Correlation and Data Analysis**

In this section, we will present the data and correlation analysis that was applied to the dataset to study the effect of each feature on Mean Opinion Scores (MOS).

We evaluate the performance of each feature using the Spearman correlation coefficient (SRCC) and Pearson correlation coefficient (PLCC).

The PLCC is used to measure the linear association strength between two variables. Given paired data $\{\{x_1, y_1\}, \ldots, \{x_n, y_n\}\}$, PLCC computed by the following: [51]

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad (3.4)$$

where:

- n is the sample size
- x_i and y_i are individual sample points.
- $\bar{x}$ is the first sample mean defined as: $\bar{x} = \frac{1}{n}\sum_{i-1}^{n} x_i$
- $\bar{y}$ is the second sample mean defined as: $\bar{y} = \frac{1}{n}\sum_{i-1}^{n} y_i$

The coefficient is calculated by dividing the covariance of two variables by the product of their standard deviations. A correlation coefficient value of 1 means a perfect positive correlation, while a value of -1 means a perfect negative correlation. A value of 0 means there is no correlation at all.

The SRCC is used to measure the strength and direction of monotonic association between two variables in a single value between -1 and +1. It uses ranks instead of assumptions about the distributions of the two variables ranking (from low to high) is obtained by assigning a rank of 1 to the lowest value, 2 to the next lowest and so on. SRCC is computed by the following [51].

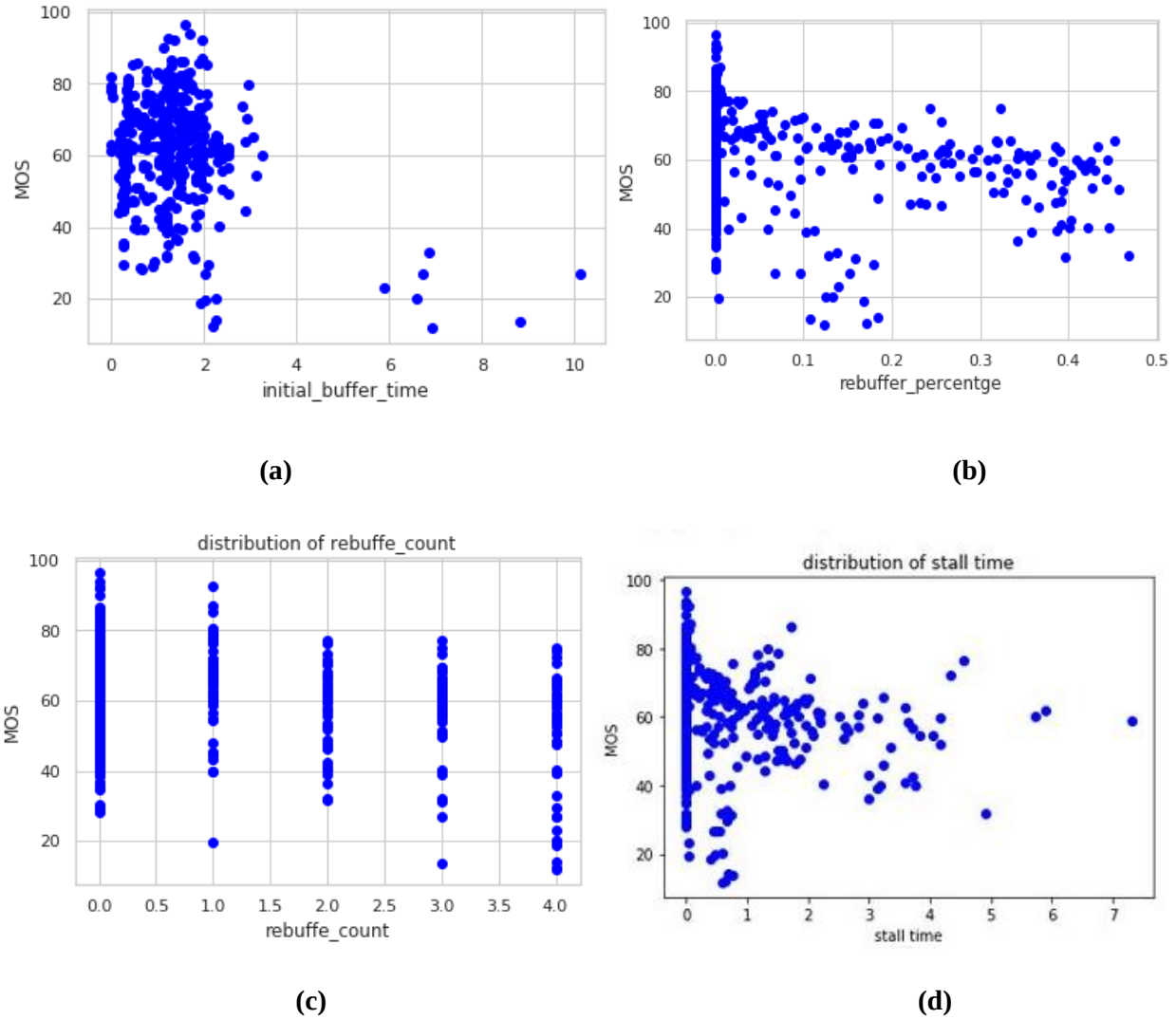
$$\rho = \frac{\sum_{i=1}^{n}(R(x_i)-\overline{R(x)}).(R((y_i)-\overline{R(y)})}{\sqrt{(\sum_{i=1}^{n}(R(x_i)-\overline{R(x)}^2)\,).\sqrt{(\sum_{i=1}^{n}(R(y_i)-\overline{R(y)}^2).}}} \quad (3.5)$$

where:

- R(x) and R(y) are the ranks of the observation in the sample
- $\overline{R(x)}$ and $\overline{R(y)}$are the mean ranks.

The scatter plots of the above-mentioned quality features versus the MOS, are presented in 2 parts. Part (1) is shown in Figure 3-9(a)-(d) and part (2) is shown in Figure 3-10 (e) –(h). For initial rebuffer_time in Figure 3-9 (a), we can see that it has low impact on QoE as users tend to be more patient to longer initial delays than other video distortions such as stalls and quality changes. Figure 3-9 (b, c, d) highlights the relationships between the MOS scores and the number and duration of rebuffering events respectively. It can be observed that a larger number of rebuffering events tends to decrease user experience. In Figure 3-9 (c), one can see that in certain video sessions that there are 2, 3 and 4 rebuffering events but the average duration is not in decreasing MOS order. In this case, the average rebuffering duration was 3.2,2.56 and 2.3

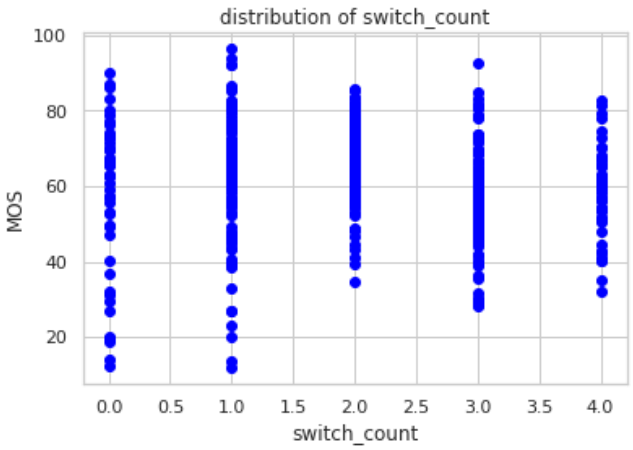
sec. This means that larger rebuffering occurrence not necessary results in larger rebuffering duration, but users are sensitive to combined effect of rebuffering occurrences and duration. As shown in Figure 3-9(b) that longer rebuffer duration lowers QoE. But when rebuffering time is more than two seconds, duration neglect effect [52] appear that may reduce this effect, according to this duration neglect aspect, users may remember the duration of an impairment, but they become insensitive to its duration after certain time when making subjective QoE evaluation. For the quality switch magnitude in Figure 3-10 (e) and (f), users prefer multiple switches with small quality switch difference than sudden quality switches. In Figure 3-10 (h), when user stays more time on highest quality level the MOS values increase.



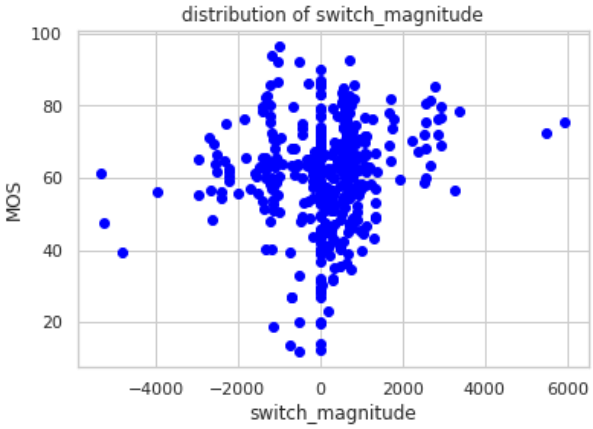
(a) **(b)**

(c) **(d)**

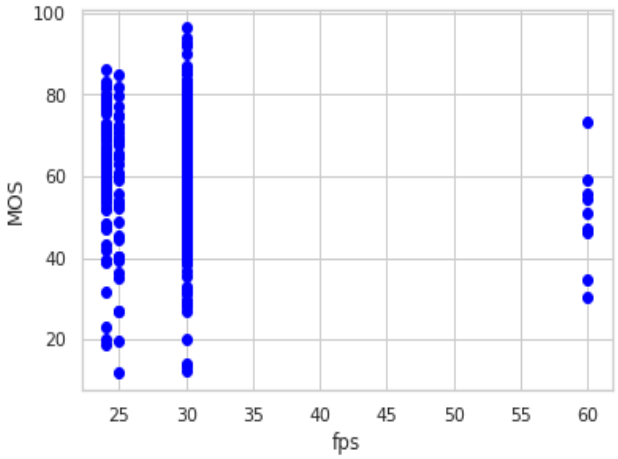
Figure 3-9:Quality features versus MOS (part 1)



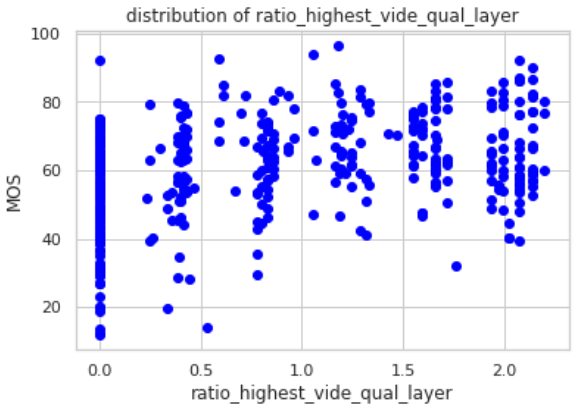
(e)



(f)



(g)



(h)

Figure 3-10:Quality features versus MOS (part2)

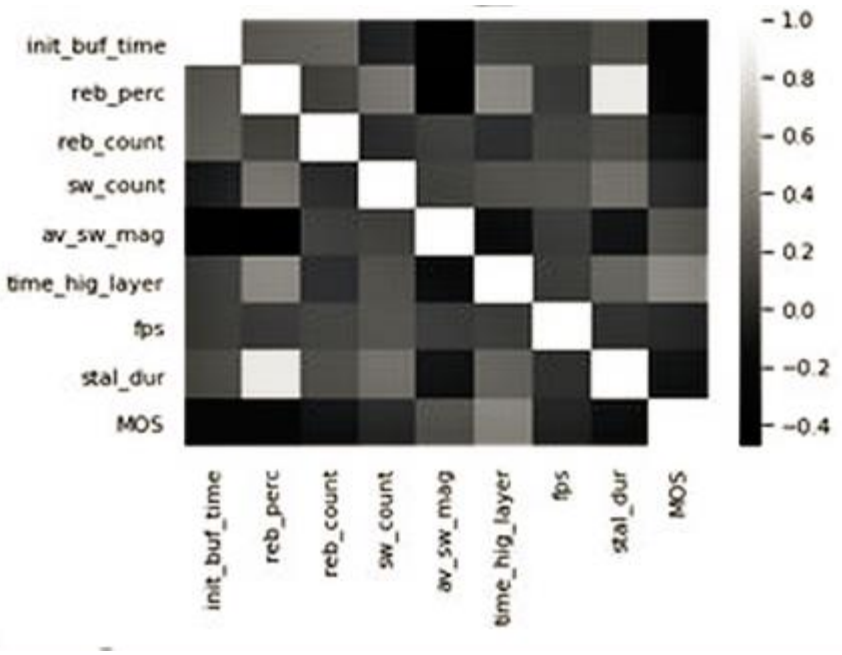
Table 3-4 SRCC and PLCC between features and MOS

Quality Metric	SRCC	PLCC
Initial_buffer_time	-0.029	0.269
Rebuffer_percentge	**-0.266**	**-0.269**
Rebuffer_count	**-0.136**	**-0.183**
quality_switch_count	**-0.164**	**-0.076**
Average_switch_mag	**0.140**	**0.113**
ratio_highest_video_qual_layer	**0.355**	**0.369**
stall_duration	**-0.186**	**-0.328**
Fps	**0.002**	**-0.072**

Table 3-4 shows that the time spent on the highest video quality layer reveals noticeably moderate correlation with MOS. This result agree with previous studies [49] [50] that the time spent on the highest video quality layer is an important influence factor on QoE. The explanation is that the number of quality difference between successive quality layers in this study are relatively large, as shown in Table 3.2, that videos are encoded into 11 quality levels/layers. In our study, we consider layers 5 till 11 as higher layers. As the difference between consecutive quality layers are significantly large, this allow us to consider more layers in our computation without throwing away important information about the reaming quality layers. The quality switch count and average equality switch magnitude have relatively little impact on MOS. For the rebuffer_percentage and count, the correlation is small.

We have compared the performance of each metric separately and the result of correlation measured by SRCC and PLCC presented in Table 3-4 suggests that none of them, by itself is sufficient to predict QoE accurately. Figure 3-11 shows the correlation matrix between different features and the MOS to get insights about the relationship between the different features amongst each other. It can be seen that most pairs are quite independent of each other, which indicates that the used features may supplement each other, and their combined use may provide better performance. The correlation between ratio_on_highest quality layer

has moderate correlation with MOS. The rebuffer_percentage have moderate correlation with stall duration which is expected as they are dependent on each other.



.

Figure 3-11:correlation matrix between Quality Features and MOS

To summarize this section, we presented Waterloo III database and the different Quality influence features which reflect the impairments that interrupt video during the video playback. There is another type of impairments of videos that occur while video being transmitted and encoded. In the next section, we discuss these impairments.

3.4 Objective Video Quality Assessment

In a classic adaptive bitrate video streaming scenario, an adaptive bitrate allocation strategy is applied to optimize bandwidth consumption. These adaptive bitrate schemes lead to compression and blocking artifacts [53], commonly introduced during video encoding, and are one of the major causes of reduced perceptual video quality. Other network-related distortions arise from packet losses [54] or impairments of

the source videos such as blurring, and scaling artifacts. These type of impairments are common in that there are no implied playback interruptions. To help measure the video quality degradations induced by these video distortions, a variety of video quality assessment (VQA) algorithms/models have been proposed which include full reference (FR) video quality model such as Peak Signal-to-Noise Ratio (PSNR) [12], Structural Similarity Index metric (SSIM) [55] and Multi-Scale Structural Similarity Index (MS-SSIM) [56] , temporal full reference (FR) models such as Video Quality Metric (VQM) [14], Video Multimethod Assessment Fusion (VMAF) [57], and reduced-reference models like Spatio-Temporal Reduced Reference (STRRED) [58].

Previous work done on QoE prediction studied the effect of each type of impairment separately. Since most VQA models do not consider the effect of playback interruption that occurs in video session (rebuffering, quality switching, intial_buffering, etc), it is important to understand whether they can be applied to streaming videos and test their performance in case of video streaming. In this regard, a number of objective quality metrics including PSNR, SSIM [55], MS-SSIM [56], STRRED [58], VQM [14], VMAF [57], SSIMplus [32], and Video Intrinsic Integrity and Distortion evaluation Oracle model (VIIDEO) [59] are evaluated against human subjective scores measured as MOS. These metrics are considered important and correlate well with human subjective opinion (MOS). The process of testing performance of these metrics is performed by dividing the video data into two sets, A and B. The first set, $Set_{A,}$ is the source videos with no impairment in a normal playback while the second set, Set_B, is the entire Waterloo streaming videos with impairments. The correlation analysis is discussed next.

3.4.1 Correlation Analysis between Objective VQA model against MOS

The Waterloo dataset contains the implementations of these VQA models which was obtained from the original authors. Spearman's correlation coefficient (SRCC) is used for performance evaluation by comparing MOS and objective QoE scores. Table 3-5 presents the result of the correlation.

Table 3-5 Result of correlation between VQA models and MOS

VQA model	SRCC	
	Set_A	Set_B
PSNR	0.667	0.460
SSIM [55]	0.7448	0.524
MS-SSIM [56]	0.743	0.5217
SSIMplus [60]	0.8298	0.5617
VQM [57]	0.8192	0.5650
STRRED [59]	0.6760	0.488
VMAF [58]	0.7977	0.5613
VIDEO [61]	0.4388	0.3506

Clearly, there is a drop in the correlation from set A to set B for all metrics. This suggests a performance degradation of the objective quality models for videos affected by playback impairments, which is expected. However, in any streaming application, such impairments often occur, therefore the need to combine playback interruption impairment features with objective quality features to obtain a unified QoE-aware information to improve the predictive power of QoE prediction model. Therefore, we integrate the objective video quality features with the features extracted from Waterloo dataset that measures playback interruption features namely, initial_buffering, rebuffering duration and frequency, and quality switch number, and magnitude (introduced in Section 3.2) to significantly improve the QoE prediction.

Figure.3-12 shows the scatter plots of the above-mentioned quality features versus the MOS scores. The figure shows that SSIMplus and VQM are the best performing models. While SSIM performed similar to MS_SSIM, while VIIDEO has the worst performance.

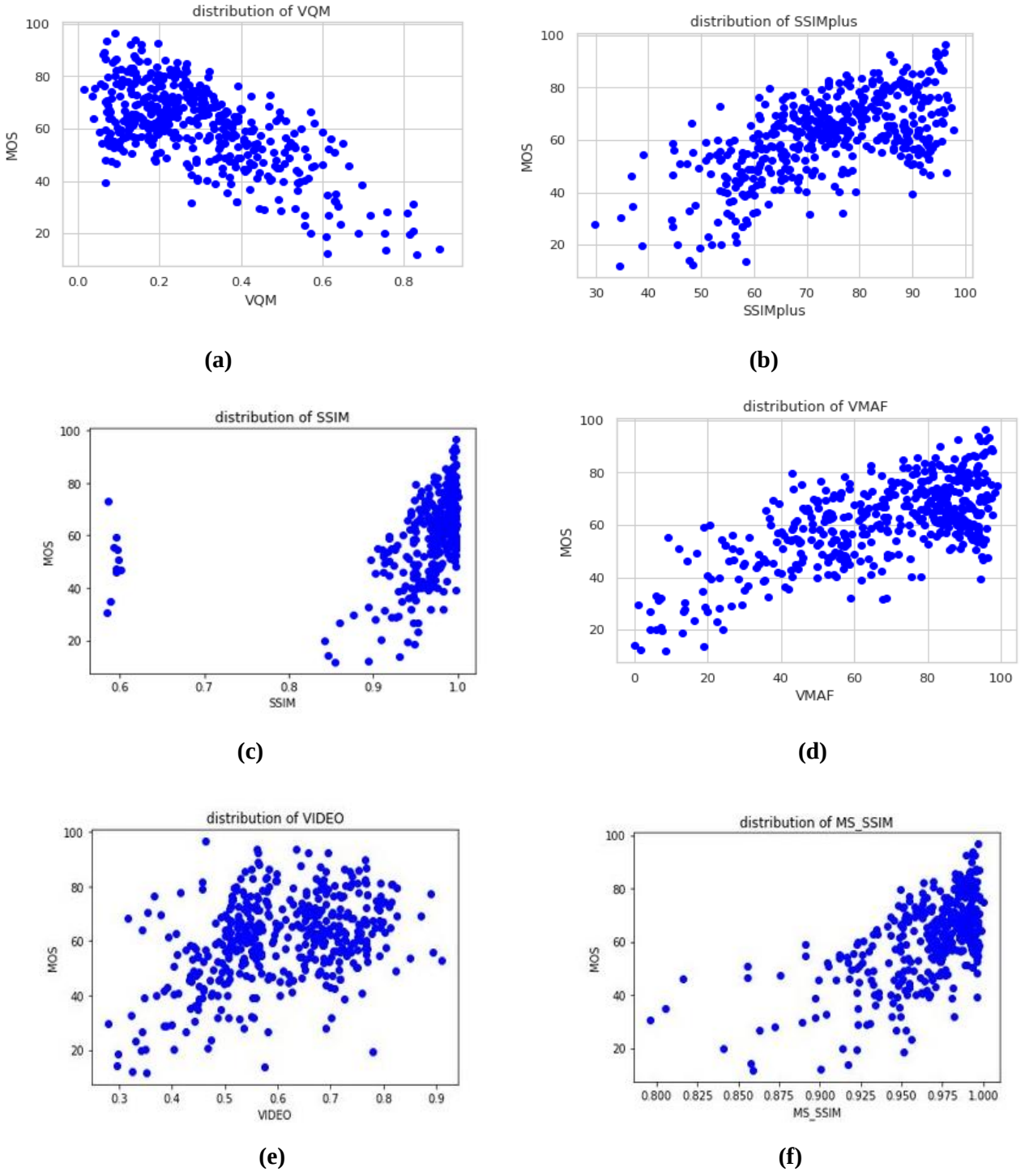


Figure 3-12:correlation between VQA models against MOS

We also plotted the correlation matrix between all features combined with SSIMplus as our VQA metric against the MOS to get insights about relationship between VQA and our features as shown in Figure 3-13. It is clear that there is correlation between reb_per, stall duration and SSIMplus. Additionally, there is

correlation between ratio_on_highest quality layer and SSIMplus. This correlation map results shows there is dependency between VQA metric namely SSIMplus in our case and stalling features. The same hold for correlation between VQA and ratio_on_higher_layer.

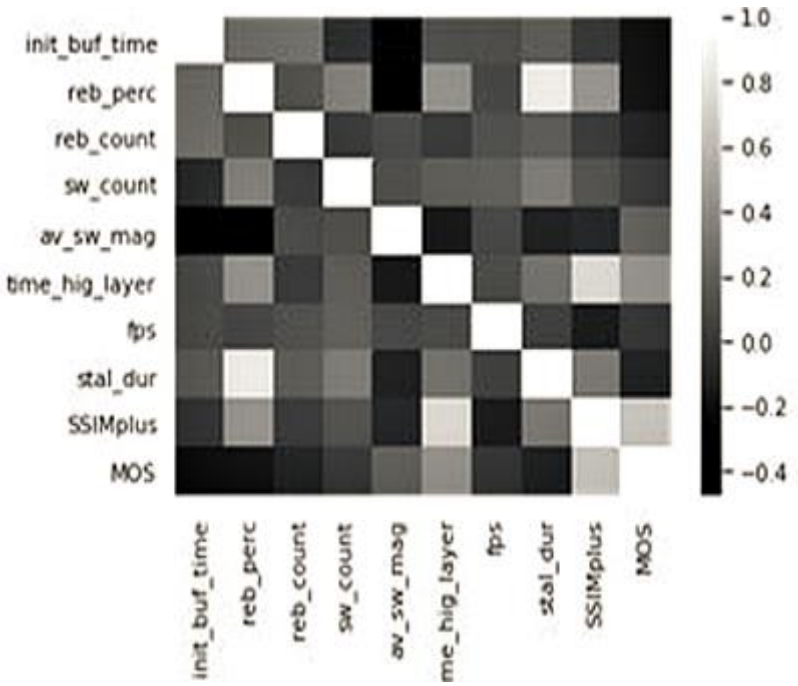


Figure 3-13:correlation matrix between Quality features combined with SSIMplus against MOS

3.5 **Proposed Feature Enhancement Scheme**

The work in this thesis is based on combining VQA models features with quality influence features to perform the QoE prediction using different machine learning techniques. Our proposed feature enhancement scheme is shown in Figure 3-14.

The proposed Feature Enhancement Scheme is based on combination of QoE-related features extracted from Waterloo SQoE (III) video dataset such as detection of streaming stalling, and quality switching, combined with video quality metrics that measures compression artifacts resulted from the video encoding process such as blocking, blurring or ringing, and is computed on frame-by-frame level with different

pooling techniques are used to collapse frame score into a final score. We presented examples of existing pooling techniques in Chapter 2, such as averaging, exponential weighting, etc. The average pooling is the methods used in all VQA algorithms in this work. Examples of these metrics includes: SSIMplus [55], VMAF [57], STRRED [58].

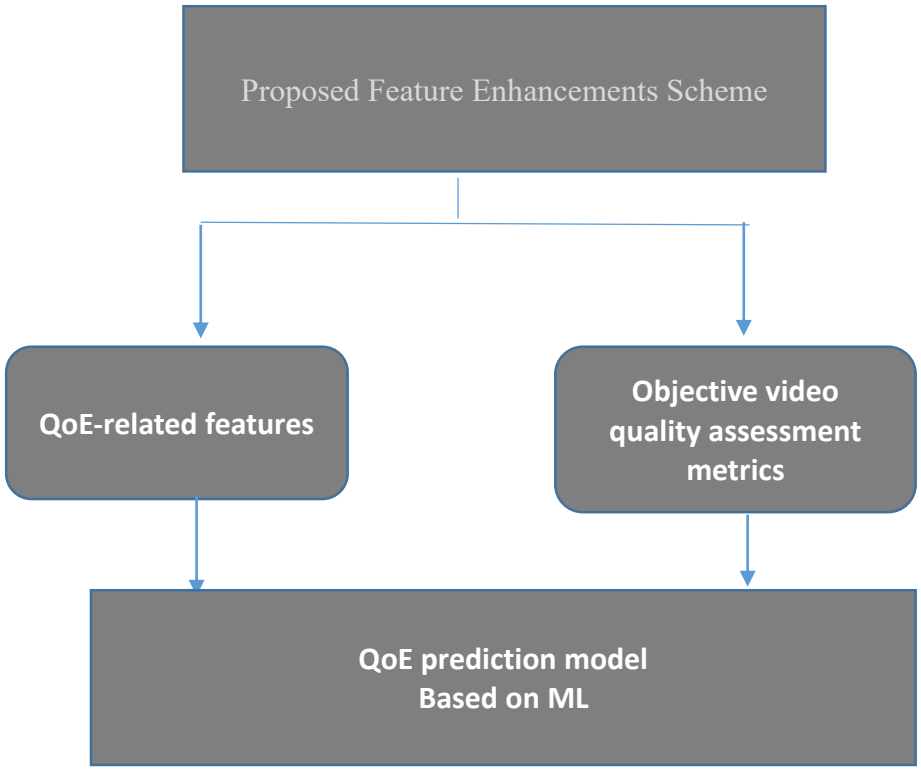


Figure 3-14:Proposed Feature Enhancement Scheme

In our first experiment, we apply linear regression model to investigate the effect of integrating VMAF as our VQA metric with the QoE features. To do this, the four impairment features with highest correlation with the MOS score are picked up as the independent parameters of the regression model. In Figure 3-15. one can observe the importance and strength of correlation of each feature with the MOS.

The regression model is assessed using Spearman's Rank Correlation (SRCC) and Pearson Linear Correlation Coefficient (PLCC). The bottom part of Table 3.6 shows the results of linear regression

prediction model using only the QoE features. The top part of the table shows the result when the VMAF is added as the objective models (VQA) to the QoE features.

The results in the table were produced as follows. The dataset is split into 80% training and 20% testing subsets, and we apply linear regression on the training subset and the resulting model is tested using the test-subset. To avoid any bias due to data division we repeated the process 100 times. We computed the SRCC between the predicted and actual quality scores at end of each iteration and took median of SRCC scores. Rather than showing all combinations, we include the first 4 regression models with highest correlation as shown in Figure 3-15 with VMAF as our objective model.

Table 3-6: Result of Linear Regression

Regression model	SRCC	PLCC
61.55-1.026 Ti +0.0004C_s -8.187P_r- 0.0038P_h+12.758VMAF	0.815	0.769

Regression Model	SRCC	PLCC
61.57-2.05 Ti -0.33C_s -6.738P_r+ 7.224P_h	0.398	0.301

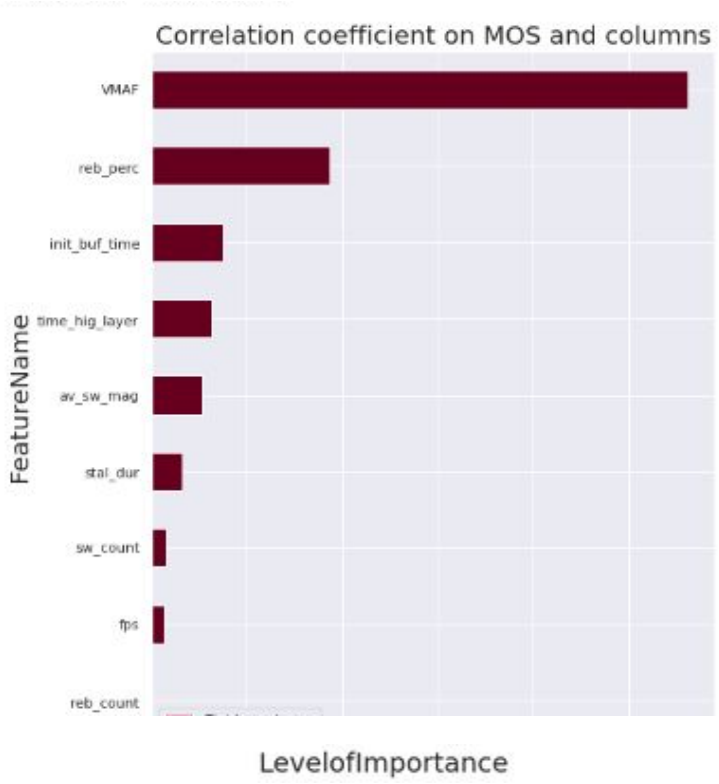


Figure 3-15:Correlation coefficient between all features combined with VMAF against MOS

From result of linear regression, we can see that there is great improvement in the monotonicity and linearity when combining all features. The detailed results of our experiments is presented in Chapter 5.

3.6 Summary

In this chapter, we described the dataset attributes and its characterization is explored. Also, we did correlation analysis between the different features and MOS, in addition, we performed correlation analysis between MOS and VQA model. Then we presented our proposed feature enhancement Scheme. In the next chapter, we present the data preprocessing steps that we have done and discuss the process of QoE prediction using different machine learning

Chapter 4

Modeling and Prediction Methods

In Chapter 3, we presented our methodology in extracting the different set features from the videos in the dataset. These features will be applied as the independent input variables to any prediction model. In this chapter, we will use different machine learning prediction models to calculate the Quality of Experience (QoE) as a predicted variable.

In this chapter, we explain the steps that were carried out to apply the machine learning models on the Waterloo dataset for Quality of Experience (QoE) prediction. In Section 4.1, we present the steps to prepare the data to be processed by the machine learning models. In Section 4.2, we discuss the different techniques used for feature scaling and selection. In Section 4.3, we present the various machine learning models that we applied for QoE prediction. In Section 4.4, the hyperparameters tuning process for the different models is outlined. In Section 4.5, we describe different techniques for data splitting. Finally, in Section 4.6, we present the evaluation metrics that were used to assess the performance of the models.

4.1 Data Preprocessing

Raw data is often noisy and incomplete; therefore, machine learning models cannot be applied directly on raw data. To ensure the accuracy and efficiency of the machine learning models, some preprocessing steps need to be performed before the data can be fed to the models. In the next subsections, the preprocessing steps used in this work are presented.

4.1.1 Feature Scaling

Feature scaling is often required in machine learning when the data features have different ranges, especially when tested models rely on the distance between the features. Therefore, features with larger ranges would influence the result more than those with smaller ranges. The following are a few more commonly used scaling techniques:

1- Mean Normalization

Mean normalization causes the data to have values between -1 and 1 and mean of 0. Mean normalization can be performed using the following formula:

$$x' = \frac{x - mean(x)}{\max(x) - \min(x)} \quad \textbf{(4.1)}$$

To perform mean normalization, it is necessary to know the maximum and minimum values of the data.

2- Min-Max Scaling

Min-Max scaling can also be used to scale data features. In Min-Max scaling, the features are scaled between 0 and 1 [60]. The formula for Min-Max scaling is given by the following equation:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad \textbf{(4.2)}$$

3- Standard Scaling (Z-score)

The Standard scaling method standardizes the features by removing the mean and scaling to a unit variance value. The standard score of a sample x is calculated as:

$$x_{new} = \frac{x - \mu}{\sigma} \quad \textbf{(4.3)}$$

where μ and σ are the mean and standard deviation of the training samples. Centering and scaling happen independently on each feature by computing the relevant statistics on the samples in the training set.

In this work, the standard scaling method was used.

4.2 Feature Selection

Feature selection is the process of selecting the features that contribute the most to the prediction variable. It is an important step in the machine learning pipeline as training a model with irrelevant features could decrease the accuracy of the model and result in erroneous predictions. Moreover, training the model with fewer attributes reduces the complexity of the model, and makes the model simpler and easier to understand. In the following subsections, three commonly used methods for feature selection are discussed.

4.2.1 Univariate Selection

This method performs statistical tests that indicate which features have the strongest relationship with the target variable. The Chi-square test is one example of these statistical tests and is often used to test the association between two variables [61]. Basically, the Chi-square between each feature and the target is calculated, and then features with best Chi-square scores are selected.

The Chi-square is calculated using the following equation:

$$x_c^2 = \sum \frac{(O_i - E_i)^2}{E_i} \quad \textbf{(4.4)}$$

where c is the degree of freedom which represents the number of values in the final calculation that are free to vary, O is the observed value and E is the expected value.

When a feature has a high correlation with the target value, the observed count is far from the expected count. Therefore, the features will have a high Chi-square value. Having a high Chi-square value means that the target value is dependent on the feature, and thus the feature can be selected for model training.

4.2.2 Feature Importance

The importance of each feature in the dataset can be computed using the feature importance property of the model. The idea of feature importance is to calculate a certain score for each feature in the dataset, where higher scores indicate that this feature contributes more toward the prediction of the target variable. Two common techniques that are used for computing feature importance are model specific feature importance and permutation feature importance.

1- Model Specific Feature Importance

This technique is implemented by the Random Forest algorithm. By using this technique, the average reduction in impurity caused by each feature is calculated across all the trees in the forest. Furthermore, the features that cause the nodes to split closer to the root will have a higher feature importance value [62].

2- Permutation Feature Importance

Permutation feature importance measures the importance of a certain feature by permuting that feature and then measuring the mean decrease in performance. A feature has a high importance value when the model's

prediction performance decreases and when the feature's values are permuted, meaning that the feature contributed to the model's prediction. Along with the feature importance estimate, this technique also gives a measure of uncertainty of that estimate. On the contrary, the main disadvantage of the permutation feature importance is that it becomes computationally expensive when the feature space increases.

4.2.3 Correlation Matrix with Heatmap

Correlation is a statistical term that measures the association between variables [63]. A positive correlation between two variables means that when one variable increases, the other increases as well. On the contrary, a negative correlation means that when one variable increases, the other decreases.

Correlation analysis can help with feature selection, as a feature that has a high correlation with the target variable can be selected as input to the model. Furthermore, features with high correlation with each other can be linearly dependent, and therefore have a similar effect on the target variable. In that case, one of the two features can be dropped.

We used the univariate feature selection method to select the best k-features regarding the strength of its relationship with target variable. We found that the most important features for QoE prediction are the init_buf_time, reb_perc, time_hig_layer, av_sw_mag, stall_dur, sw_count combined with any objective video quality assessment (VQA) metrics presented in Chapter 3. Therefore, we only used these variables as input to machine learning model and disregarded the remaining data variables.

4.3 Models

In order to demonstrate the behavior of the regression process, we evaluated several types of regression models [43]: linear models (Ridge and Lasso regression), Support Vector regression (SVR) using RBF kernel and ensemble methods such as Random Forest (RF). For the ensemble method, feature normalization was not required, but we preprocessed the features for all regression models using standard scaling only on the training data. For each of the regression models, we determined the best parameters using 10-fold cross validation method with the training set. This process was repeated on all possible train/test splits.

To determine the best model that could be used for QoE prediction on this dataset, we evaluate and compare the result of all tested models. The evaluation and comparison results are presented in Chapter 5. All the models that we used were implemented in Scikit-Learn library in Python.

The following outlines the machine learning models that we used in this work:

1- Ridge Regression

Ridge regression is a variant of linear regression [64], which is a machine learning algorithm that performs regression analysis [65]. Regression models a target value based on the input features. Particularly, regression predicts a dependent variable value based on a given independent variable. In linear regression, this is done by finding a linear relationship between the dependent and independent variables, as shown in Figure 4-1. A simple regression line can be modelled by:

$$\hat{y} = B_0 + B_1 x \quad \textbf{(4.5)}$$

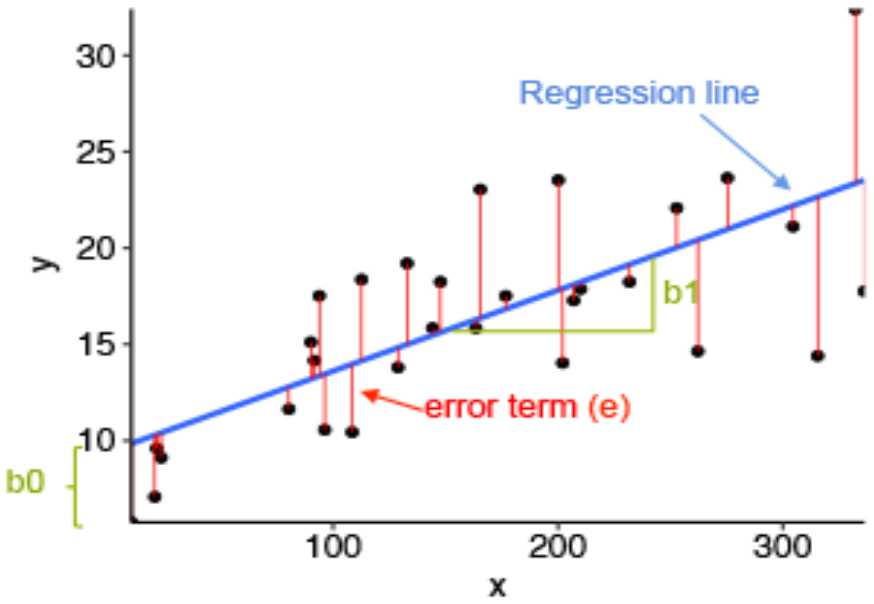


Figure 4-1: Linear regression example

where x is the input training data and $\hat{y}$ is the predicted target variable. During training, the model tries to find the best-fit line to predict the value of y, the observed value, for a given value of x, by finding the optimal values for the coefficients B_0 and B_1. Figure 4-1, shows an example of linear regression with one independent variable.

The task of finding the best-fit line is often solved by minimizing the sum of squares of the residuals [66], where a residual is the difference between the observed value and the value fitted by the model. This formula for minimizing the sum of squares is shown below:

$$\boldsymbol{min}_{\boldsymbol{B_0, B_1,}} \| \hat{\boldsymbol{y}} - \boldsymbol{y} \|_2^{\,2} \tag{4.6}$$

Two common problems that occur in machine learning are the underfitting and overfitting. Underfitting occurs when the model does not learn enough from the data and cannot capture its underlying structure [67]. This causes the model to have unreliable predictions and low generalization power. By comparison, overfitting occurs when the model is very flexible, and cannot generalize well from the training data to other unseen data [68]. In other words, the model performs well on the training set, but performs poorly on the test set. When there is collinearity between the data variables, that is, the data variables are highly correlated, the least squares method causes overfitting. Overfitting is a major concern, because it negatively impacts the performance of the model. To prevent overfitting, regularization techniques are often used to reduce the complexity of the model. There are two variants of linear regression that perform regularization:

Lasso Regression: Performs L1 regularization [69], where the least squares method is modified to also minimize the absolute sum of β through the shrinkage hyperparameter λ, as illustrated in the following formula:

$$\boldsymbol{argmin}_{\boldsymbol{\beta} \in \mathbb{R}^{\mathrm{P}}} \| \boldsymbol{y} - \boldsymbol{X\beta} \|_2^{\,2} + \boldsymbol{\lambda} \| \boldsymbol{\beta} \|_1 \tag{4.7}$$

where β is the coefficient vector [B_0, B_1].

Ridge Regression: performs L2 regularization, where the least squares method is modified to also minimize the squared absolute sum of β through the shrinkage hyperparameter λ, as shown in the following formula:

$$\boldsymbol{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^{\mathrm{P}}} \ \| \boldsymbol{y} - \boldsymbol{X\beta} \|_2^{\,2} + \boldsymbol{\lambda} \ \| \boldsymbol{\beta} \|_2^{\,2} \quad \textbf{(4.8)}$$

where β is the coefficient vector [B_0, B_1].

2- K-Nearest Neighbor for Regression

The K-Nearest Neighbor (KNN) is a supervised learning algorithm that is commonly used in machine learning tasks for classification and regression due to its simplicity and applicability in many real-world problems. It is a non-parametric algorithm that does not make assumptions about the data distribution. The KNN algorithm predicts the target value based on the similarity between different points. To find similar points, the algorithm uses a distance measure such as Euclidean distance, Hamming distance, Manhattan distance, or Minkowski distance [70].

For every point P_i it takes as input, it finds the nearest k neighboring points of P_i using one of the distance measures mentioned above, and then predicts the output of P_i based on the value of its nearest neighbors. In case of classification, the output label of P_i would be equal to the majority vote of its k neighbors. In case of regression, the output target value of P_i would be the mean of its nearest k neighbors. Accordingly, for the algorithm to compute the output of any point, the algorithm needs to have available all the other points. As a result, all the training data is stored and used in the testing stage. This process takes up memory and disk space which causes the testing to be slower. Figure 4-2 displays two examples of KNN for regression and classification, where the number of k is three.

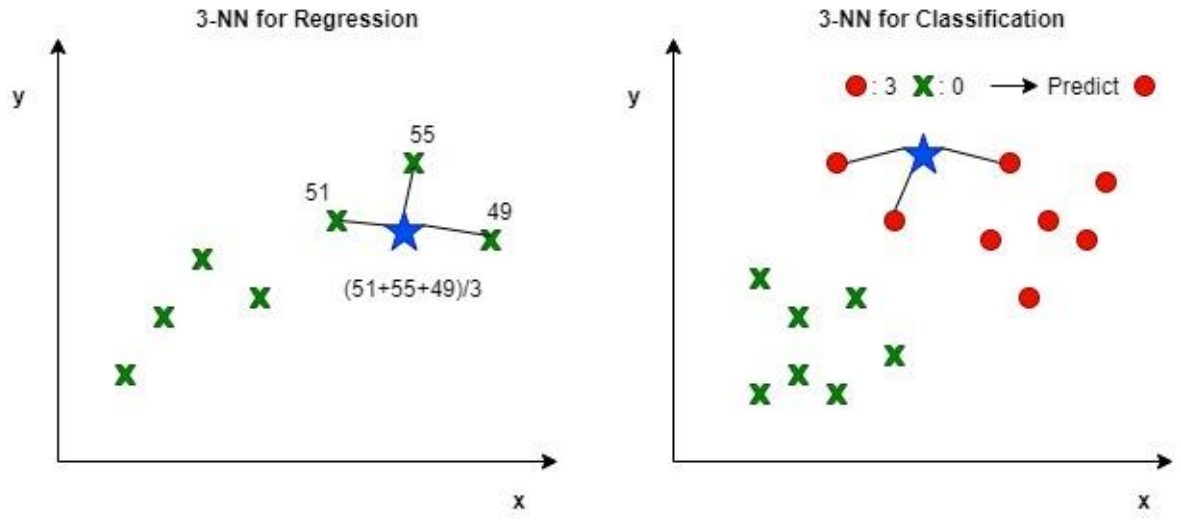


Figure 4-2: KNN Example for (a) regression, (b) classification.

Other than taking up space and memory, the algorithm has yet another disadvantage, it requires feature scaling since distance measures like the Euclidean distance are very sensitive to the magnitude of the data. The one advantage of the KNN algorithm is that its training phase is significantly faster than other machine learning algorithms, as it is not required to train the model for generalization.

3- Support Vector Machine Regression (SVR)

Support vector machine (SVM) is a supervised learning algorithm that can be used for both classification (SVC) and regression (SVR). SVM uses a technique called *the kernel trick* to transform the data variables into a higher dimensional space, and then finds the optimal linear boundary between the possible outputs based on these transformations [71]. Before explaining how SVM works, we should first clarify a few terms. A hyperplane is a separation line that helps classify different data points in the case of classification, or a line that fits the data in the case of regression. Additionally, there are two other lines in SVM called the boundary lines, and their function is to create a boundary. In classification, these boundary lines separate the two classes. In regression, the boundary lines bound the points that are considered for the prediction

process. Another term we need to define is the support vectors, which are points that are closest to the boundary, and can even lie on it.

In the case of classification, the SVC uses the kernel trick to convert a problem with data that is not linearly separable into data that is separable. For example, if there are two classes in the data and the classes are not linearly separable, the kernel trick can add one more dimension to the data, and then the algorithm can find a hyperplane that could separate the two classes in the higher dimensional space.

In the case of regression, SVR uses the kernel trick to map the input into a higher dimensional feature space and then constructs a linear regression model in this higher dimensional space [72].

In SVR, the distance between the hyperplane and the boundary line is denoted by ε. The goal is to find the optimal value of ε so that the support vectors lie within that boundary line. Figure 4-3 illustrates the concept of SVR, where the hyperplane is shown by the solid line, and the boundary lines are shown by the dashed lines in the figure.

The linear function is given by:

$$\hat{y} = \boldsymbol{wx} + \boldsymbol{b} \tag{4.9}$$

where y is the predicted target variable, x is the input variable, w is the weight coefficient and b is the bias. The optimization process needs to maximize the margin by minimizing the squared sum of the weight coefficients to ensure the function is as flat as possible, and it is given by the following formula:

$$\frac{\mathbf{1}}{\mathbf{2}} \ |\boldsymbol{w}|^{\mathbf{2}} \tag{4.10}$$

The constraint that all of the residuals are less than ε is ensured by the following rule:

$$\forall_{\boldsymbol{i}}: |\boldsymbol{y_i} - (\boldsymbol{wx_i} + \boldsymbol{b})| \leq \boldsymbol{\varepsilon} \tag{4.11}$$

where $\boldsymbol{i}$ is data points.

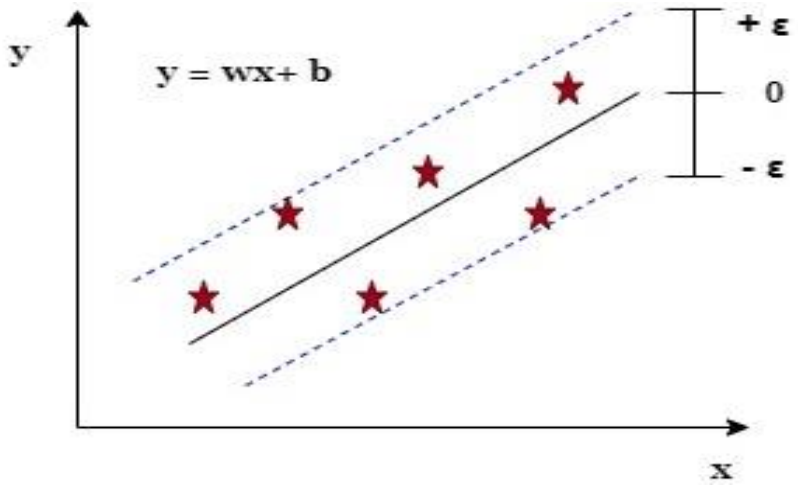


Figure 4-3: SVR Algorithm

4- Random Forest for Regression

Random Forest is a type of ensemble learning method, where a group of weak models are combined to form a strong model [73]. It is a supervised learning algorithm that is constructed through an aggregation of decision trees, where each tree is trained on a subset taken from the data. Generally, the larger the number of decision trees, the more accurate the prediction results are.

Decision trees use a tree-like graph to formulate rules and make predictions based on these rules. In decision trees, each node represents a feature, each branch represents a decision, and each leaf represents an output. The goal is to create a tree for each of the features in the dataset and use the tree to produce a different output at each leaf. The output that is produced depends on the set of decisions made by the tree as it processes the input feature vector.

The decision trees are constructed in a top-down approach that is also known as recursive binary splitting. The construction begins at the top of the tree where entire dataset is available, and then splits the predictor space into two new branches down the tree. During construction, an important step is to identify the feature that will be represented by the root node at each level. For efficient prediction, we need to split the nodes at the most informative features [74]. In the case of classification, a metric commonly used in this step is

the information gain [71], which is the decrease in entropy. Entropy is a measure of the impurity in the data samples and is computed by:

$$\boldsymbol{E(t) = -\sum_{i=1}^{m} p(i|t)\, log_2\, p(i|t)} \quad \textbf{(4.12)}$$

where $p(i|t)$ is the number of samples and t represents a node in the tree.

Whenever we split a node, the entropy changes. Information gain computes the difference between entropy before and after the split of a node.

In the case of regression, however, another metric is needed to accommodate for the continuous variables. A metric that could be used in regression is the weighted mean square error (MSE), which is given by:

$$\boldsymbol{MSE\ (t) = \frac{1}{N_t}\sum_{i \in D_t}(y^{(i)} - \hat{y}_t)^2} \quad \textbf{(4.13)}$$

where N_t is the number of training samples at node t, D_t is the training subset at node t, $y^{(i)}$ is the actual target value, and $\hat{y}_t$ is the predicted target.

Decision trees are widely used due to their high level of accuracy and stability. Unlike linear models, they are capable of mapping nonlinear relationships within the data attributes.

As we mentioned before, random forest is a collection of decision trees. It works by randomly picking a few sub-samples from the data samples to build each tree. Then, several features are selected randomly from all features to ensure that the trees are not highly correlated. To make a prediction, each tree makes a vote by predicting the target value. The forest then takes the average of all the votes by the different trees in the forest. The Random Forest algorithm is illustrated in Figure 4-4.

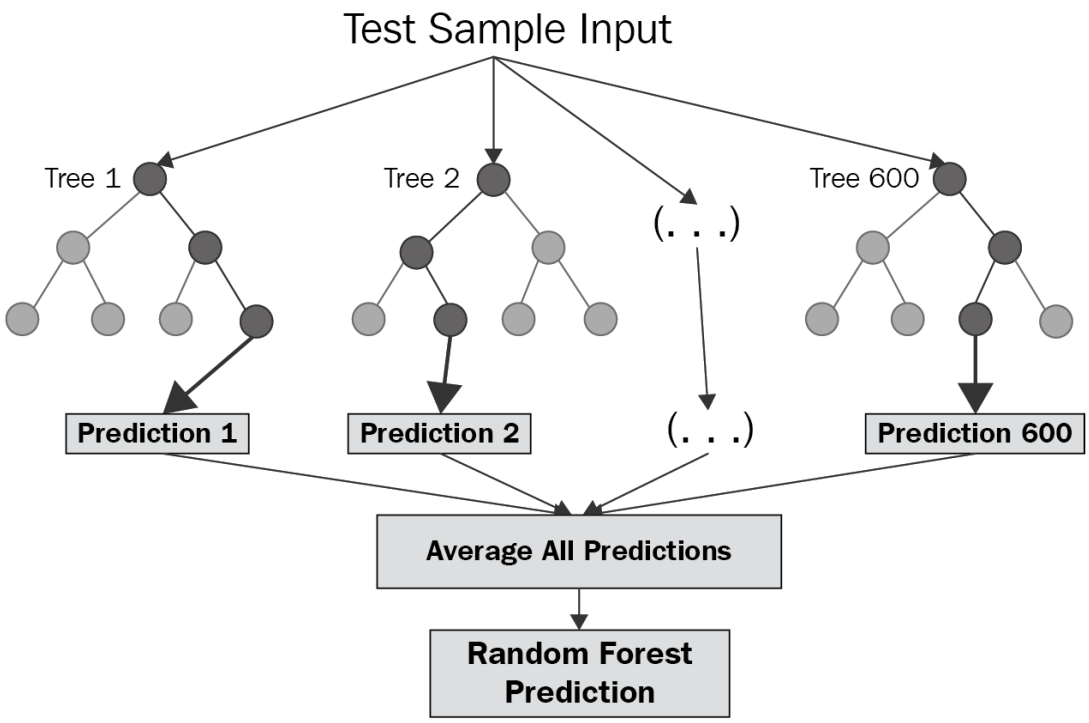


Figure 4-4: Random Forest algorithm structure [75]

There are numerous advantages to use the Random Forest algorithm. It is a robust model that can handle datasets with high dimensionality. Moreover, the model is very useful in data exploration, as it has a property called feature importance that shows the importance of each feature to the prediction, which is useful for feature selection. Additionally, the model requires less data cleaning than many other modeling techniques and it does not make any assumptions about the data distribution.

4.4 Hyperparameters Tuning

Hyperparameters are configurable parameters that define the model architecture and govern the training process [76]. They are tuned by training the model, testing it and examining the error, and then adjusting the parameters. Moreover, hyperparameters are not learned automatically by the model, they must be set manually. Various methods have been proposed for hyperparameter tuning.

The following are the most often used:

1- Grid Search

In this method, the set of hyperparameters are defined and then an extensive search is performed to search through all combinations of hyperparameters to find the optimal one [77]. Moreover, the method determines the optimal hyperparameter by measuring the performance using cross-validation. The grid search is a simple, easy to use method, but it can be computationally expensive when the data has a high dimensional space.

2- Random Search

Unlike grid search, where all different hyperparameter combinations are tested, this method pulls a set of random values from the hyperparameter space until the optimal value is found [75]. The random search method has a higher speed and performance level than the grid search. However, a disadvantage of this method is that it is not affected by previous selections when choosing the next set of hyperparameters.

3- Bayesian Optimization

In this method, a probabilistic model is constructed between the hyperparameter values and the performance measure using the test data [78]. Additionally, different hyperparameters combinations are tested and the model is adapted based on the evaluation of these combinations.

In addition, different machine learning algorithms require different hyperparameters to be tuned. For each algorithm below, we discuss the hyperparameters that need to be optimized.

Ridge Regression: The hyperparameter that is used in ridge regression is shrinkage hyperparameter λ as we described earlier. A larger value of λ reduces the model complexity and prevents overfitting. However, after a certain point, increasing the value of λ may lead to underfitting. Hyperparameter tuning in ridge regression involves balancing the regularization strength to achieve the best possible model performance.

KNN for Regression: Choosing the value of k affects the performance of the model. Increasing the value of k reduces the noise and leads to smoother predictions although, increasing the value may cause underfitting. As with ridge regression, the idea of hyperparameter tuning in KNN regression is to find the optimal value of k that improves the performance of the model.

SVR: The hyperparameters used in SVR training are the kernel function, ε, and the regularization parameter C, which penalizes misclassification and margin errors. As there are multiple combinations of these hyperparameters, they are often tuned using the grid search. This approach is based on the concept of exhaustive search; in where it tries out different hyperparameter combinations and determines the optimal value of each hyperparameter.

Random Forest for Regression: Hyperparameters in a Random Forest include the number of decision trees in the forest and the maximum number of features considered by each tree for splitting a node. Increasing the number of decision trees improves the accuracy of the model as it considers the votes from various trees, however, this process can be computationally expensive. Moreover, increasing the maximum number of features considered by each tree for splitting a node can lead to a higher model performance as it causes the trees to have more features to select from the optimal split, but this can cause overfitting.

4.5 Data Splitting

To effectively evaluate the performance of machine learning models, cross-validation is a commonly used method. In cross-validation, the dataset is split into two subsets a training set and a test set [79]. The training set is the partition of the dataset that the model is trained on. The test set is the partition of the dataset that the model has not seen before. Both the training set and the test set consist of some input features and a target variable. The model is fit on the features and target variable from the training set and then the fitted model is used to predict the target variable for the features in the test set. The test set should be large enough to produce statistically meaningful results and should be representative of the entire dataset.

The dataset split ratio depends on two factors the number of samples in the dataset and the type of model that is being trained, as some models require a large amount of data to train on.

There are three main methods for performing cross-validation: the holdout method, the k-fold cross-validation and the leave-one-out cross-validation.

Holdout Method: The method of this approach is to remove a subset of the dataset and then use the subset or the dataset to evaluate a model trained on the rest of the data [80]. The error is an indication of how well

the model is going to perform on the test set. A common split ratio is 70%-30%, where the training set consists of 70% of the dataset and test set consists of 30% of the dataset.

K-Fold Cross-Validation: In this approach, the dataset is divided into k subsets. Next, the model is trained on all subsets except for one, and then the model is evaluated on that one subset that was not used in training. This process is repeated k times, where each time a different subset is used for evaluation [81]. Thus, k-fold cross-validation performs the holdout method k times. This approach is commonly used in small datasets to prevent overfitting, as it improves the generalization power of the model.

Leave-One-Out Cross-Validation: This approach is a special version of k-fold cross-validation. In leave-one-out cross-validation, the number of folds is equal to the dataset samples [82]. Furthermore, the model is trained repeatedly, where each time one sample is selected as a test set, and all the other samples are selected as a training set.

We trained and evaluated the models using the Holdout Method validation method with various data split ratios and reached the best performance with a split ratio of 80% training set and 20% test set.

4.6 Evaluation Metrics

We evaluate the performance of the models using two well-known metrics: the R^2 score and the root mean square error (RMSE). In the following subsections, these two metrics are described.

4.6.1 R^2 score

Also called the coefficient of determination, the R^2 score is a goodness-of-fit measure for regression models. It indicates the percentage of the variance in the dependent variable that can be explained by the independent variables [80]. The R^2 score measures the strength of the relationship between the model and the dependent variable on a convenient 0 – 100% scale.

The R^2 score is computed by:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2} \quad \textbf{(4.14)}$$

where y is the actual value, $\hat{y}$ is the predicted value and $\bar{y}$ is the mean of all y values. The R^2 has a range from 0-1 , where 0 indicates that the model does not explain any of variability of the response data around its mean, and 1 indicates that the model explains all the variability of the response data around its mean... Accordingly, a higher R^2 score specifies that the model fits the data better.

4.6.2 Root Mean Square Error (RMSE)

The RMSE is the standard deviation of the prediction errors, also called residuals. In Figure 4-5, we can see an example of a fitted linear model, where the residuals are the vertical lines showing the difference between the actual data values and values predicted by the model.

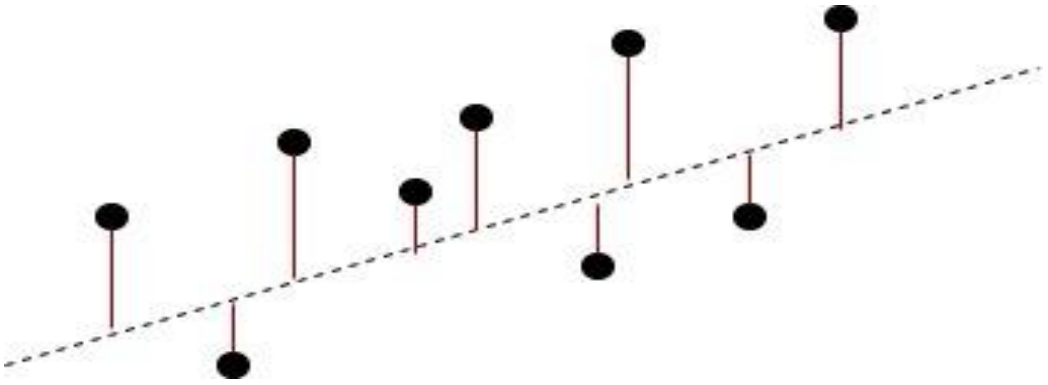


Figure 4-5: Residuals in a linear model

The RMSE is used to measure how the residuals are dispersed. Moreover, it can be used to compare the prediction errors of different models to determine the model with the highest performance on the data.

The RMSE can be calculated by the following equation:

$$\boldsymbol{RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted_i - Actual_i)^2}{N}}} \quad \textbf{(4.15)}$$

where $Predicted_i$ corresponds to the model predictions, $Actual_i$ corresponds to the actual values of the data samples, and N is the number of samples. Lower RMSE values indicate that the model fits the data better.

4.6.3 Mean Absolute Error (MAE)

We also used the MAE as our performance evaluation metric, this was calculated by the following equation:

$$\boldsymbol{MAE = \frac{1}{N}\sum_{i=1}^{N}|Predicted_i - Actual_i|} \quad \textbf{(4.16)}$$

4.7 Summary

In this chapter, we described the preprocessing steps and explained how we performed feature scaling. We discussed the machine learning models that we have trained and the hyperparameters tuning methods that we have explored. Furthermore, we talked about the evaluation metrics that we used to evaluate the performance of our models. In the next chapter, we will present the results of training and testing the models for QoE prediction and compare them based on the evaluation metrics that we mentioned in this chapter.

Chapter 5

Results and Discussion

We presented in Chapter 3 the data preparation and feature extraction methods that were applied to the Waterloo dataset in order to extract the quality features needed for the prediction process. In Chapter 4, we discussed the machine learning prediction models that will be trained to carry out the QoE prediction. In this chapter, we present the results of training different machine learning regression models using the proposed enhanced set of features for QoE prediction. In Section 5.1, we describe the results produced with machine learning regression models, we also illustrate the performance of each regression model and the metrics used to measure the performance. In Section 5.2, we present performance of each regression model on feature subsets. Moreover, we perform a comparative analysis between the different trained models.

5.1 Results of Machine Learning Models

As mentioned in the Chapter 4, we trained several machine learning models on the collected data for QoE prediction. We performed a comparative analysis between the different models to determine the best predictor for our problem. The test results for each model are presented in the following subsections.

We trained several regression models on the Waterloo III Video QoE Database, by creating two separate video sequence sets one for training and one for testing. Within each set, training or testing, all video sequences were used for training or testing, it is a common approach used to account for content dependencies.

The tested regression models are linear models (Ridge and Lasso regression), K-nearest neighbors, Support Vector Regression (SVR) using an RBF kernel and one ensemble method, Random Forests (RF). We preprocessed the features for all regression models by mean subtraction and scaling to unit variance. Note that we computed the data mean and variance in the feature transformation step using only the training data. For each of the regression models, we determined the best parameters using 10-fold cross validation on the training set. This process was repeated on all possible train/test splits.

First we trained each of the regression models, next we applied regression on the test set to make QoE predictions. We computed the correlation between the predicted QoE values which resulted after regression and the true MOS scores in the test set, using the Spearman Rank Order Correlation Coefficients (SRCC) and the Pearson Linear Correlation Coefficients (PLCC). We also assessed the learning model performance using all evaluation metrics mentioned in Chapter 4, the R^2 score and the root mean square error (RMSE), the mean absolute error (MAE) and model train time.

We conducted 100 different runs, each using a random 80% train and 20% test split of the video content. The calculations of all previously mentioned evaluation metrics were repeated on each of the runs yielding a set of SRCC and PLCC, R2 score, and RMSE values for all possible train/test content combinations. Then the median value for each set is calculated to generate a single number describing the performance level of our model.

For preprocessing the data and training the machine learning models, we used a Lenovo computer, INTEL® CORE™ i5-7200U Processor and a 4 GB RAM. Moreover, we used Python 3 on a Linux operating system. Scikit-Learn [83], Numpy [84], Pandas [85] and Matplotlib [86] libraries in Python comprise the machine learning framework.

Tables 5.1 show the regression results using only the quality features namely, init_buf_time, reb_perc, av_sw_mag, time_hig_layer, stall_dur, sw_count without using any VQA metric. while the results of using all features with different VQA metrics added are shown in Tables 5.2 -5. 6.. The table columns list the metrics used, the R2 score, along with the model train_time, the RMSE and the MAE. All results are reported on 100 runs of 80% train and 20% test splits.

Table 5-1 Result of regression on quality features

Before	R2	Train_time	RMSE	MAE
Ridge	**0.272**	**0.140**	**12.004**	**9.379**
Lasso	**0.324**	**0.135**	**8.450**	**6.123**
SVR	**0.345**	**9.424**	**11.622**	**8.755**
RF	**0.498**	**10.842**	**9.837**	**6.918**
KNN	**0.402**	**1.098**	**11.509**	**8.941**

Table 5-2 Results of Regression on all features when Using VMAF as VQA metric

After	R2	Train_time	RMSE	MAE
Ridge	0.705	0.124	8.935	7.376
Lasso	0.710	0.139	8.883	7.248
SVR	0.691	10.353	8.349	6.621
RF	0.730	7.420	8.044	5.839
KNN	0.649	0.979	8.980	6.968

Table 5-3 Results of Regression on all features when Using VQM as VQA metric

After	R2	Train_time	RMSE	MAE
Ridge	0.654	0.223	8.920	6.888
Lasso	0.692	0.204	9.252	7.460
SVR	0.732	9.925	7.783	6.435
RF	0.688	11.549	7.625	6.095
KNN	0.682	0.9762	9.006	6.961

Table 5-4 Results of Regression on all features when Using SSIMplus as VQA metric

After	R2	Train_time	RMSE	MAE
Ridge	0.6537	0.142	9.730	7.758
Lasso	0.701	0.1480	8.449	6.801
SVR	0.738	9.780	8.372	6.054
RF	0.752	11.473	8.162	6.088
KNN	0.7306	0.9718	7.767	9.760

Table 5-5 Results of Regression on all features when Using STRRED as VQA metric

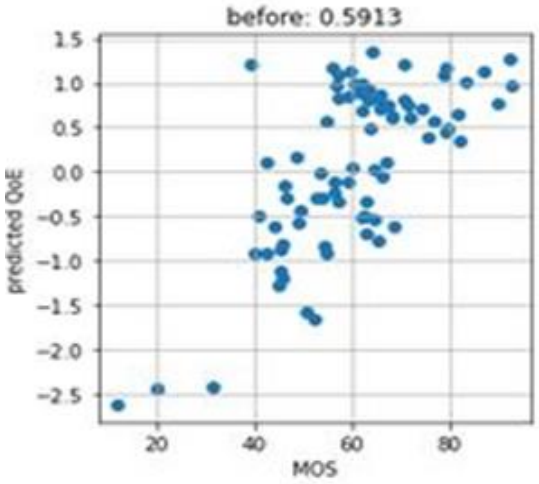
After	R2	Train_time	RMSE	MAE
Ridge	0.424	0.134	11.711	8.980
Lasso	0.454	0.155	10.522	8.507
SVR	0.513	9.599	9.653	7.429
RF	0.690	7.384	9.309	7.424
KNN	0.546	1.027	8.814	6.661

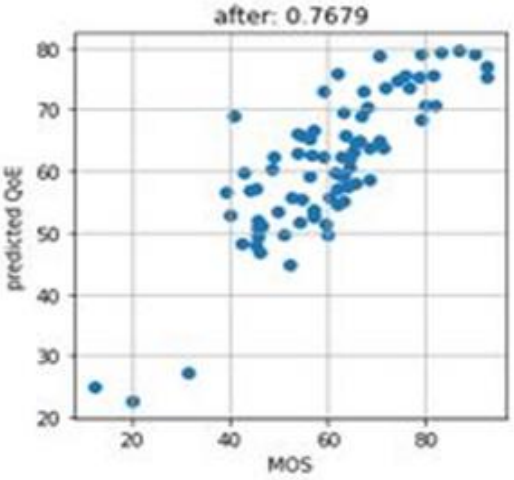
Table 5-6 Results of Regression on all features when Using PSNR as VQA metric

After	R2	Train_time	RMSE	MAE
Ridge	**0.465**	**0.131**	**11.230**	**8.400**
Lasso	**0.418**	**0.147**	**11.547**	**9.580**
SVR	**0.636**	**9.859**	**7.987**	**7.883**
RF	**0.565**	**7.312**	**10.381**	**7.768**
KNN	**0.482**	**1.144**	**9.398**	**6.949**

It can be noted that the inclusion of different VQA metrics to the features has resulted in an improvement in the prediction score of all regression models. Specifically, for the metric, VMAF [57], Random Forest regression has reached 0.730%, followed by Lasso with 0.710% and ridge with 0.705%. When using VQM [42] as the VQA metric, SVR regression result in 0.73% prediction score (accuracy)**.** Using SSIMplus as VQA metric, Random Forest results in the highest prediction score among all models and all metrics which is 0.752%followed by SVR at 0.738% and next come KNN with 0.7306%

Figures 5.1-5.7 demonstrate samples of the scatter plots of the predicted QoE score vs the MOS score once with using just the objective quality metrics (up) and when they are combined with all features (down). The SRCC and PLCC values in both cases are also shown. Clearly, the predicted QoE significantly improved both in terms of the linear relation between variables refereed to as linearity and strength and direction of monotonic relationship referred to as monotonicity.






```
SRCC before (VMAF): 0.591306793884942
SRCC using allfeatures (SVR): 0.7678621659634318

PLCC before (VMAF): (0.6988248661744908
PLCC using allfeatures (SVR): (0.8285503369070746
```

Figure 5-2: MOS scores (horizontal axis) against predicted QoE scores (vertical axis) using VMAF and Support vector regressors. Up: when using only VMAF to predict the QoE; Down: QoE scores after regression when using all features.

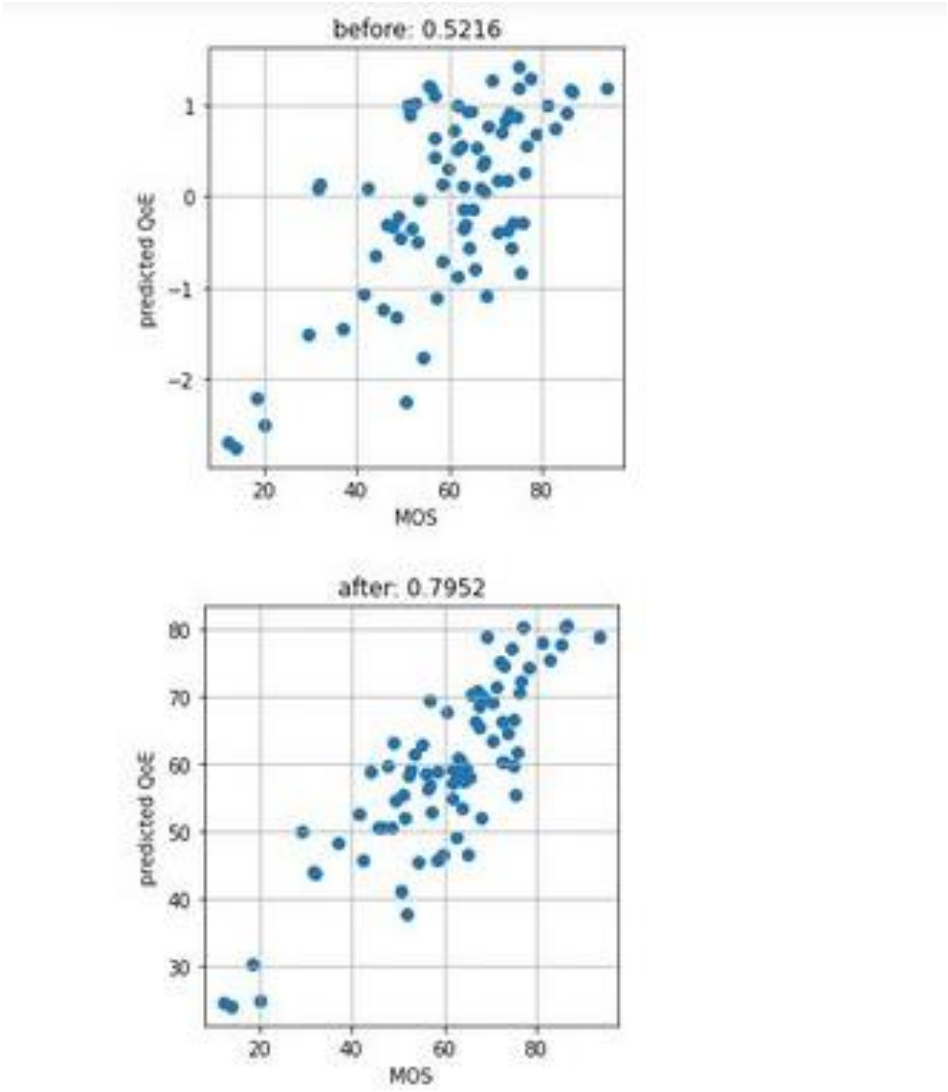
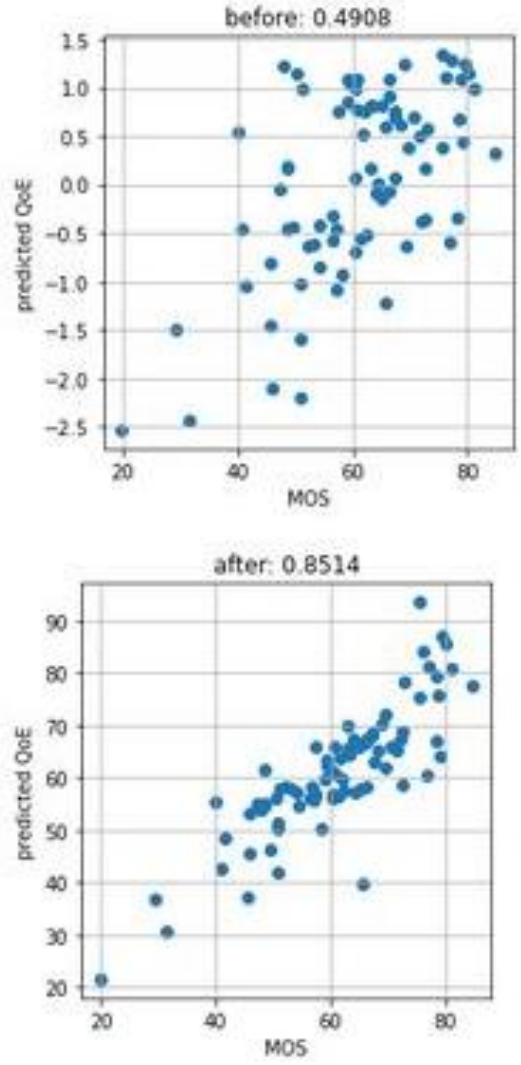


Figure 5-2: MOS scores (horizontal axis) against predicted QoE scores (vertical axis) using VMAF and Ridge regressors. Up: when using only VMAF to predict the QoE; Down: QoE scores after regression when using all features




```
SRCC before (VMAF): 0.49079337548349405
SRCC using allfeatures (RandomForest): 0.8513595874355369

PLCC before (VMAF): (0.5891311371900778
PLCC using allfeatures (RandomForest): (0.8468449234415192
```

Figure 5-3: MOS scores (horizontal axis) against predicted QoE scores (vertical axis) using VMAF and Random Forest regressors. Up: when using only VMAF to predict the QoE; Down: QoE scores after regression when using all features.

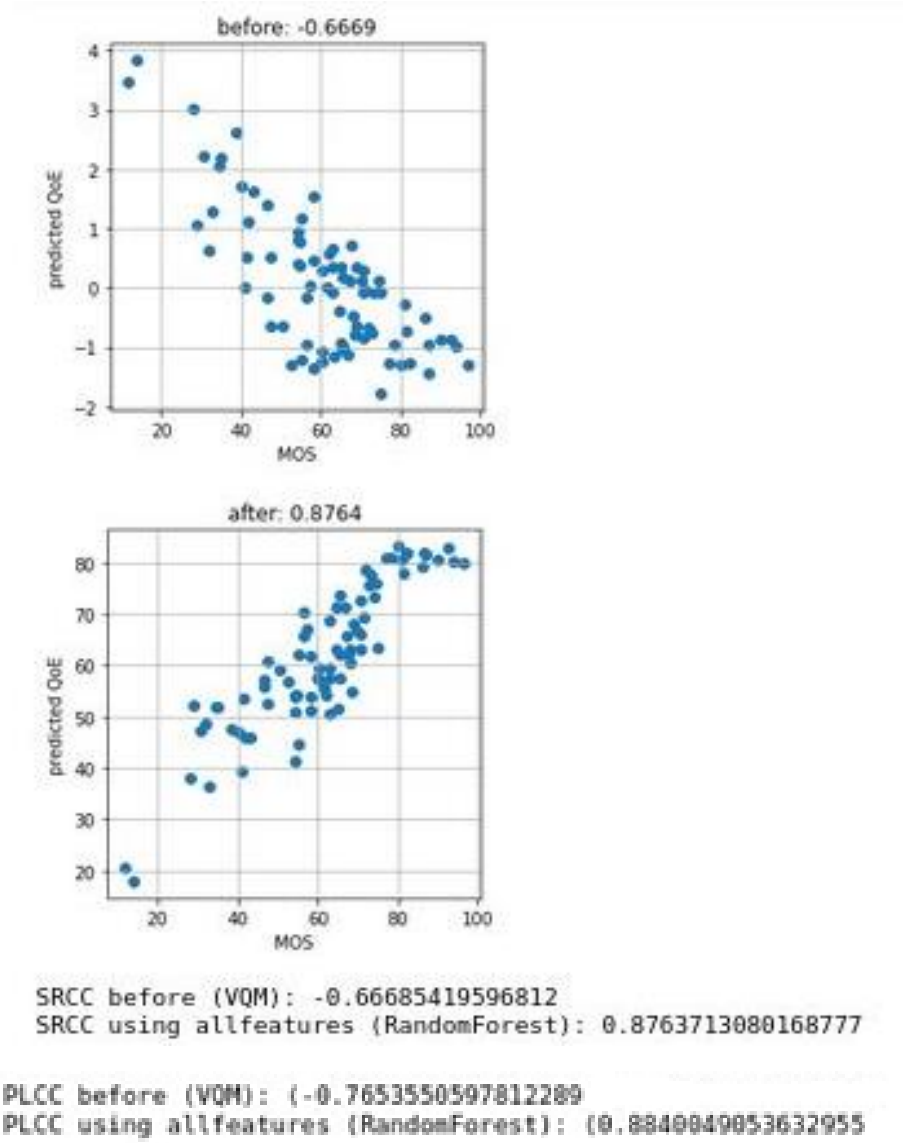


Figure 5-4: MOS scores (horizontal axis) against predicted QoE scores (vertical axis) using VQM and Random Forest regressors. Up: when using only VMAF to predict the QoE; Down: QoE scores after regression when using all features.

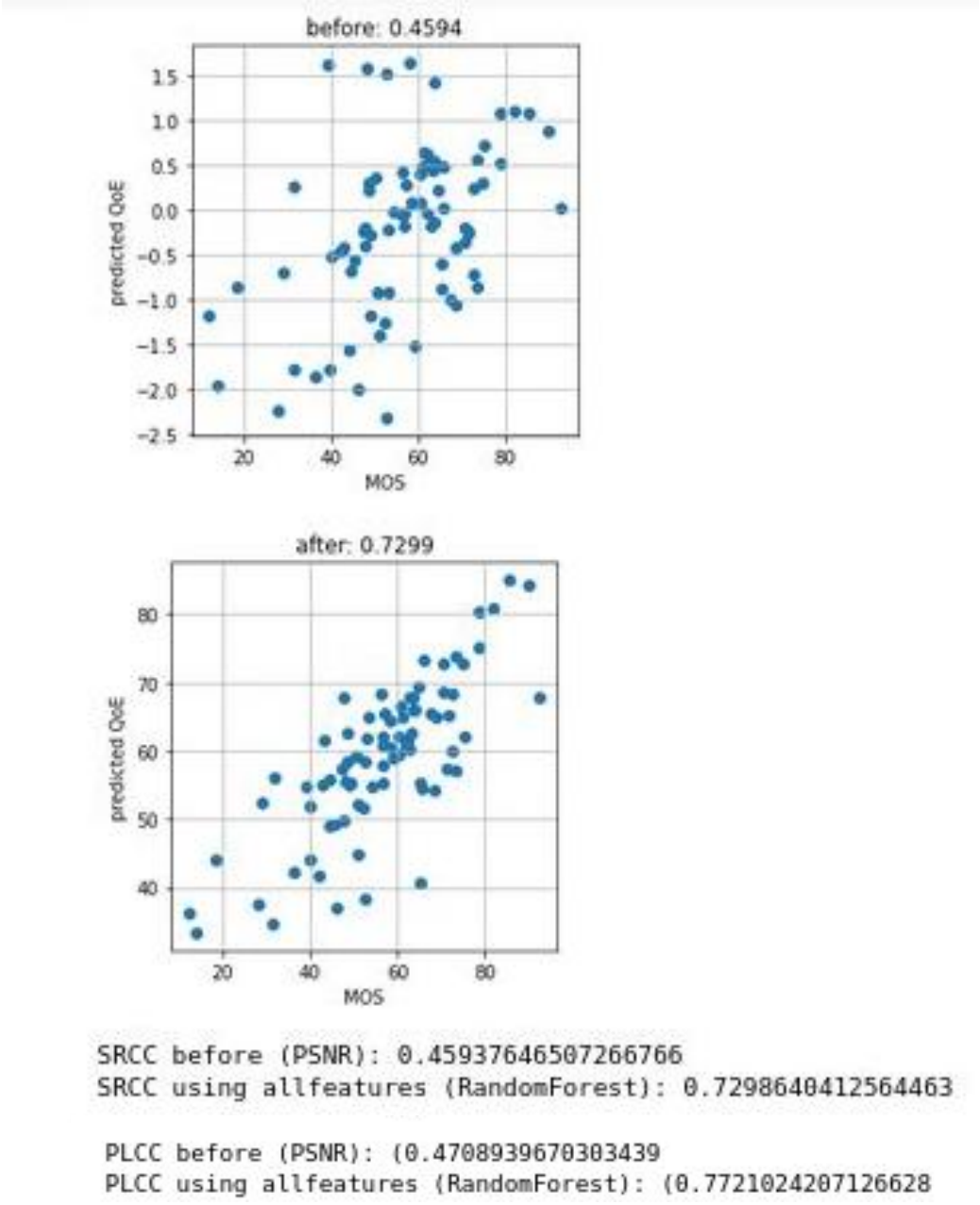
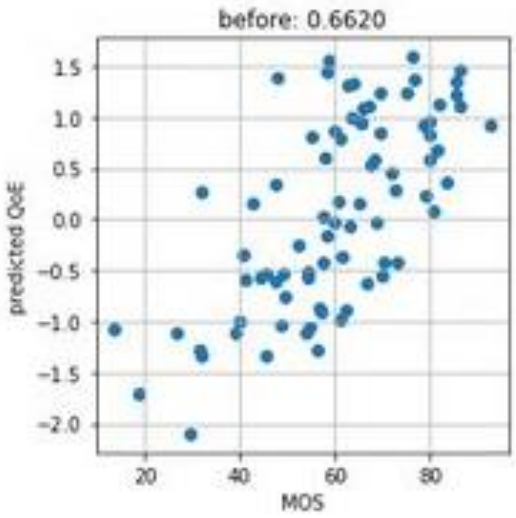
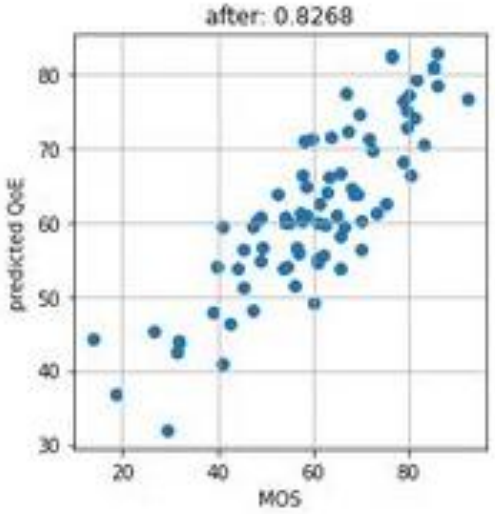


Figure 5-5 MOS scores (horizontal axis) against predicted QoE scores (vertical axis) using PSNR and Random Forest regressors. Up: when using only PSNR to predict the QoE; Down: QoE scores after regression when using all features.

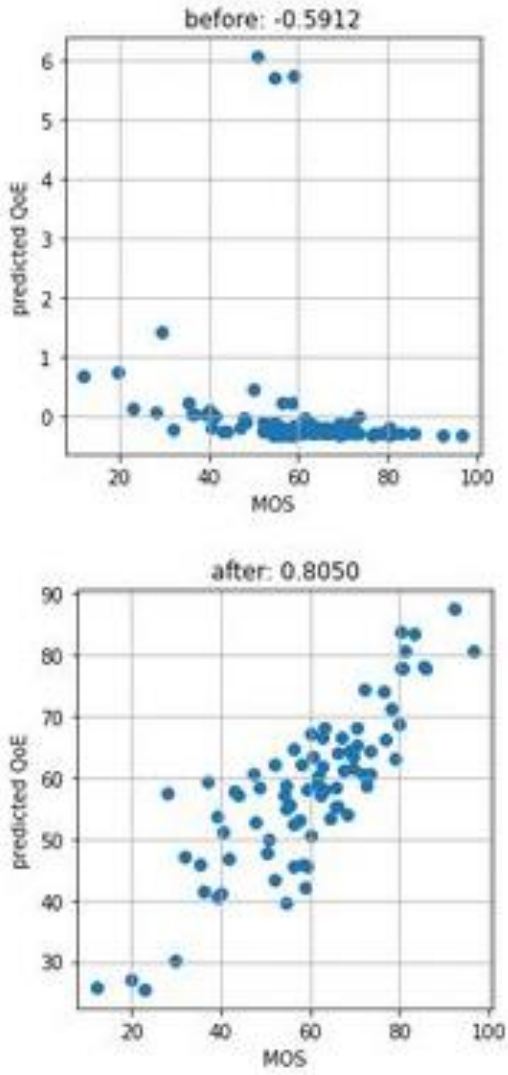





```
SRCC before (SSIMplus): 0.6620253164556963
SRCC using allfeatures (lasso): 0.8267698077824661

PLCC before (SSIMplus): (0.672798600615445
PLCC using allfeatures (lasso): (0.853547252184404
```

Figure 5-6: MOS scores (horizontal axis) against predicted QoE scores (vertical axis) using SSIMplus and Lasso regressors. Up: when using only SSIMplus to predict the QoE; Down: QoE scores after regression when using all features.




```
SRCC before (STRRED): -0.5912095639943742
SRCC using allfeatures (RandomForest): 0.8049929676511955

PLCC before (STRRED): (-0.2028593764830843
PLCC using allfeatures (RandomForest): (0.8360869964205825
```

Figure 5-7: MOS scores (horizontal axis) against predicted QoE scores (vertical axis) using STRRED and Random Forest regressors. Up: when using only STRREDs to predict the QoE; Down: QoE scores after regression when using all features.

Table 5.7 (a,b) summarizes the results of SRCC , and PLCC respectively, for the different video quality assessment (VQA) algorithms where the median SRCC and PLCC before regression, which is calculated

on data set after we split into train/test, and we compute the correlation between VQA metric and MOS score before using any regressor and the median SRCC and PLCC values after regression when using different objective quality metrics added to the features. All results are reported on 100 runs of 80% train and 20% test splits.

Table 5.7 shows the SRCC and PLCC results on different video quality assessment algorithm

a) SRCC Results

VQA	VMAF	VQM	SSIMplus	STRRED	PSNR
Before	**0.417**	**0.342**	**0.535**	**0.604**	**0.343**
Ridge	**0.795**	**0.828**	**0.734**	**0.690**	**0.653**
Lasso	**0.863**	**0.817**	**0.827**	**0.628**	**0.578**
SVR	**0.768**	**0.822**	**0.848**	**0.692**	**0.621**
RF	**0.851**	**0.800**	**0.858**	**0.747**	**0.730**
KNN	**0.806**	**0.857**	**0.850**	**0.748**	**0.677**

(b) PLCC Results

VQA	VMAF	VQM	SSIMplus	STRRED	PSNR
Before	**0.437**	**0.356**	**0.546**	**0.617**	**0.354**
Ridge	**0.845**	**0.838**	**0.754**	**0.693**	**0.673**
Lasso	**0.821**	**0.834**	**0.712**	**0.610**	**0.579**
SVR	**0.829**	**0.822**	**0.812**	**0.692**	**0.800**
RF	**0.847**	**0.800**	**0.858**	**0.747**	**0.772**
KNN	**0.816**	**0.857**	**0.859**	**0.747**	**0.677**

Note that the SRCC and PLCC values improve when using the regression scheme for all quality metrics. For STRRED and PSNR, the improvements were much less than those obtained with other objective metrics. However, the improvements of VQM, VMAF, SSIMplus were remarkably higher for all the regression models. SSIMplus using Random Forest and VMAF using SVR yielded the best overall performance in terms of SRCC and PLCC.

5.2 performance of regression model on feature subsets

To enhance this study, it is important to analyze the relative feature contribution to the overall prediction results. One way to do that is to utilize the tree-based method. First, we picked the best and the worst performing quality models before regression, i.e., VMAF and STRRED along with the highest performing RF regression model (in terms of SRCC, PLCC). Figure 5.8 shows the feature importance after 100 runs on 20% 80% train/test splits.

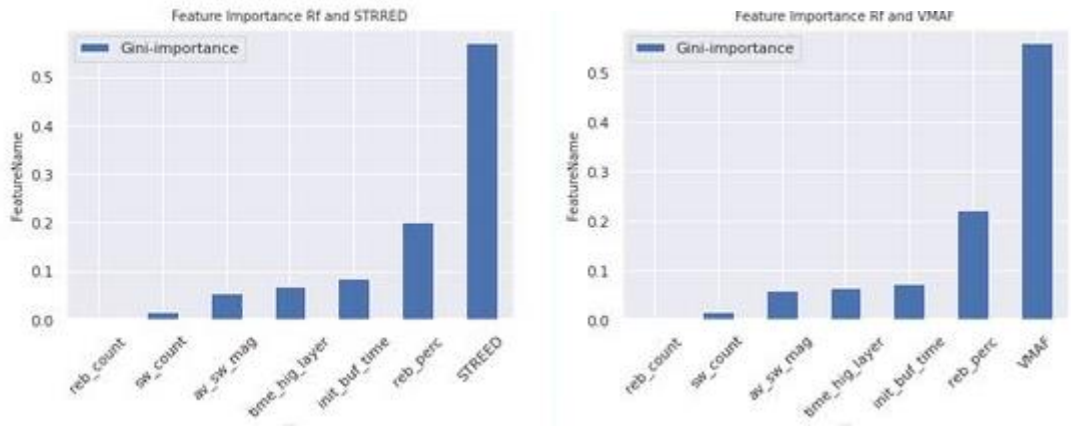


Figure 5-8 Feature Importance

The result from feature importance tree-model shown in figure 5.8 suggests that av_sw_mag, reb_perc and init_buf_time, time_hig_layer are the four most important features. To further investigate the effects of those features on the QoE prediction task, we experimented further by using different feature subsets, and recorded the QoE prediction performance of each. First, consider the following feature subsets with their indices: as shown in Table 5.8, (1) VQA (2) ,time_hig_layer ,(3) av_sw_mag,, (4) reb_perc+init_buf_time (5)VQA+time_hig_layer, (6),VQA+av_sw_mag,, (7) VQA+ time_hig_layer+ init_buf_time ,(8) time_hig_layer+ init_buf_time+ reb_perc, (9) time_hig_layer + init_buf_time + reb_perc +av_sw_mag,(10) VQA+ av_sw_mag +reb_per + init_buf_time, (11). VQA+ time_hig_layer+ reb_perc+init_buf_time, (12) VQA+4 important features. We measured the performance of each feature subset using certain VQA metrics in terms of Spearsman correlation score and Pearson linear correlation. The upper part of Table 5.8 presents SRCC results and the lower part presents the PLCC results.

Features	1	2	3	4	5	6	7	8	9	10	11	12
Ridge	0.604	0.316	0.021	0.163	0.576	0.670	0.544	0.496	0.508	0.770	0.795	0.795
Lasso	0.581	0.229	0.0700	0.164	0.59	0.60	0.565	0.487	0.564	0.787	0.790	0.801
SVR	0.577	0.315	0.264	0.488	0.676	0.610	0.800	0.582	0.625	0.796	0.840	0.866
RF	0.521	0.330	0.315	0.620	0.600	0.567	0.701	0.535	0.65	0.819	0.829	0.863
KNN	0.321	0.322	0.034	0.231	0.510	0.45	0.654	0.431	0.501	0.732	0.72	0.755

SRCC Results

Features	1	2	3	4	5	6	7	8	9	10	11	12
Ridge	0.672	0.341	0.083	0.131	0.747	0.793	0.643	0.544	0.539	0.815	0.821	0.845
Lasso	0.599	0.265	0.097	0.156	0.543	0.665	0.651	0.532	0.543	0.801	0.811	0.823
SVR	0.783	0.424	0.284	0.550	0.763	0.673	0.661	0.529	0.679	0.810	0.814	0.838
RF	0.522	0.362	0.290	0.624	0.701	0.646	0.739	0.537	0.65	0.853	0.840	0.870
KNN	0.325	0.326	0.033	0.321	0.556	0.47	0.657	0.422	0.521	0.734	0.74	0.768

PLCC Results

Table 5-8: Results on different feature subsets when VMAF was used as the quality metric (VQA)

One can note that the regression performance for the feature subsets using VQA was high for almost all regressors. For the different feature combinations and their effect on QoE prediction, when VQA metrics were not included with the feature set, the prediction performance decreased considerably. Regarding the regression models, Ridge and Lasso gave very similar performances when using fewer feature types, but as the number of features grew, the Ridge regression yielded better results. Overall, the combination of all feature types gave the best performance over most regression models as we added more features the performance improved. As our recommendation that the inclusion of diverse QoE-aware features would result in a successful QoE prediction model that help approximate the subjective QoE.

5.3 Summary

In this chapter, we presented results of the QoE prediction where we combined VQA features and quality influence features using different machine learning models. We presented the results of our first experiment

to test performance of regression models that showed using all features with different VQA metrics, the prediction score of all regression models was improved.

Next, we presented the results of our second experiment to test linearity and monotonicity of regression models by computing the values of SRCC and PLCC. Using all features with different objective quality metrics has significantly improved the two values.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The rapid growth of streaming services due to the deployment of higher capacity mobile networks and more efficient video compression and streaming techniques required the development of new standard for video streaming service where DASH was adopted by service providers. Despite its capability in providing adaptive bitrates that match fluctuating bandwidth, DASH results in perceptual quality degradation. This makes service providers look for a way to measure the quality of experience of users to provide higher quality that meets user experience expectation. Therefore, there has been strong demand for QoE prediction models. In this work, we proposed to improve the performance of existing QoE prediction models that still suffer from some limitations such as limited size of dataset or using simulated streaming scenarios. The features that most existing database collect is not complete, as there are many key influence factors that affect the of quality of streaming video such as the initial delay, the rebuffering time and length and the quality switches. Most of existing dataset measures one or two of these factors while ignoring the others. In addition, there is not enough work on developing an integrated QoE aware prediction model that considers different impairments in a streaming session either during normal or during playback interruption such as rebuffering (video freeze), initial delay.

We used the SQoE-III database, which contains data that measures compression artefacts, in addition to initial delay, switching and rebuffering features. The streaming scenarios are realistic.

We proposed a feature enhancement scheme that integrates/combines features obtained during playback interruption or during normal playback. We first did correlation and data analysis to study the effect of both set of features on the subjective user QoE measured as MOS. From the result of correlation analysis of playback interruption features on MOS, we found out that using these features alone is not enough to predict MOS. From the results of the correlation of various VQA models we find higher correlation against MOS.

To predict QoE effectively, we combined both sets of features as a single enhanced set applied to different machine learning models to predict the QoE value.

The results of our first experiment, to test the performance of regression models shows that after using all the features with different VQA metrics, the prediction score of all regression models was improved. Specifically, for

VMAF [42], Random Forest regression has reached 0.730%, followed by Lasso with 0.710% and ridge with 0.705%. When using VQM [42] as our VQA metric, SVR regression resulted in 0.732% prediction score (accuracy)**.** Using SSIMplus as VQA metric, Random Forest results in the highest prediction score among all models and all metrics which is 0.752% followed by SVR at 0.738% and next came KNN with 0.730%

The results of our second experiment to test linearity and monotonicity of regression models shows that the effect of using only the VQA metric and then using VQA combined with all features has significantly improved the SRCC and PLCC values both in terms of monotonicity and linearity.

It is worth mentioning that despite the demonstrated contributions provided by our work, there are certain limitations:

- We worked on just one dataset. Including multiple datasets would enable us to reach to more definitive conclusions.
- The number of volunteers that was involved in the study is limited.
- We were interested in the studies of discrete QoE, where subjects provide a single score describing their overall QoE on each presented video sequence. Studies of continuous-time QoE that involve real-time measurements of each subject's instantaneous QoE are needed.

6.2 Future Research Directions

As future research directions and possible extensions of this work:

1. Develop QoE prediction models directly used for continuous time QoE monitoring. Time series models can be explored to achieve this objective.
2. Construct a new database that includes continuous time subjective data that makes it suitable for designing continuous time QoE models.
3. Deploy methods which integrate temporal aspects of user QoE in order to design better strategies for the resource allocation problem.
4. Other influence factors such as user and context (e.g., habits, cultural background, environment, etc.) can be also considered to design a robust and holistic QoE models.
5. Study the complexity of the proposed models in terms of resource allocation (e.g, storage, energy consumption, etc.) perspective.
6. Propose a scheme that uses the QoE information to deliver adequate levels of video to specific users.

References

[1] G. Dimopoulos, I. Leontiadis, P. Barlet-Ros and K. Papagiannaki, "Measuring Video QoE from Encrypted Traffic," in *Proceedings of the 2016 Internet Measurement Conference*, Santa Monica, California, USA, 2016.

[2] Y. Liu, S. Dey, D. Gillies, F. Ulupinar and M. Luby, "User experience modeling for DASH video," in *20th International Packet Video Workshop, PV 2013.*, San Jose, CA, (2013).

[3] Y. L. Mao, S. Dey, F. Ulupinar, M. Luby and Yinian, "Deriving and Validating User Experience Model for DASH Video Streaming," *IEEE Transactions on Broadcasting,* vol. 61, pp. 651-665, 2015.

[4] Cisco (Jun. 2017)., "Cisco Visual Networking Index: Forecast and Methodology 2016-2021.," [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf.

[5] *A Proposed Media Delivery Index (MDI),* Standard RFC 4445, Apr.2006.

[6] *Parametric Bitstream-Based Quality Assessment of Progressive Download and Adaptive Audiovisual Streaming Services Over Reliable Transport,* document P.1203 ITU-T Recommendation, Nov. 2016.

[7] *Parametric Non-Intrusive Assessment of Audiovisual Media Streaming Quality,* document ITU-T P.1201 Recommendation, Oct.2012.

[8] P. Le Callet, S. Moller and A. Perkis, "Qualinet White Paper on Definitions of Quality of Experience (2012)," in *in Proc. Eur. Netw. Qual. Exper.Multimedia Syst. Services (COST Action IC)*, 2012.

[9] *Vocabulary for Performance and Quality of Service.Amendment 5:New Definitions for Inclusion in Recommendation,* document ITU-T P.10/G.100 Recommendation, Jul. 2016..

[10] *Subjective Video Quality Assessment Methods for Multimedia Applications,* document ITU-T P.910 Recommendation, Apr. 2008.

[11] *Methodology for the Subjective Assessment of the Quality of Television Pictures,* document ITU-T BT.500 Recommendation, Jan. 2012.

[12] *Reference Algorithm for Computing Peak Signal to Noise Ratio of a Processed Video Sequence With Compensation for Constant Spatial Shifts,Constant Temporal Shift, and Constant Luminance Gain and Offset,,* document ITU-T J.340 Recommendation, Jun. 2010.

[13] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "`Image quality assessment: From error visibility to structural similarity," vol. 13 no. 4, pp. 600-612, Apr. 2004.

[14] M. H. Pinson and S. Wolf, "A new standardized method for objectively measuring vidoe quality," *IEEE Trans. Broadcast,* vol. . 50 no. 3, pp. 312-322, Sep. 2004..

[15] M. Seufert, M. Slanina, S. Egger and M. Kottkamp, "To pool or not to pool: A comparison of temporal pooling methods for HTTP adaptive video streaming,," in *in Proc. 5th Int. Workshop Qual. Multimedia Exper. (QoMEX),,* Klagenfurt, Austria, Jul. 2013,.

[16] L. Skorin-Kapov and M. Varela, "A multi-dimensional view of QoE: the ARCU model,," in *in Proc. 35th Int. Conv. MIPRO,,* Opatija, Croatia,, May 2012,.

[17] VQEG (2000), "VQEG FRTV Phase I Final Report," [Online]. Available: https://www.its.bldrdoc.gov/vqeg/projects/frtv-phase-i/frtv-phase-i.aspx.

[18] VQEG (2003), "VQEG FRTV Phase II Final Report," [Online]. Available: https://www.its.bldrdoc.gov/vqeg/projects/frtv-phase-ii/frtv-phase-i.aspx.

[19] Z. Wang, L. Lu and A. Bovik, "Video Quality Assessment Based on Structural Distortion Measurement," *In Signal Processing: Image Communication,* vol. 19, pp. 121-132, February 2004.

[20] "Information Technology-Dynamic Adaptive Streaming Over HTTP (DASH)-Part 1: Media Presentation Description and Segment Formats,Standard ISO/IEC 23009-1:2014, 2017," [Online]. Available: https://www.iso.org/standard/65274.html. [Accessed 17 Nov 2018].

[21] I. Sodagar, “The MPEG-DASH standard for multimedia streaming over the Internet,” *IEEE Multimedia,* vol. . 18 no. 4, pp. 62-67, Apr. 2011.

[22] Apple, “HTTP Live Streaming,” [Online]. Available: https://developer.apple.com/streaming/. [Accessed 17 Nov 2018].

[23] J. Kua, G. Armitage and P. Branch, “A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP,” *IEEE Commun Surveys Tuts,* vol. . 19 no. 3, pp. 1842-1866, 2017.

[24] “Global Internet Phenomena Report: North America and Latin America,” 14 Nov. 2016. [Online]. Available: https://www.sandvine.com/resources/global-internet-phenomena/2016/north-america-and-latin-america.html. [Accessed 4 May 2020].

[25] R. P. Mok, E. Chan and R. Chang, “Measuring the quality of experience of HTTP video streaming,” in *Proc. 12th IFIP/IEEE Int Symp. Integr. Netw. Manage. (IM)Workshops,* Dublin Ireland, May 2011.

[26] D. Z. Rodriguez, R. L. Rossa, E. C. Alfaia, J. I. Abrahao and G. Bressan, “Video quality metric for streaming service using DASH standard,” Vols. 62, no. 3, pp. 628-639, Sep 2016.

[27] T. Hoßfeld, M. Seufert, C. Sieber and T. Zinner, “Assessing effect sizes of influence factors towards a QoE model for HTTP adaptive streaming,” in *Proc. 6th Int.Workshop Qual. Multimedia Exper. (QoMEX),* Singapore, Sep. 2014.

[28] K. Yamagishi and T. Hayashi, “Parametric quality-estimation model for adaptive-bitrate-streaming services,” *IEEE Trans. Multimedia,* vol. 19 no. 7, pp. 1545-1557, Jul. 2017.

[29] F.Wang, Z. Fei, J.Wang, Y. Liu, and Z.Wu, “HAS QoE prediction based on dynamic video features with data mining in LTE network,” in *Sci. China Inf. Sci.*, China, 2017.

[30] Z. Duanmu, K. Zeng, K. Ma, A. Rehman and Z. Wang, “A quality of experience index for streaming video,” *IEEE J. Sel. Topics Signal Process,* Vols. 11, no. 1, pp. 154-166, Feb. 2017.

[31] H. Ebbinghaus, “Memory: A Contribution to Experimental Psychology,” *(H. A. Ruger & C. E. Bussenius, Trans.) ,* ,1913.

[32] A. Rehman, K. Zeng and Z. Wang, "Display device-adapted video quality-of-experience assessment," *Proc. SPIE,* vol. 9394, pp. 9394-1-9394-11, Mar 2015.

[33] T. Hoßfeld,, M. Stufert, M. Hirth, T. Zinner, P. Tran-Gia and R. Schatz, "Quantification of YouTube QoE via crowdsourcing," in *Proc. IEEE Int. Symp. Multimedia, Dana Point, CA, USA*, Dana Point, CA, USA, Dec. 2011.

[34] C. G. Bampis and A. C. Bovik, *Learning to predict streaming video QoE: Distortions, rebuffering and memory,CoRR,,* vol. abs/1703.00633, 2017.

[35] W. Robitza, M. N. Garcia and A. Raake, "A modular HTTP adaptive streaming QoE model--Candidate for ITU-T P.1203 (`P.NATS'),"in *Proc. 9th Int. Conf. Qual. Multimedia Exper. (QoMEX)*, Erfurt, Germany, May/Jun. 2017.

[36] D. Ghadiyaram, J. Pan and A. C. Bovik, "Learning a continuous-time streaming video QoE model," *IEEE Trans. Image Process,* Vols. 27, no. 5, pp. 2257-2271, May 2018.

[37] Z. Duanmu and Z. Wang, "Quality-of-experience for adaptive streaming videos: An expectation confirmation theory motivated approach," *IEEE Trans. Image Process*, vol. 27 no. 12, pp. 6135-6146, Dec 2018.

[38] C. M. Bishop, *Pattern Recognition and Machine Learning,* New York: Springer, 2006.

[39] P. Norvig and S. J. Russell, *Artificial Intelligence: A Modern Approach,* 3rd Edition ed., Prentice Hall, 1995.

[40] E. Alpaydin, *"Introduction to Machine Learning,* MIT Press, 2010, p. 9.

[41] R. D. Cook and S. Weisberg, *Criticism and Influence Analysis in Regression*, vol. 13, 1982, p. 313–361.

[42] G. Hinton and T. Sejnowski, *Unsupervised Learning: Foundations of Neural Computation,* MIT Press, 1999.

[43] R. C. Tryon, *Cluster Analysis: Correlation Profile and Orthometric (factor) Analysis for the Isolation of Unities in Mind and Personality,* Edwards Brothers, 1939.

[44] S. T. Roweis and L. K. Saul, *Nonlinear Dimensionality Reduction by Locally Linear Embedding,* vol. 290, Science, 2000, p. 2323–2326.

[45] L. P. Kaelbing, M. L. Littman and A. W. Moore, "Reinforcement Learning: A Survey," *Journal of Artificial Intelligence Research,* vol. 4, p. 237–285, 1996.

[46] Z. Duanmu, A. Rehman and Z. Wang, "A Quality-of-Experience Database for Adaptive Video Streaming," *IEEE IEEE Transactions on Broadcasting,* Vols. 64, no. 2, pp. 474-487, June 2018.

[47] "DASH Industry Forum. For Promotion of MPEG-DASH," 2013. [Online]. Available: http://dashif.org. [Accessed 5 May 2020].

[48] T. Y. Huang, R. Johari, N. McKeown, M. Trunnell and M. Watson, "A buffer-based approach to rate adaptation:Evidence from a large vidoe streaming service,," *ACM SIGCOMM Comput. Comm. Rev.,* vol. 44, no. 4, pp. 187-198, Feb 2015.

[49] C. Liu, I. Bouazizi and M. Gabbouj, "Rate adaptation for adaptive HTTP streaming," in *ACM Conf. Mutlimedia Syst.,* San Jose, CA, USA, 2011.

[50] L. De Cicco, V. Caldaralo, V. Palmisano and S. Mascolo, "Elastic: A client-side controller for dynamic adaptive streaming over http (DASH)," in *IEEE Int. Packet Vidoe Workshop,* San Jose, CA,USA, 2013.

[51] R. P. Mok, X. Luo, E. W. Chan and R. C. Chang, "QDASH: A QoE-awareDASH system," in *ACM Conf. Multimedia Syst.,* Chapel Hill, NC USA, 2012.

[52] J. Jiang, V. Sekar and H. Zhang, "Improving fairness, efficiency and stability in http-based adaptive vidoe streaming with festive," in *ACM Int, Conf.Emerg. Netw. Exp. Teclnol,* Nice France, 2012.

[53] ITU-Recommendation BT.500-12, *Methodology for the subjective assessment of the quality of television ITU,* Geneva, Switzerland, 1993.

[54] N. Barman and M. G. Martini, "QoE Modeling for HTTP Adaptive Video Streaming–A Survey and Open Challenges," *EEE Access,* vol. 7, pp. 30831-30859, 2019.

[55] C. G. Bampis, Z. Li, A. k. Moorthy, l. Katsavounidis, A. Aaron and A. C. Bovik, "Study of temporal effects on subjective video quality of experience," *IEEE Trans. Image Process,* Vols. 26, no. 11, p. 5217–5231, Nov. 2017.

[56] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hossfeld and P. Tran-Gia, "survey on quality of experience of HTTP adaptive streaming," *IEEE Commun. Surveys Tuts,* Vols. 17, no. 1, no. 1st Quart, pp. 469–492,, 2014.

[57] L. AM, *Measures of Association,* Beverly Hills and London:Sage Publication, Inc, 1976.

[58] D. S. Hands and S. Avons, "Recency and duration neglect in subjective assessment of television picture quality," *Applied cognitive psychology,* vol. 15, no. 6, p. 639–657, 2001.

[59] A. k. Moorthy, L. K. Choi, A. C. Bovik and G. D. Veciana, "Video quality assessment on mobile devices: Subjective, behavioral and objective studies," *IEEE Journal on Selected Topics in Signal Processing,* Vols. 6, no. 6, p. 652–671, 2012.

[60] Y. J. Liang, J. G. Apostolopoulos and B. Girod, "analysis of packet loss for compressed video: Effect of burst losses and correlation between error frames," *IEEE Transactions on Circuits and Systems for Video technology,* Vols. 18, no. 7, p. 861–874, 2008.

[61] A. Rehman, K. Zeng and Z. Wang, "Display device-adapted video quality-of-experience assessment," *Proc. SPIE,* vol. . 9394, 1–11 Feb 2015.

[62] Z. Wang, E. P. Simoncelli and A. C. Bovik, "Multiscale structural similarity for image quality assessment," *Proc. IEEE Asilomar Conf. Signals Syst. Comput,* vol. vol. 2, p. 1398–1402, 2003.

[63] Z. Li, A. Aaron, L. Katsavnidis, A. Moorthy and M. Manohara, "Toward a Practical Perceptual Video Quality Metric," Jun 2016. [Online]. Available: http://techblog.netflix.com/2016/06/toward-practical-perceptual-video.html.

[64] R. Soundararajan and A. C. Bovik, “vidoe quality assessment by reduced reference spatio-temporal entropic differencing,” *IEEE Trans Circuits Syst. Video Technol,* Vols. 23, no. 4, p. 684–694, Apr. 2013..

[65] A. Mittal, M. A. Saad and A. C. Bovik, “A Completely Blind Video Integrity Oracle,” *IEEE Transactions on Image Processing,* vol. 25, no. 1, pp. 289-300, Jan 2016.

[66] K. N. A. R. A. Jain, *Score normalization in multimodal biometric systems,* Vols. 38, no. 12, Pattern Recognition, 2005, pp. 2270-2285.

[67] K. Pearson, *On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling,* Vols. 50, no. 302, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 1900, p. 157–175.

[68] S. Stringer, *Feature Importance – What’s in a name?,* Medium, 2018.

[69] T. C. Urdan, *Statistics in plain English,* Santa Clara University, 2016.

[70] A. N. Tikhonov, A. Goncharsky, V. V. Stepanov and A. G. Yagola, *Numerical Methods for the Solution of Ill-Posed Problems,* Netherlands: Springer Netherlands, 1995.

[71] X. Yan, *Linear Regression Analysis: Theory and Computing,* World Scientific, 2009, p. 1–2.

[72] T. J. Archdeacon, *Correlation and regression analysis: a historian's guide,* University of Wisconsin Press, 1994., p. 161–162.

[73] W. M. Van der Aalst, V. Rubin, H. M. Verbeek, B. f. Van Dongen, E. Kindler and C. W. Gunther, *Process mining: a two-step approach to balance between underfitting and overfitting,* Vols. 9, no. 87, Softw Syst Model, 2010.

[74] P. Domingos, “ A few useful things to know about machine learning,” *Commun. ACM,* Vols. 55, no. 10, pp. 78–87, 2012..

[75] R. Tibshirani, “Regression Shrinkage and Selection via the lasso,” *Journal of the Royal Statistical Society. Series B (methodological). Wiley,* Vols. 58, no. 1, p. 267–88, 1996.

[76] W. contributors, “Minkowski distance,” [Online]. Available: https://en.wikipedia.org. [Accessed 10 Feb 2020].

[77] H. Drucker, C. C. Burges, L. Kaufamn, A. J. Samola and V. N. Vanpink, *Support Vector Regression Machines,* vol. 9, in Advances in Neural Information Processing Systems, 1996, p. 155–161.

[78] V. N. Vapnik and C. Cortes, *support-vector networks,* vol. 20 no. 3, Machine Learning, 1995, p. 273–297.

[79] H. TK, “ The Random Subspace Method for Constructing Decision Forests,,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vols. 20, no.8, p. 832–844, 1998.

[80] L. Li, “Classification and Regression Analysis with Decision Trees,” May 2019. [Online]. Available: Https://towardsdatascience.com/https-medium-com-lorrli-classification-and-regression-analysis-with-decision-trees-c43cdbc58054. [Accessed 16 March 2020].

[81] J. Bergstra and Y. Bengio, “Random Search for Hyper-Parameter Optimization,” *Journal of Machine Learning Research,* vol. 13, pp. 281-305, 2012.

[82] M. C. a. B. D. Moor, “Hyperparameter Search in Machine Learning,” in *The XI Metaheuristics International Conference,* 2015.

[83] J. Pfeffer and R. Ghawi, *Efficient Hyperparameter Tuning with Grid Search for Text Categorization using KNN Approach with BM25 Similarity,* Vols. 9, no. 1, Open Computer Science, 2019, pp. 160-180.

[84] J. Snoek, H. Larochelle and R. P. Adams, “Practical Bayesian optimization of machine learning algorithms,” *Proc. of the 25th International Conf. on Neural Information Processing Systems,* vol. 2, p. 2951–2959, 2012.

[85] J. Kittler and P. A. Devijver, *Pattern Recognition: A Statistical Approach,* London, GB: Prentice-Hall, 1982.

[86] F. Ogwueleka and J. Awwalu, "On Holdout and Cross Validation: A Comparison between Neural Network and Support Vector Machine," *International Journal of Trend in Research and Development,,* Vols. 6, no.2, pp. 2394-9333, 2019.

[87] T. Fushiki, "Estimation of prediction error by using K-fold cross-validation," *Statistics and Computing* , vol. 21, p. 137–146, 2011.

[88] C. Sammut and G. I. Webb, *Encyclopedia of Machine Learning.,* Boston, MA: Springer, 2011.

[89] F. pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion and O. Grisel, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research,,* p. 2825–2830, 2011.

[90] T. E. Olipant, "A guide to NumPy (Vol. 1)," Trelgol Publishing, USA, 2006.

[91] W. McKinney, "Data structures for statistical computing in python," in *Proc. of the 9th Python in Science Conf,* 2010.

[92] J. D. Hunter, "Matplotlib: A 2D graphics environment," vol. 9, no. 3, pp. 90-95, May-June 2007.

[93] G. E. P. B. a. G. M. Jenkins, " Time Series Analysis: Forecasting and Control.," Revised edition. Holden-Day, San Francisco, 1976.

[94] R. L. R. E. C. A. J. I. A. a. G. B. D. Z. Rodríguez, "Video quality metric for streaming service usingDASH standard," *IEEE Trans. Broadcasting,* Vols. 62, no. 3, pp. 628-639, Sep 2016.

[95] M. A. S. a. A. C. B. A. Mittal, "a completely blind video integrity oracle," *IEEE Trans. Image Process,* Vols. 25, no. 1,, p. 289–300, Jan 2016.

[96] Z. Wang, L. Lu and A. C. Bovik, "Vidoe qulaity Assessment based on Structural distortion Measurement," in *In signal Processing:Image Communicaation*, 2004.