Using Machine Learning to Predict and Monitor Quality of Experience in LTE network

John Prashanth Gnaniah Manohar Lincoln, B.Tech.

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science

(Future Networked Systems)

Supervisor: Meriel Huggard

September 2019

Declaration

I, John Prashanth Gnaniah Manohar Lincoln, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

John Prashanth Gnaniah Manohar Lincoln

August 14, 2019

Permission to Lend and/or Copy

I, John Prashanth Gnaniah Manohar Lincoln, agree that Trinity College Library may lend or copy this thesis upon request.

John Prashanth Gnaniah Manohar Lincoln

August 14, 2019

Acknowledgments

I would like to thank my supervisor, Dr. Meriel Huggard for her guidance and encouragement without which this dissertation would have been possible. I would like to thank my parents and my sister for their support throughout the course. Lastly, I want to thank my friends and flatmates for their company.

John Prashanth Gnaniah Manohar Lincoln

University of Dublin, Trinity College
September 2019

Using Machine Learning to Predict and Monitor Quality of Experience in LTE network

John Prashanth Gnaniah Manohar Lincoln, Master of Science in Computer Science
University of Dublin, Trinity College, 2019

Supervisor: Meriel Huggard

There is rising popularity for video services over mobile networks. The service providers require an effective method to meet the user' Quality of Experience. There are a number of existing techniques, but they either lack in terms of accuracy or real-time capabilities. An effective model to monitor Quality of Experience should be objective, accurate, light-weight, and non-intrusive. Machine learning algorithms predicting QoE using network QoS can predict in realtime. In this work, an LTE network simulation is designed for video streaming. During the simulation the network conditions are varied and the network QoS parameter along with QoE of impaired videos measured using Full-reference objective model is collected. Six Machine learning algorithms namely Linear regression, Lasso regression, Ridge regression, Random Forrest, Regression tree and Gradient boosting algorithms were trained and evaluated using the collected data. It was observed that Machine learning algorithms were suitable for QoE monitoring as they were able to perform nearly as good as the Full reference objective model, and at the same time they could perform real-time quality assessment.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Video services over the Internet, especially through mobile networks is becoming extremely popular. Quality of service (QoS) measures the performance of the service provided, while Quality of Experience (QoE) measures the end user satisfaction. QoE is user-centric and subjective in nature, hence quantifying it is a challenging task. Service providers, Content providers, and Network providers are the three key players involved in providing video services. These players are responsible to cater to the quality expectation of the customers. In order to keep the users happy, they require an efficient system to predict and monitor the QoE.

1.1 Motivation

The motivation for this research work is the recent trends in video over mobile networks and the wide scope of using Machine learning to build an effective real-time video QoE prediction tool.

Why Videos in LTE network?

The recent advancements in mobile connectivity is driving the transition from the Information Age to the Experience Age[12]. Users have started opting for in-the-moment data sharing capabilities and visual interactions, which is evident from increasing popularity of applications such as Instagram stories, Snapchat, Skype, Facebook Live, and Periscope. Gradually traditional broadcast television is being replaced by video portals

such as Netflix, Vimeo, and Amazon Prime. These video portals have introduced High bandwidth-consuming Ultra High Definition(UHD) and High Definition(HD) videos. Cisco's Forecast in Figure 1.1 reveals that the sum of all forms of IP videos (TV, video-on-Demand, Internet, P2P, gaming, and video conferencing) is estimated to make 82 percent of global IP traffic by 2022[3].HD videos are estimated to account for 22 percent of global IP video traffic by the year 2022[3].

The global average mobile network connection speed is forecast to triple reaching nearly 28.5 Mbps between 2017 and 2022, resulting in mobile data traffic growth of 46 percent(Figure 1.2)[3]. A crucial factor promoting the increase in mobile speeds during the forecast period is the increasing proportion of LTE mobile connections[3]. With exponential demand for video services, the network and service providers have to improve the video QoE, to avoid customer churn.

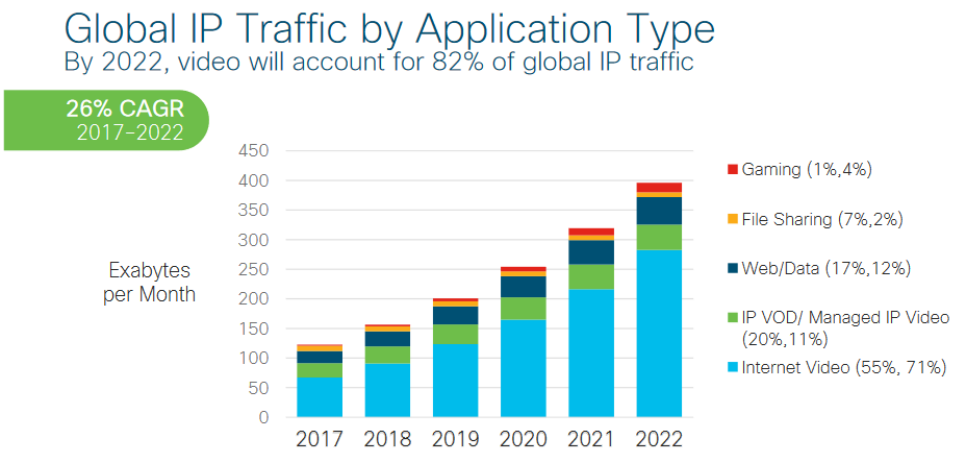

Figure 1.1: Cisco - Global IP Traffic by Application Type[3]

Why QoE monitoring system?

QoE is defined by Qualinet[13] as *the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and/or enjoyment of the application or service in the light of the users personality and current state.* QoE is difficult to measure because of its subjective nature. QoE is affected by many factors called Influence Factors(IF). The IFs have been identified and classified by[14] as Human Influence Factors(age, gender,

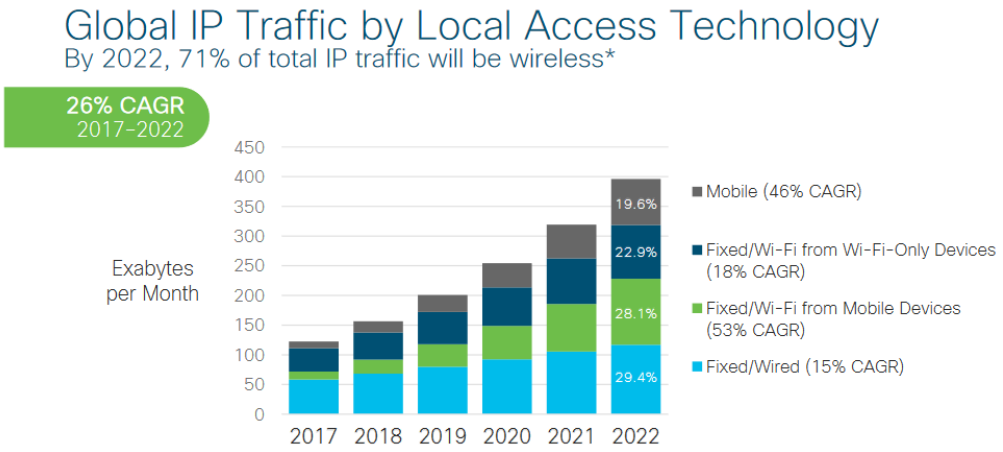

Figure 1.2: Cisco - Global IP Traffic by Local Access Technology[3]

mood,etc),Context Influence Factors(location, space, time of day,etc), and System Influence Factors(device screen size, bandwidth, video encoding, etc).System influence Factors are the most researched, and it is further classified as Content-related(video genre), Media-related (encoding, resolution, fps, etc.), Network-related(delay, jitter, loss, etc.) and device-related (screen resolution, display size, etc).

Studies conducted by [15, 16] shows that there is a relationship between willingness to pay and video QoE. Service providers and network providers have to choose the state of the art QoE metric to monitor the experience of their users to increase profit. There are a number of QoE metrics to choose from, but the most important characteristic of a QoE metric is to work in real-time. Comprehensive research is being done to discover an effective methodology to measure the QoE in real-time using Network related System Influence Factors. The key is to understand the relationship between video QoE and network-related key performance indicators (KPIs) or QoS parameters. Machine Learning techniques are widely experimented to explain this complex relationship between QoS and QoE in real-time.

1.2 Research Objectives

The aim of the research is to build and evaluate various Machine Learning models to predict the QoE of HD video. Rather than synthetically impairing videos, a simulation

environment has to created to transmit the videos over the LTE network. The LTE network has be designed keeping in mind the real-world internet traffic. Parameters associated with the LTE network have to be varied suitably to generate distorted videos which are evaluated using an existing QoE metric, and the corresponding network related QoS parameters has to be captured. Machine learning models to predict the QoE using the QoS parameters has to be implemented. The built models are to be evaluated based on their ability to be real-time and as accurate as the state of the art Full Reference QoE metric. Finally, for each model, the parameters that influence the QoE value the highest(feature importance) has to be determined and analyzed.

1.3 Thesis Road Map

Chapter 2 - Background of this dissertation will cover the required concepts for understanding and building a video QoE prediction System.
Chapter 3 - Related Work of this dissertation consists of some of the previous work done using Machine Learning for QoE.
Chapter 4 - Design and Implementation of this dissertation will covers the design principle taken to build a QoE model.
Chapter 5 - Evaluation and Results of this dissertation consists of the results obtained and evaluation of Machine Learning models.
Chapter 6 - Conclusion consists of the thesis conclusion and future work that can be implemented.

Chapter 2

Background Research

This research work requires a sound understanding of the concepts of Networking, Multimedia Services, Video Quality Evaluation, and Machine Learning. This chapter provides insights on the above mentioned topics along with some tools and frameworks required to design, build, and validate a QoE monitoring system.

2.1 Networking

2.1.1 The Internet Protocol Stack

The Internet Protocol stack also known as TCP/IP is the standard model and set of communications protocols used in the Internet. This model specifies how data moves over the internet by identifying how it should be broken down into packets, addressed, transmitted, routed, and received at the destination. A TCP/IP model is a layered architecture consisting of 4 layers, each stacked on top of one another. The data from the top layer is encapsulated into the payload of the layer below. Figure 2.1 shows the 4 layered TCP/IP model:

- **Application Layer**: Application layer includes the protocols used by end user application for exchanging application data over the network. Examples HTTP, FTP, SMTP etc.
- **Transport Layer**: Transport layer ensure delivery of data. It breaks application message into segments and passes them to the Internet layer. On the receiving

end, the segments are reassembled and passed on to the application layer. Examples TCP and UDP.

- **Internet Layer**: The Internet layer or Network layer is responsible for the logical transmission of data packets over the internet. It identifies the network into which the host machine is connected for message delivery. Example IP.
- **Link Layer**: The link layer ensure transporting information between the Internet layer interfaces of two different hosts on the same link. Each network-layer datagram is encapsulated in a link-layer frame. Examples Ethernet, P2P

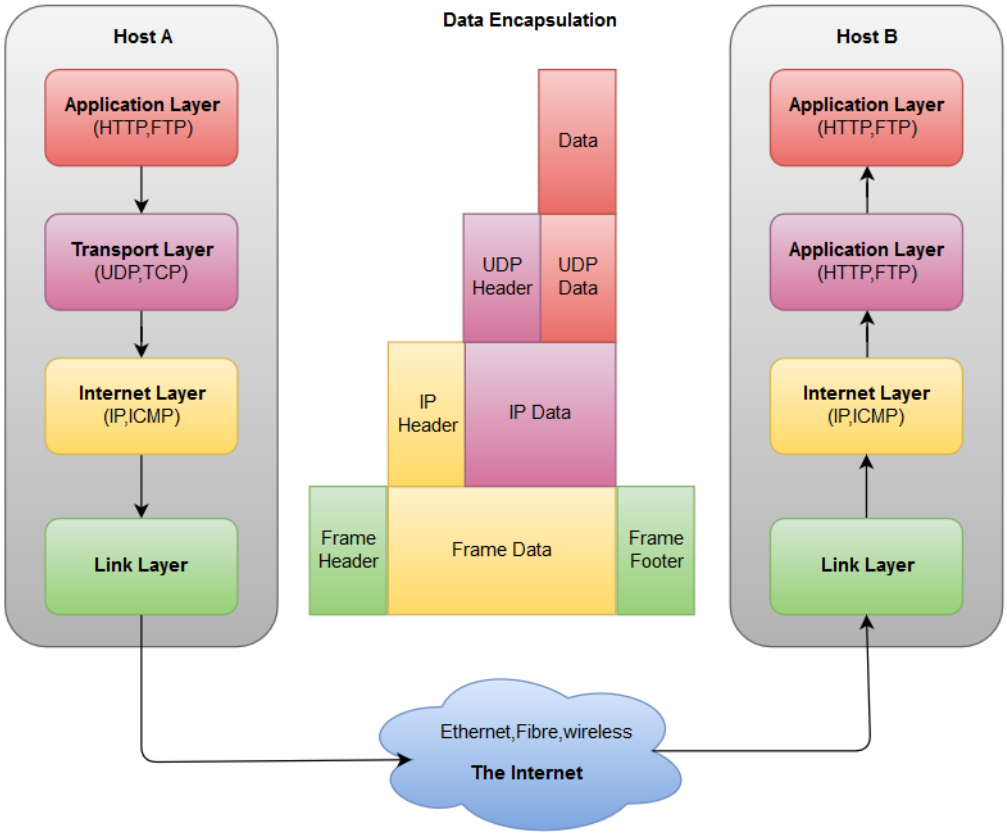

Figure 2.1: TCP/IP Protocol Stack and Encapsulation of application data descending through the layers

2.1.2 Network Quality of Service

The term QoS refers to the probability of the telecommunication network meeting a given traffic contract, and in networking terms, it is the probability of a packet successfully passing between two points in the network[17]. QoS is identified through the QoS parameter. Each layer of the TCP/IP network model will have a different set of QoS parameters. The overall QoS can be calculated by combining the individual QoS parameter of each layer. This research is aimed to use Network related QoS parameters to predict the QoE. Some of the Network related QoS parameters and their descriptions are:

- **Packet Loss percent**: It is the ratio of the number of packets lost to the total number of packets sent during a transmission.
- **Jitter**: Jitter is defined as a variation in the delay of received packets, and it is measured in seconds.
- **Delay**: It is the time taken for a packet to reach the destination, and it is measured in seconds.
- **Throughput**: Throughput is the amount of information received in a unit time. It is measured in bits per second (bps).

2.1.3 Long Term Evolution(LTE) Overview

Wireless communication technology has evolved to a greater extent, starting from 1G to the latest 4G technology.3rd Generation Partnership Project(3Gpp) is a telecommunication association responsible for setting the standards/protocols for wireless communication technology. They created 2G GSM, 3G UMTS, and now 4G LTE. LTE has evolved from Universal Mobile Telecommunication System (UMTS) with the main goal to provide high data rate, good quality of service, low latency, and Packet Switch optimized system[18]. Figure 2.2 shows the architecture of LTE. The high level LTE architecture consists of User equipment(UE), the Evolved UMTS Terrestrial Radio Access Network (E-UTRAN), and Evolved Packet Core (EPC).

User Equipment(UE)

The UE is nothing but the Mobile devices that supports the LTE standard. The UE consists of Mobile Termination(MT) to handle communication functions,Terminal Equipment(TE) to terminate data streams, and Universal Integrated Circuit Card (UICC) to run the Universal Subscriber Identity Module(USIM) application.

E-UTRAN

Evolved Universal Mobile Telecommunications System Terrestrial Radio Access Network (E-UTRAN)is made up of eNodeB, which takes care of the wireless radio communication between the UE and EPC. Each eNodeB is a base station and they are strategically placed all over the network to provide communication service.

The Evolved Packet Core (EPC)

The EPC or System Architecture Evolution (SAE) forms the brain of the LTE system consisting of five nodes[19]:

- **MME**:Mobility Management Entity(MME) is the central control node in the EPC network which is responsible for mobility and security signalling, tracking, and paging of UE.
- **S-GW**:Serving Gateway(S-GW) transports the user traffic between the UE and external networks. It also interconnects the radio access network with the EPC network.
- **P-GW**: Packet Data Network(PDN) Gateway connects the EPC to the outside world.
- **HSS**: Home Subscriber Server(HSS) consists the information of all mobile users along with the subscriber data. It is responsible for authentication and call and session setup.
- **PCRF**: The real-time policy rules and charging in EPC is taken care by Policy and Charging Rules Function(PCRF)

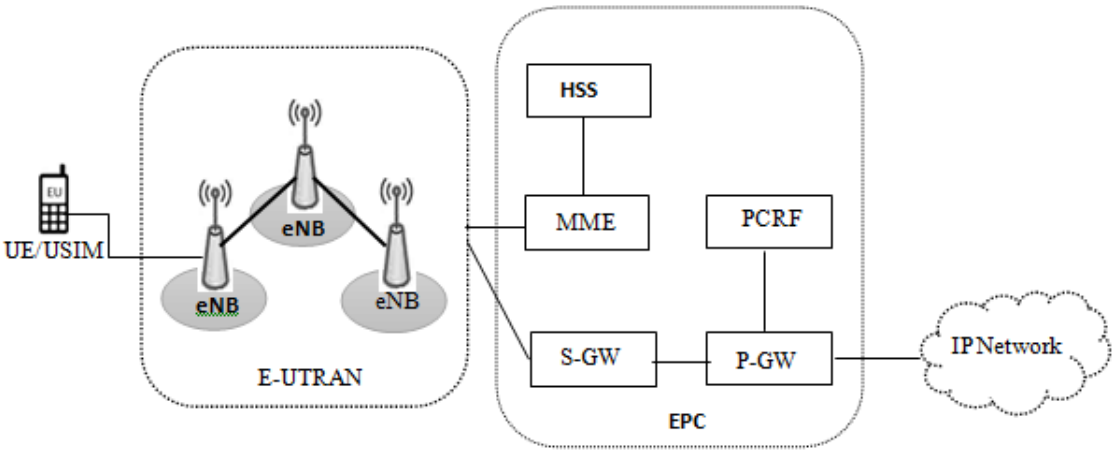

Figure 2.2: LTE Network Architecture [4]

EPS Bearers

EPS bearer is a virtual tunnel between the UE and PDN. An EPS bearer has to cross multiple interfaces as shown in Figure 2.3 the S5/S8 interface from the P-GW to the S-GW, the S1 interface from the S-GW to the eNodeB, and the radio interface from the eNodeB to the UE[5]. QoS in LTE networks is achieved through EPS bearer. EPS bearer aids in identifying the type of traffic and treat them differently based on the required QoS. For example, VoIP traffic has more stringent requirements for QoS in terms of delay and jitter. Each bearer has an associated Class Identifier(QCI) which is characterized by priority, packet delay budget and acceptable packet loss[5]. Table 2.1 details the parameter setting of each QCI value. There can be many bearers acting between the UE and PDN at any point in time. Bearers are classified into different types as shown in figure 2.4:

- **Guaranteed Bit Rate(GBR) Bearer**: Guaranteed Bit Rate Bearers has a predefined GBR associated with it. Dedicated resource is allocated to achieve the GBR value. If a UE Bitrate requirement is more than the GBR then the Maximum Bit Rate (MBR), an upper limit on the bit rate, the parameter is checked before dedicating the requested resource.

- **Non-Guaranteed Bit Rate Bearer**: Non-Guaranteed Bit Rate Bearer do not guarantee any particular bit rate.

- **Dedicated Bearer**: Dedicated bearers can be assigned in addition to the default bearer. This acts a dedicated tunnel suitable for a particular service such as Video, VoIP, or, gaming. It offers both GBR and Non-GBR based bearers(i.e any QCI value).

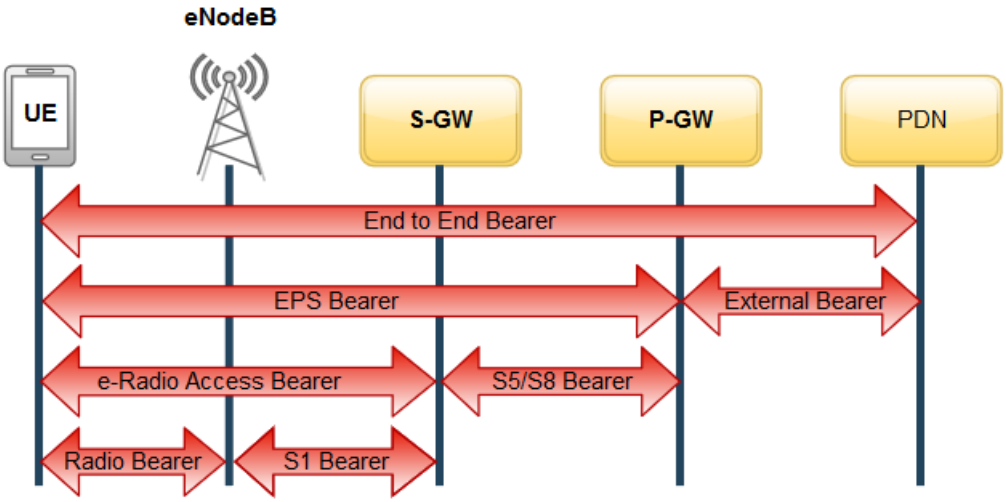

Figure 2.3: The overall EPS bearer service architecture, Recreated with reference to[5]

QCI	Bearer Type	Priority	Packet Delay (ms)	Packet Error Loss Rate	Example services
1	GBR	2	100	10^{-2}	Conversational voice(VoIP)
2	GBR	4	150	10^{-3}	Conversational video (live streaming)
3	GBR	3	50	10^{-3}	Real-time gaming
4	GBR	5	300	10^{-6}	Non-conversation video (Video streaming)
5	Non-GBR	1	100	10^{-6}	IMS Signalling
6	Non-GBR	6	300	10^{-6}	Buffered Video, TCP-based services
7	Non-GBR	7	100	10^{-3}	Voice, Live Video, Interactive Gaming
8	Non-GBR	8	300	10^{-6}	Buffered Video, TCP-based services
9	Non-GBR	9	300	10^{-6}	TCP-based services

Table 2.1: QoS Class Identifier Values[1]

SISO and MIMO

Single Input Single Output(SISO) and Multiple Input Multiple Output(MIMO) are types of Radio Frequency(RF)control systems. SISO is an older technique when compared to MIMO. MIMO is becoming very popular these days. The core difference

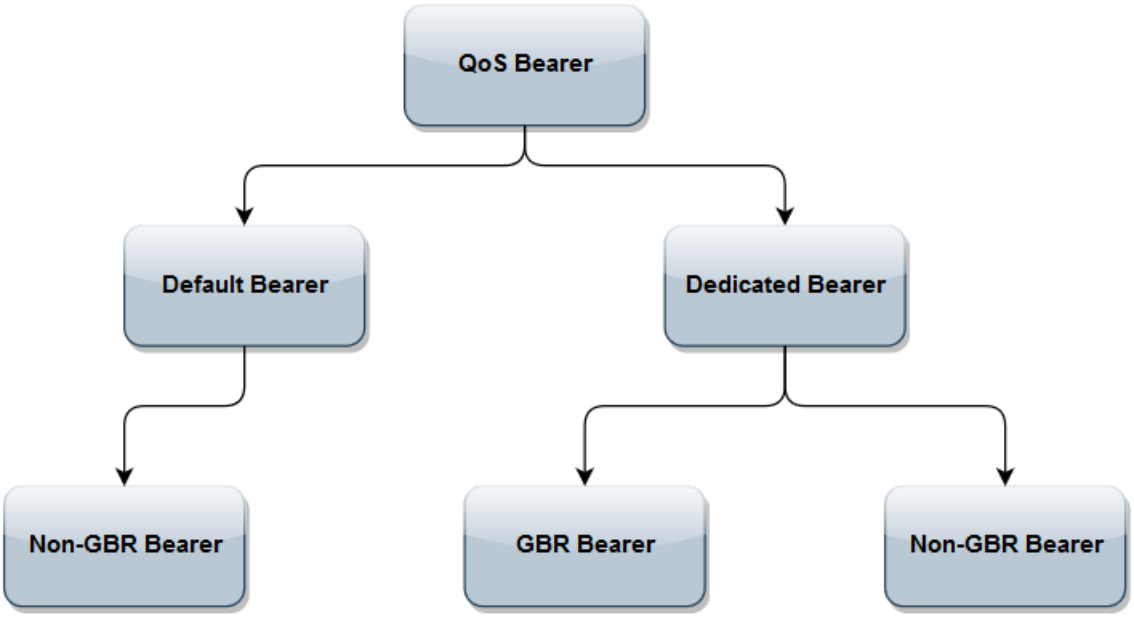

Figure 2.4: Types of QoS bearers, Recreated with reference to[5]

is MIMO uses multiple antennas, while SISO uses just one antenna to transmit and receive data. MIMO is particularly used in LTE mobile networks and they help to achieve better Bit Error rate and high Bandwidth.

2.1.4 Network Simulator

Network simulation is a technique where the behavior of a real-world network is modeled using a software program called Network simulator. The main idea is that if a network can be modeled then the attributes/parameters associated with it could be changed and the corresponding result could be analyzed. Also, Network simulators overcome the difficulty of time consuming and expensive physical implementation of prototypes. Network simulators are extremely popular among the network research and development community. Although network simulators cannot model all the intricate details of the network, a well-modeled simulator will yield meaningful insights close enough to a real-world system. It is important to wisely choose the network simulator for research. NS-2, NS-3, OMNet++, GloMoSiM, OPNET, NetSim, and, AODV are some commonly used network simulator.Out of which NS-3, OMNeT++,

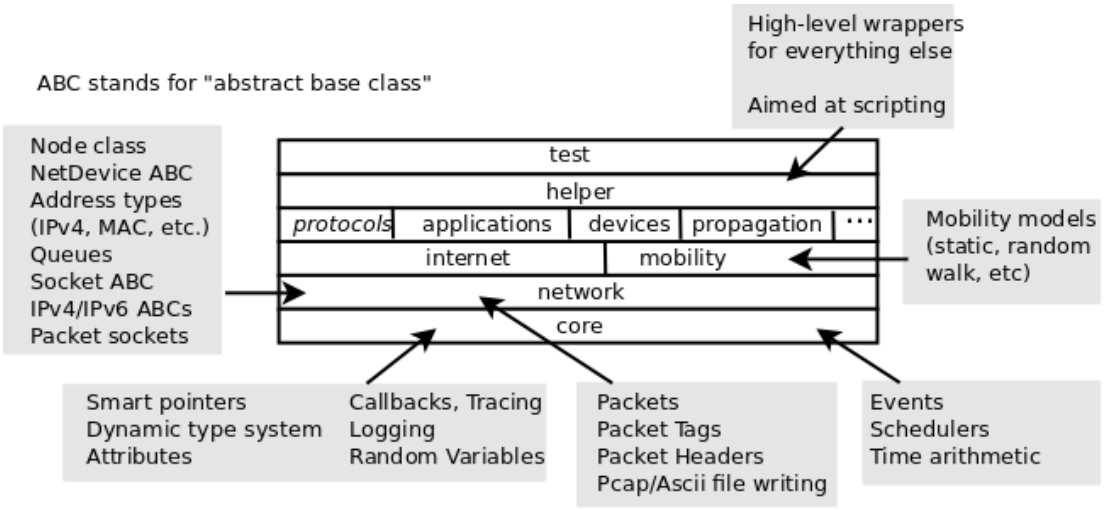

Figure 2.5: Software organization of ns-3 [6]

and GloMoSiM are found to be capable of carrying out large scale network simulations, and NS-3 proved to be best in terms of processing speed and CPU. utilization[20]. In addition, NS-3 provides all necessary features to design and monitor an LTE Video transmission system and it got a good reputation within the networking community.

NS-3 Overview

NS-3 is a discrete-event network simulator built using C++, and it is open source available for commercial and non-commercial use under the GNU GPLv2 license. It is not officially supported by any company, but support is provided through ns-3-users forum (ns-3-users@googlegroups.com). It is built as a system of software libraries that work together and the simulation programs can be written that links these libraries[21]. The modular design of the libraries allows it to be easily combined with external libraries. Developers can either use C++ or Python programming(through exposed API's), but some libraries are not supported in python. For the purpose of this research NS-3 version, 3.29 is used. Installation and configuration instructions are available in [21].

In order to build network simulations using NS-3, it is important to understand the core concepts and abstractions in the system. These abstractions form the basis for any simulation script and any module build on top of NS-3 should follow this. Below

are the 5 key core abstractions in ns3[21]:

- **Node**: The Node represents a basic computing device which provides methods for managing the representations of computing devices in simulations.
- **Application**: Application is the functionality of the Node and they run on the Nodes to drive the simulation. The Node can behave as a Client or Server depending on the type of application installed on it. For example, a UdpEchoClientApplication installed on a Node will request UDP Packets from another node where UdpEchoServerApplication is installed.
- **Channel**: Channels are the media over which data flows in the networks. The primary goal is to manage communication subnetwork and connecting nodes to them. Example of Channels are CsmaChannel (Ethernet) and WifiChannel (IEEE 802.11).
- **Net Device**: Net Device acts as both the software driver and the simulated hardware required to access a Channel. They are installed in a Node to communicate with other Nodes in the simulation via Channels. A Node may be connected to more than one Channel using multiple NetDevices installations. Example CsmaNetDevice designed to work with a CsmaChannel and WifiNetNevice designed to work with a WifiChannel.
- **Topology Helpers**: For building a large simulated network, a number of operations such as the creation of a NetDevice, the addition of MAC address, installation of NetDevice on a Node, Configuring Protocol stack and finally connecting the NetDevice to a channel. Topology helper combines these distinct operations into an easy to use model. Figure 2.5 Shows the software organization of NS-3 and it could be seen how Topology Helper acts as a High-level wrapper for different operations.

LENA NS-3 LTE Module

The ns-3 LTE model is an open-source software library that allows the simulation of LTE networks with an optional feature to include the Evolved Packet Core (EPC).

Figure 2.6 portrays the LTE-EPC simulation model and it can be seen that the architecture consists of two main components:

- **LTE Model**: All the LTE Radio Protocol stacks such as RRC, PDCP, RLC, MAC, and PHY that resides in UE and eNodeB forms the LTE Model. This model is designed to scale up to tens of eNodeBs and hundreds of UE. Each eNodeB can be configured with different carrier frequencies and bandwidths. The model supports a number of scheduler algorithm, which decides which UE should be given the resources (RBs) and how much resource (RBs) should be given to send or receive data. This research will be focused on simulating video over the network by varying the parameters that are associated with LTE model, such as RLC buffer Size, MIMO system, Number of UE, and total Resource Block allocated.
- **EPC Model**: EPC Model consists of core network interfaces and protocols that reside within SGW, PGW, and MME. This model provides end-to-end IP connectivity over the LTE model. It aids in establishing Internet connectivity for UE. It supports the creation of multiple EPS bearers for a UE with different QoS profiles. It can model different application that generates TCP or UDP traffic.

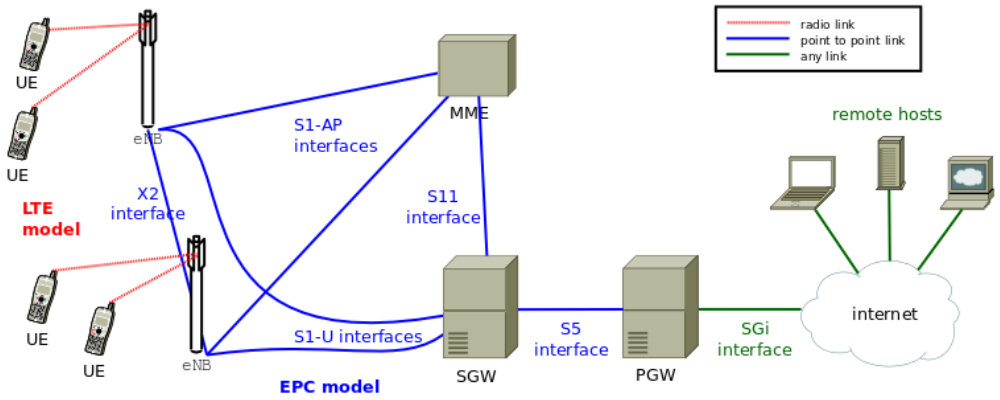

Figure 2.6: Overview of the LTE-EPC simulation model [7]

NS3 FlowMonitor

All Networking related research based on simulation requires a module to measure the set of performance metrics. FlowMonitor is such a module for NS-3 based simulations.

It automatically detects all flows passing through the network by installing probes in the networking devices(switches,routers) and stores in a file[22]. It measures parameters such as bitrates, duration, delays, packet sizes, and packet loss ratio which the researcher requires to the data flow.

NS3 Evalvid module

GERCOM is a North Brazil based research group, and they have built a module for video transmission on NS3 for Evalvid. This module provides features to build a Server which sends video traffic from a video trace file generating a sender dump file and a Client which receives the video traffic to generate a receiver dump file. The dump files created by the simulation are compatible with Evalvid for further processing.

2.2 Video Multimedia Service

Videos are nothing but a series of images displayed continuously. Videos are primarily classified into analog and digital videos based on the type of signal they generate. The Analog video technology is being completely replaced by digital technology due to its ease of storage and transmission. In this research, we focus on the transmission of digital videos.

2.2.1 What makes a Digital video

Video Resolution

Digital videos are made up of digital images. Each image is divided into series of horizontal lines and the lines are further divided into a series of dots, called pixels and each dots intensity and the color is represented by a number [8]. Figure 2.7 shows a low resolution and high resolution image. A low resolution image contains lesser number(240x320) of pixels compared to a High Definition image(1920x1080).Higher the resolution of the image, that make up a video, higher is the perceived quality of video and occupied disk space.

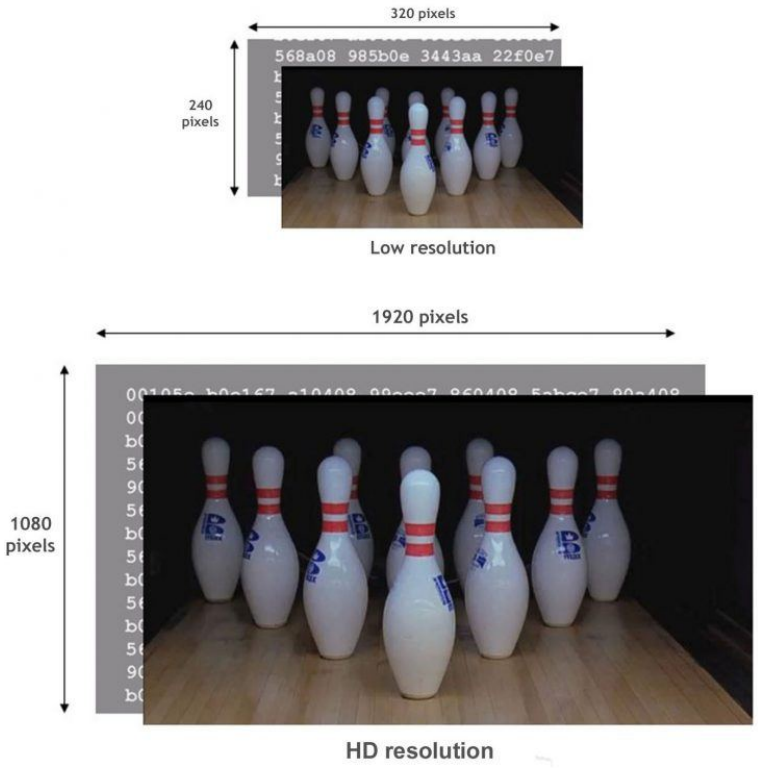

Figure 2.7: Image Resolution comparison [8]

Frame Rate

The number of images/frames displayed in a second is called Frame Rate and is measured in Frames Per Second(fps). The human visual system (HVS), can process 10 to 12 images per second anything beyond it is perceived as a video[23]. Setting higher frame rates to video files results in smoother videos.

Colour space

A colour space is a spectrum of colours that can represent an image. An image rendered using a colour space consisting of 1000 colours will look more accurate than a colour space of just 100 colours. Some familiar colour spaces are RGB, YUV, CMYK, and HSB.YUV has a major advantage over other colour spaces as it can separate luminance(brightness) from chrominance(colours). A Raw YUV video denotes that the file stores video data in raw uncompressed YUV format with constant bitrate.

Video Codec

Uncompressed raw video files consume a lot of disk space making it hard to transmit in a network of limited capacity. Video codec (coder-decoder) is a piece of software which can compress and decompress video files. Some popular video codecs are H.264, H.265, and AVI. Codecs can compress spatially and temporally. In spatial compression, the pixels use information from their neighbors to reduce storage consumption. In temporal compression, frames use information from their neighbors to reduce storage consumption. For the purpose of temporal compression the frames are divided into 3 types:

- **Intra-Frames**: It is commonly known as I-Frames, contains complete image and they are not compressible.
- **Predicted-Frames**: It is also known as P-Frames, is dependent only on previous frames for display. It could achieve better compression than I-Frame.
- **Bidirectional-Frames**:Commonly know as B-Frames are dependent on both prior and subsequent frames for display. These frames can achieve high data compression amount.

Bitrate

Video Bitrate is the rate(bits/sec) at which the Codec outputs data, and it is directly proportional to perceived video quality. Bitrate is divided into Constant bitrate (CBR) and variable bitrate (VBR) based on the variability in the size of each frame.

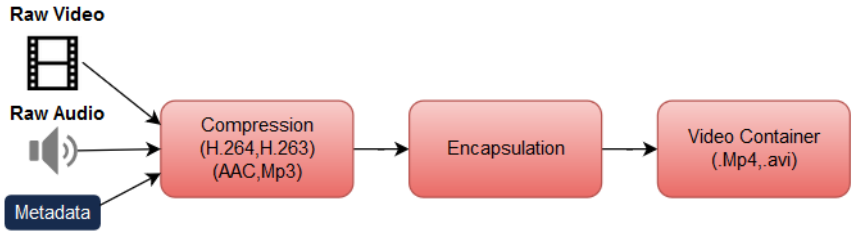

Figure 2.8: Stages of Video Container Format

Video Container

The encapsulation of compressed video(H.264), compressed audio(mp3), and metadata(subtitles) forms a video format container as show in figure 2.8.

2.2.2 Video Transmission

Figure 2.9 shows a typical video transmission path from server to client. The server encodes the raw uncompressed video stream before sending it over the network. The client receives the video stream and decodes to display them to the user. The video received on the client machine is likely to be distorted due to network impairments affecting the viewers' perception of the video quality. Blocking, Blurring, Edginess, and Motion Jerkiness are the common visual distortion that can be found in the received distorted video stream[24]. This research needs a comprehensive tool for handling all video related formatting such as compressing, decompressing, and setting video properties(bitrate, framerate, and frame size).FFmpeg is a free, open-source, command line multimedia framework consisting of a number of libraries and programs for handling video, audio, and other multimedia files and streams[25]. It is used in this research work to manipulate video files.

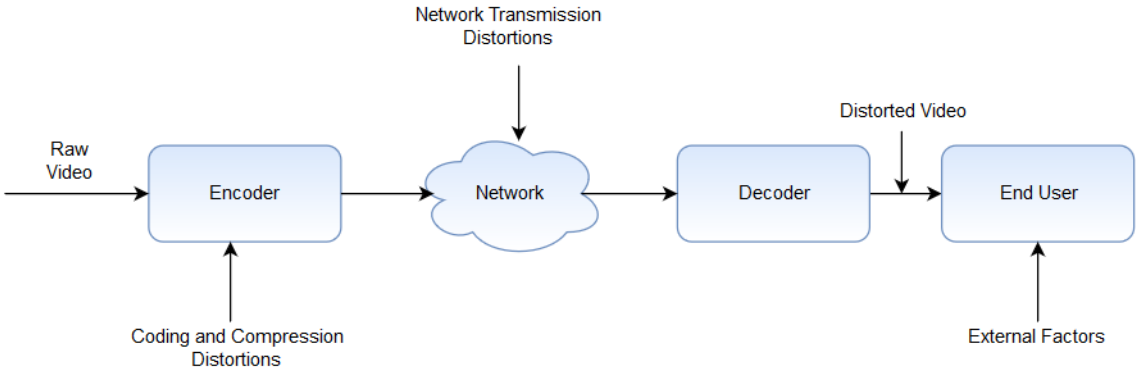

Figure 2.9: Video Transmission Path, Recreated from [9]

A video streaming protocol is a standardized delivery method for breaking up a video into chunks, sending it to the viewer, and reassembling it[26]. Some example of streaming protocols are RTP/RTSP, MPEG-DASH, and HAS. The streaming protocols work along with the transport layer protocol. User Datagram Protocol(UDP)

and Transmission control protocols are the two widely used transport layer protocol to provide end-to-end message transfer service. TCP is a connection-oriented protocol, which uses acknowledgment,re-transmission and congestion control mechanisms to ensure reliable data delivery. On the other hand, UDP is a connectionless protocol which does not promises 100 percent delivery of messages, but it has lesser delay. UDP is more suitable for live video streaming where delay is not acceptable, while TCP is used when small delay does not make a big difference. This research is focused on using RTP on top of UDP. Before sending a video file over a network it has to be prepared for RTP streaming. This is achieved by adding a hint track in the Video container format, which describes how to packetize the frames for the transport with RTP. GPAC is an open-source software dedicated to rich-media and broadcast technologies consist of MP4Box which is used in this research work to prepare a video file for RTP streaming.

2.2.3 Evalvid

Evalvid is a framework and tool-set for evaluation of the quality of video transmitted over a real or simulated communication network[27]. It measures QoS parameters of the underlying network(loss rates, delays, and jitter) and standard video quality metrics(PSNR and SSIM)[27].In addition, it provides a toolset to generate trace file from a hinted video file and recreate the distorted video from the received dump files. It supports H.264, MPEG-4, and H.263 video Codecs. In this research the below two tools of Evalvid version 2.7 is used:

- **mp4trace**: mp4trace takes input an Mp4 video file containing an RTP hinted track and Maximum Transmission Unit(MTU). It converts the video file into trace file containing information about Frame number, Frame Type(I/H, P, B), frame size(bytes), number of packets, and timestamp. The structure of an evalvid trace file is shown in table 2.2. The MTU size is used to divide a frame into packets. MTU of 1460 bytes for a frame of size 2233 bytes will result in 2 packets. The network simulator uses the trace file for simulation, not the actual video file.

- **etmp4**: etmp4 requires the video trace file(Table 2.1), actual Mp4 video file, and sender and receiver dump files to calculate the QoS parameters and recreate the distorted video file. The sender and receiver dump file are obtained from

Frame No.	Frame Type	Frame Size	Packet Count	Timestamp
1	H	119285	82	0.001
2	P	2233	2	4.4
3	P	2224	2	4.44
4	P	61517	43	0.119

Table 2.2: Structure of a trace file

4.1229 id 1 udp 1460
4.1229 id 2 udp 1460
4.1229 id 3 udp 1460
4.1229 id 4 udp 1460

Table 2.3: Structure of a Dump file

network simulation, and it contains information on Packet size, Packet type, Packet Number, and the timestamp as shown in Table 2.3.

2.3 Quality of Experience

Initially, the researchers believed that setting appropriate QoS parameters will increase the end user video quality.Especially optimization of network bandwidth allocation[28, 29], video compression, and Congestion Control[30] to enhance video quality[31, 32] was studied.In [29] a system to dynamically allocate network bandwidth through prediction of future network traffic is designed, to cater high quality variable bitrate video transmission. [31] presents an analysis of the trade-offs between video bitrate and distortion to be considered while designing a video coder. QoS parameter never evaluated the video quality from the end users' perspective, so a concept of QoE was introduced. Quality of Experience(QoE) is defined as the overall acceptability of an application or service, as perceived subjectively by the end-user[33]. QoE of a video is very hard to measure and they can be measured subjectively and Objectively.

2.3.1 Subjective Video QoE Assessment

Subjective Video QoE Assessment is a direct approach where human assessors watch the test video and give their scores for the quality. Scores obtained from the subjective

test are often considered as the ground truth for validating the performance of the objective quality model. The scores given can either be categorical(example Bad, Poor, Fair, Good, and Excellent) or Numerical grading(1 - 100) There are several ways the tests are conducted below are the two most popular techniques:

- **Double-Stimulus Impairment Scale (DSIS)**: In DSIS, the assessors are first presented the original source video then presented the distorted video and they only grade the processed video, based on the knowledge of the source video[34]. The final score is calculated using a technique called Difference Mean Opinion Score (DMOS). DMOS is calculated as the average of the arithmetic difference between the grades given to the processed video and the grades given to the source video[9].

- **Method,Single-Stimulus (SS)**:In SS, the assessors are presented only the processed videos.Each video can either be shown once or thrice.Final score is processed using Mean Opinion Score (MOS).MOS is nothing but the average score for a test video given by a number of users

2.3.2 Objective Video QoE Assessment

Subjective tests are not always the best method to measure QoE especially when there is unavailability of human assessors. To overcome this problem researchers developed objective measurement techniques. Objective QoE models identify a mathematical model that outputs a metric that is well correlated with the subjective test results. These methods can provide a reasonably accurate measurement of QoE, but require careful designing and implementation. Objective models can be broadly classified into five types, as shown in Table 2.4, based on the data they input. Apart from these, they can also be classified based on the amount of reference information they require as Full Reference, Reduced Reference and, No Reference models.

Full Reference (FR)

FR models measure the visual quality degradation in a distorted video with respect to a reference video[33]. To compute a FR metric the complete reference video and distorted videos are required. Usually, every pixel in every frame of reference video

S.No	Layer	Input data to compute QoE	Primary Application
1	Media layer Model	Video signal	Quality benchmarking
2	Parametric Packet layer Model	Packet header	In-service nonintrusive monitoring (e.g. network probe)
3	Parametric Planning Model	Quality design parameters	Network planning, terminal/application designing
4	bitstream layer model	Packet header and payload	In-service nonintrusive monitoring (e.g. terminal-embedded operation)
5	Hybrid Model	Combination of any of the above model	In-service nonintrusive monitoring

Table 2.4: Classification of objective video QoE on type of Input[2]

is compared to the distorted video. Many studies have shown that FR metrics are well correlated with the human vision system(HVS) [35][36], hence it is often used to benchmark other models. Figure 2.10 shows the architecture of the full reference model. Some of the widely used FR metrics are Peak Signal to Noise Ratio (PSNR), Structural Similarity Index (SSIM), Video Multimedia Assessment Fusion (VMAF), and Video Quality Metric (VQM).

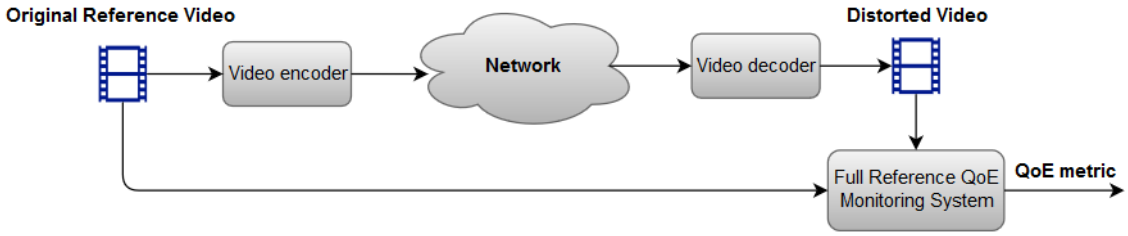

Figure 2.10: Full Reference Model

Reduced Reference (RR)

RR models use partial information extracted from the reference videos. A number of features are extracted from the reference and/or the distorted test video, and these features are compared to computer RR metric[37]. Figure 2.11 shows the architecture of the reduced reference model.RR models are mainly Packet Loss Visibility(PLV)[[38]] and natural scene statistic (NSS)[39] based models.

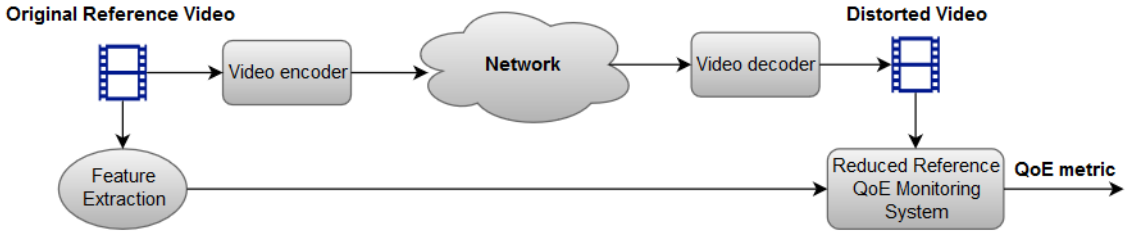

Figure 2.11: Reduced Reference Model

No Reference(NR)

NR models evaluate just the distorted video. These model does not have any dependency on the reference video, hence can meet the demand of real-time QoE monitoring system. Also, NR models are less accurate when compared to RR of FR models due to the lack of the original sample. Figure 2.12 shows the architecture of No reference model. Example Natural Image Quality Evaluator(NIQE), Spatial perceptual information(SI), and Temporal perceptual information(TI).

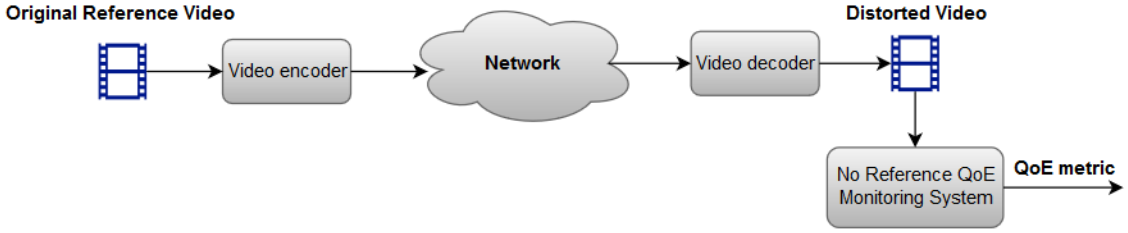

Figure 2.12: No Reference Model

Media layer Model

The input to this model is the video signal. This can directly or indirectly model human visual processes. These models just require the media signal and do not require any information about the system such as packet loss rate or codec type. Media Layer Model can fall into any of FR, NR and RR categories. An Examples of a medial layer model is ITU-T Recommendations J.144[2],

Parametric Packet layer Model

In this model, just the packet-header information is used for QoE estimation. Specifically, the information gathered from IP and RTP layers. This is lightweight and causes lesser computational load, hence these can be used for in-service non-intrusive QoE measurement. These models are extensively researched due to its non-intrusive nature.

bitstream layer model

These models use the payload bitstream information for QoE prediction. For example, the video spatial complexity can be obtained from DCT coefficient in MPEG types. Figure 2.13 shows the comparison of the models with respect to the protocol stack.

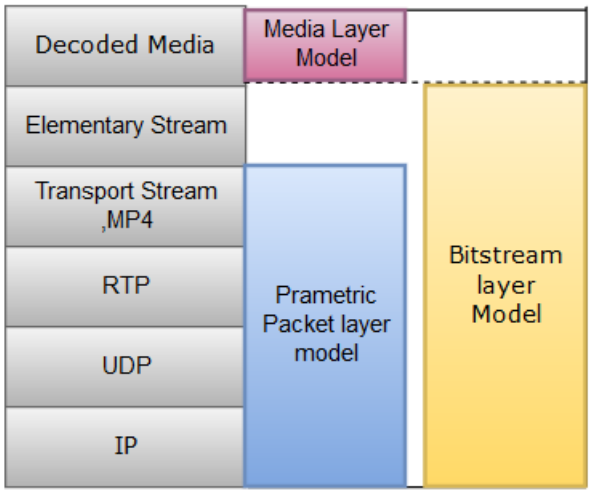

Figure 2.13: Bitstream model in comparison with media and packet layer models, Recreated from [2]

Parametric Planning Model

These models take in the quality planning parameters for the networks and terminals for QoE estimation. This approach is implemented in [40, 41]

Hybrid Model

This model is just a combination of any of the above mentioned models.

2.3.3 Choosing the Benchmark QoE Metric

The main objective of this research work is to build a QoE model using Machine Learning that can be used in real-time. It is necessary to have a ground truth QoE metric for building and validating the performance of the machine learning based models. As explained earlier subjective QoE metrics are expensive and time-consuming, instead an objective QoE metric has to be chosen. Among the Objective QoE models, the Full reference models have high correlation to the subjective QoE metric[42]. In QoE metrics PSNR, SSIM and VQM for video of different bitrates are compared against subjective MOS score, and it is found that VQM is highly correlated with MOS. Several researchers have compared objective QoE models with subjective QoE models, and it is identified that VMAF, VQM, and SSIM-Plus are the well correlated metrics[43, 42, 44, 45]. Even though SSIM-Plus is claimed to be the most accurate model, it cannot be used as it is proprietary[44]. Among VQM and VMAF there is no clear winner because different data sets claim different metric to be the best[46]. VMAF is chosen as the benchmark QoE metric for the purpose of this research for the below reasons

- VMAF is a recently developed video quality assessment tool.
- It is the engine behind Netflix's video encoding stacks.
- It is enhanced by Machine learning, so it keeps improving over time and can be trained for better predictive value.
- VMAF evaluation in [45] shows strong correlation between subjective Mean Opinion Score and the computed objective VMAF score with a correlation of 0.948.
- VMAF is highly appreciated and recommended for the production deployment by the streaming media community[47].

Video Multimethod Assessment Fusion(VMAF)

VMAF is a full-reference perceptual video quality metric developed by Netflix. The core idea is that each elementary metrics may have their own strengths and weaknesses with respect to the source content characteristics, type of artifacts, and degree of

distortion[46]. As the name suggests this metric is a fusion of these elementary metrics into a final metric. Machine learning model Support Vector Machine (SVM) regressor is used for the fusion process. The 4 elementary metrics used are Visual Information Fidelity (VIF), Detail Loss Metric (DLM), Mean Co-Located Pixel Difference (MCPD), and Anti-noise signal-to-noise ratio (AN-SNR). Typically the value of VMAF ranges from 0 to 100 with 100 being quality similar to the reference. VMAF can clearly differentiate the video of different resolution, it's values are low for 180p streams and 98+ for 1080p. The VMAF is made open source, and it can be retrained incrementally using different video datasets for improved accuracy.

2.3.4 MSU Quality Measurement Tool

In this research work, a tool to measure video Quality metric for evaluation and modeling QoE is required. MSU Video Quality Measurement Tool Pro Console Edition 11.1 is used, which offers a variety of Full reference and No reference metrics such as PSNR, VMAF, SI, TI, NIQE. This tool is propriety, but a demo version is made available for non-commercial use. It has both command line and GUI versions and for the purpose of automating and batch processing the command line version is used.

2.4 Machine Learning

Machine learning(ML) is a subset of Artificial intelligence which provide a system the ability to learn automatically without any explicit programming. Machine learning algorithms build a mathematical model based on input data, known as training data, in order to make predictions or decisions. In simpler terms machine learning is to make machines learn from the data just like humans learn from the experience. Machine learning is largely used in prediction. Using the model build on the past data, prediction on newer data can be made. Machine learning is capable of identifying complex patterns in data for predicting video QoE. Machine Learning models are broadly classified into three types Supervised learning,Unsupervised learning and Reinforcement learning.

Supervised Learning

In supervised learning, the models are provided with training data that are labeled (desired output). The model will identify a function(f) to map input variable(x) to output variable(y), that is $y = f(x)$. This mapping can be used to predict the output on new input data. In our case, the inputs will be the QoS parameter and the output will be the QoE metric. Supervised learning models are further classified based on the nature of the output into two categories:

- **Regression**: In regression problems, the output variable to be predicted is continuous in nature. For example prediction of housing price in a region. Some popular algorithms are linear regression, regression trees, support vector regression, random forest regression, and gradient boosting regression.

- **Classification**: In classification problems the output variable to be predicted is categorical. For example, predicting if an image is a cat or not. Some popular algorithms used for classification are logistic regression, support vector classification, decision trees, artificial neural network.

Unsupervised Learning

In unsupervised learning the models are provided with training data that do not have any labels(desired output). The goal of unsupervised learning is to discover hidden patterns in the input data. These algorithms are further classified into:

- **Clustering**: Clustering algorithms will group a set of data points together, such that the element in the same group will behave similar to one another. For example to identify customer segments. Some widely used algorithms are K-means clustering, Hierarchical clustering, Principal Component Analysis, and Singular Value Decomposition

- **Association**: Association algorithms establishes association among the data points. It is used to determine interesting relationships between variables. For example to perform market basket analysis. Some popular association algorithms are Apriori,FP-growth, and Eclat

Reinforcement Learning

In Reinforcement learning model, an agent learns to behave in an environment using trial-and-error mechanism. The agent performs actions which result in change in the state of the environment. The agent is rewarded based on the nature of change caused. Over time the agent dynamically learns to make perform actions to receive high rewards. Reinforcement learning is suitable for modeling complex games such as chess. Some popular algorithms are Markov decision process(MDP), Q-learning, and Deep Q Network (DQN).

2.4.1 Choosing Machine Learning Models

Machine learning models based on their ability to explain the relationship between input and output are divided into white-box models and black-box models. White-box models can clearly explain how they produce the results and to what extent does an input feature influence the output. On the other hand, black-box models are very complex to explain and cannot result in any feature importance. From the wide variety of Machine learning models, it is important to choose the right algorithm to QoE prediction. Figure 2.15 shows the flow diagram to pick the appropriate category of machine learning model based on the available input data. The objective of this research is to build a machine learning model on the Network QoS data generated through simulation of video over LTE network to predict the benchmark VMAF and to identify the QoS predictability. VMAF is continuous variable ranging from 0 to 100[46]. White-box regression Machine learning models are suitable for addressing the above problem statement. Three tree based algorithms: Regression Trees, Random Forrest regression, and Gradient Boosting regression and three linear model algorithms: Linear regression, lasso regression, and Ridge regression are chosen for this research work.

2.4.2 Machine learning algorithms

Linear Models

Linear regression, lasso regression, and ridge regression are linear models where the core assumption is a linear relationship between the input(X) and output(Y). All these three algorithms use the training data set to identify a function $h_0(x)$ between the

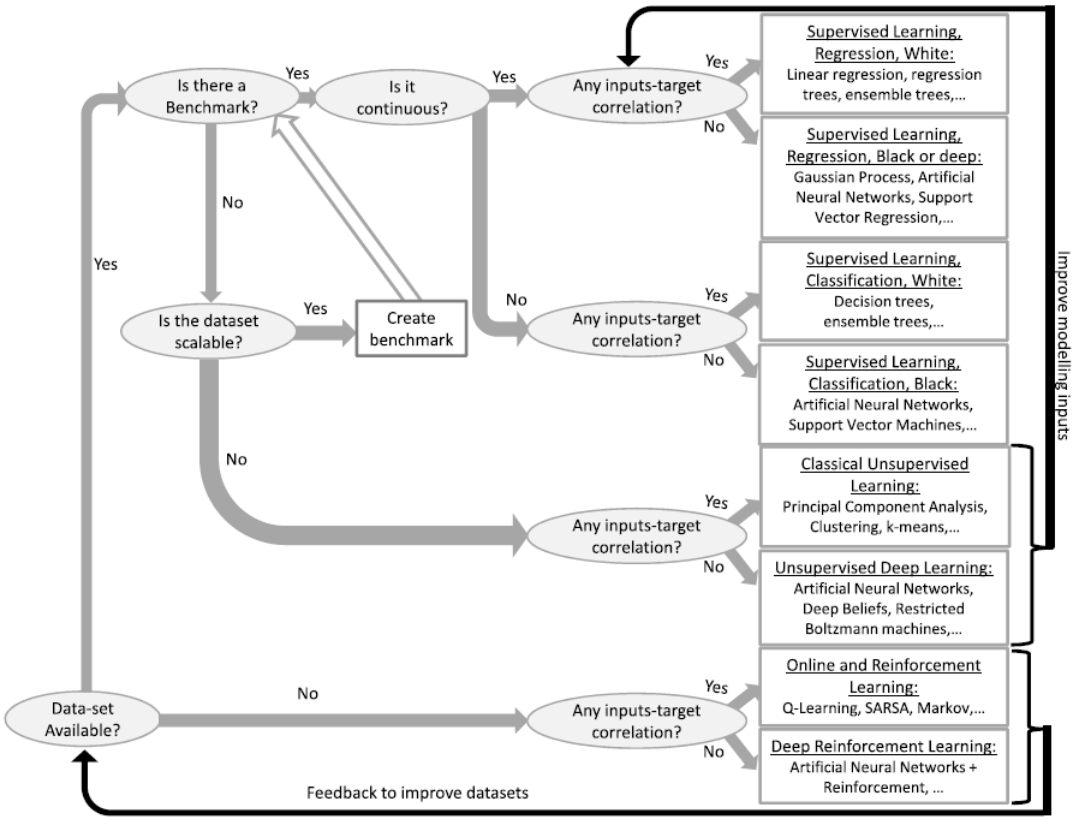

Figure 2.14: Flow diagram of which Machine Learning method to use depending on the available data[10]

input and output such as the cost function $J(\theta)$ is minimized. The hypothesis of these models is shown in equation 2.1. Here θ is the parameter for optimization, x is the input variables and $h_0(x)$ is the predicted value. These models implement an optimization algorithm(gradient descent), which identifies the right set of θ's, such that the cost function is minimized. In the cost function $y^{(i)}$ represent the output variable. Gradient descent is an iterative algorithm to identify the lowest region(minimization) of the cost function. The only difference between these algorithms is the choice of the cost function. Ridge and lasso are also called as regularization algorithm. In general machine learning algorithms tend to over-fit the training data if the complexity is increased. Over-fitting happens when the model's performance in the training set is much higher than that of the test set. Regularization is a technique to reduce

overfitting by constraining the parameter θ. Equations 2.3 and 2.4 represent the cost function of Ridge and lasso regressions respectively. λ is the hyperparameter which can be controlled. Ideally, a grid search cross-validation is done on various values of λ to find the best hyperparameter.

$$h_0(x) = \theta_0 + \theta_1 x_1 + \theta_1 x_1 + ...\theta_1 x_n \tag{2.1}$$

$$J(\theta) = 1/m \sum_{i=1}^{m} (h_0(x^{(i)}) - y^{(i)}) \tag{2.2}$$

$$J(\theta) = 1/m \sum_{i=1}^{m} (h_0(x^{(i)}) - y^{(i)}) + 1/\lambda \sum_{j=1}^{n} \theta_j^2 \tag{2.3}$$

$$J(\theta) = 1/m \sum_{i=1}^{m} (h_0(x^{(i)}) - y^{(i)}) + 1/\lambda \sum_{j=1}^{n} |\theta_j| \tag{2.4}$$

Tree based models

Regression Trees, Random Forrest regression, and gradient boosting are supervised learning algorithms based on the tree data structure. Regression tree algorithm forms the basis for random forest and gradient boosting. Regression Tree algorithm inputs features(X) and labels(Y), and it consists of two steps[11]:

1. During the training of the model, the feature space is divided into a set of possible value for X_1 to X_n into j distinct and non-overlapping regions R_1 to R_j.

2. For predicting the values, if any input observation(X) that falls into the same region R_j, then the prediction is simply the mean of the response(Y) value of the training observations. Some version of the algorithm builds a linear regression model for each of these regions. Figure 2.15 shows the 3 splits of feature space of an input variable(year, hits) and output variable(salary). For a year value greater than 4.5 and hits greater than 117.5, the prediction will the mean of the region R_3.

Regression tree iterates through different features to identify the best split. The best split is identified by reducing the mean square error(mse) for every split. Regression trees tend to overfit quickly and this can be avoided using Pre-Pruning or

Post-pruning techniques. Pre-pruning does not allow overfitting by having a stopping criterion, while post-pruning allows the full growth of tree then overfit branches are removed. Random Forrest and Gradient Boosting are ensemble models. An ensemble model combines different models to increase its predicting power. Random Forrest uses a Bootstrap Aggregation or bagging technique where multiple regression trees are modeled by randomly selecting a sample of features and datapoint from the dataset. This reduces overfitting and improves the accuracy of the final model. Gradient boosting is an ensemble using boosting. Boosting converts a number of weak leaner to a strong learner. The core idea is to minimize the loss function(mse) in each iteration of tree building. Bagging is used to reduce overfitting of high variance models, while Gradient Boosting is used to increase the power of high bias model.

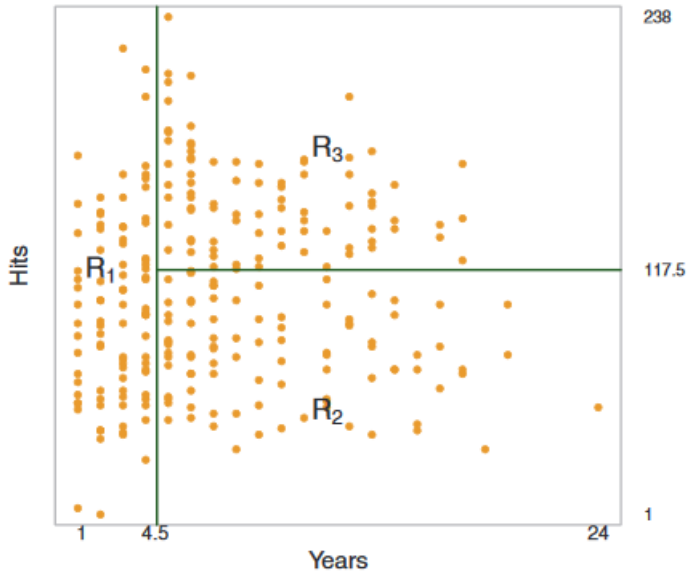

Figure 2.15: Three-region partition for features Hits and Year [11]

2.5 Summary

This chapter covered topics required for building a machine learning based video QoE prediction system using network QoS parameters of video transmitted over LTE network. A set of videos files have to be prepared for RTP streaming and then converted to

Tools/Frameworks	**Functionality**
FFmpeg	Video format convertion
GPAC's Mp4box	Prepare video files for RTP protocol
Evalvid's mp4trace	Coverting video to trace file
NS-3 LENA LTE	LTE simulation
NS-3 GERCOM Evalvid	Video transmision
NS-3 Flowmonitor	Measure packet level QoS
Evalvid's mp4trace	Recreate distorted video from network dump
Evalvid's mp4trace	Measure Frame level QoS
MSU Quality Measurement Tool	Compute VMAF,SI,TI,NIQE
Python(pandas,sklearn,numpy,etc.)	Data cleaning,Feature generation and ML Model building

Table 2.5: Tools/Frameworks used in the research work and their use-case

trace file format for simulation. Machine learning models require input data(Network related QoS parameters) and output data(QoE metric) for model building. Network related QoS parameters are created by probing the video simulation. VMAF is chosen as benchmark QoE metric and it is computed using the reference video file streamed by the sender(video server) and the distorted video obtained by the receiver(end user). Six Machine learning model has been chosen to be trained on the generated dataset.Then the models will be evaluated for performance and to understand the feature importance. The list of tools required fro Video preparation, network simulation, and QoE metric computation is shown in table 2.5.

Chapter 3

Related Work

This research work is focused on building a non-intrusive model for predicting the quality of Experience of video transmitted over LTE using Machine Learning. The previous chapter covered some of the classifications of objective QoE assessments namely based on the type of input data and amount of reference information. Parametric Packet layer model, Bitstream model, and Hybrid models are non-intrusive, table 2.4, and can be used for this work. This chapter presents some of the interesting previous work done in this area.

For over 20 years researchers have been trying to build an effective QoE assessment system. Many authors claim that machine learning is a great tool towards achieving it. Picking the right class of ML algorithms and understanding which parameter to input to the algorithms is an open research topic. Previous work mainly studies QoE prediction for video quality degradation that occurs due to video compression and/or video transmission. For experimentation video transmission degradation can be caused either synthetically, using simulator[48], or using emulator[49]. This section will mainly discuss some of the existing work based on the input type(Section 2.3.2), type of ML algorithms, the reason for video distortion, and Benchmark QoE used. Earlier in 2002, Gastaldo et al.[50] proposed a methodology using circular backpropagation (CBP) neural network based on bitstream layer parameters to study the video distortion occurring due to video compression. This model was tested and benchmarked against subjective MOS score showed the validity of the approach. Shahid et al.[51] built a two-layer feedforward artificial neural network using 43 bitstream layer features to model com-

pressed videos and benchmarked against PSNR, VQM, VIF, and PVQM. White-box ML algorithms which allow determining the importance of the input parameter such as decision trees[52] and Genetic programming based symbolic regression[53] have been researched for their compatibility for QoE prediction. Apart from bitstream parameters the network layer parameter or network QoS seems to be promising fit for model building. Omneya et al.[54] have identified the existent of exponential relationship between network QoS Packet loss or packet order with QoE PVQ and PSNR using simple linear regression. This existence of exponential relationship was further confirmed by the works done in [55, 56] which were benchmarked against subjective MOS values. Multiple linear regression models built using Packet loss rate, Frame rate, Bandwidth, Round trip time, and Jitter is proposed in [57] and it was identified be highly accurate and caused low computational load. Haiqing et al.[58] and Victor et al.[59] proposed a Multilayer Feedforward Neural Network based on gradient descent algorithms. These two models confirms that network QoS parameters such as Delay, Jitter, Loss ratio, Bandwidth are extremely useful in predicting the QoE. Network layer model does not use the information in the payload, hence a hybrid approach as mentioned in the previous chapter was explored by many authors. Torres et al.[49] had applied 9 different machine learning algorithms, classified as white-box and black-box, on a feature set consisting of bitstream and network QoS. The build models are benchmarked against VQM and it was observed that ensemble regression tree models performed the best with 91 percent Pearson correlation. Except for the work done by Vega et al.[10], none of the above discussed models consider the real-world networking conditions for generating the impaired video. They all generate video distortion synthetically. A simulator or emulator can be used to model real networking conditions. Network emulators are hardware devices that can be used for testing and prototyping networking research. They consume a lot more time for setting and configuration when compared to network simulators, hence in our research, a network simulator is used for creating video impairment. To our knowledge Khan et al. [48] was the first to use simulation(NS-2) for sending H.264 videos over UMTS networks for generating distorted video. He modeled a linear regression using network QoS and bitstream data.

Since this dissertation work is focused on modeling Network QoS for prediction of QoE for video over LTE network, understanding some of the previous work done using LTE is important. There are very few authors who had worked in the QoE moni-

toring of video in LTE. Yuan et al. [60] implemented a probabilistic neural network (PNN) for measuring the QoE of video applications using packet loss, delay, jitter, and throughput. 70 distorted videos were created using the network simulator(OPNET) by varying path loss and fast fading and the videos were subjectively scored to compute MOS values. The PNN model built is said to outperform the PSNR metric. Takahiro et al.[61] and proposes a cross-layer downlink scheduling algorithm which improves the QoE. In [62] the authors have conducted the experiment in a test lab consisting of LTE networks which can be controlled for impairments and an analytical approach was used for the QoE estimation. The important thing to observe in this work is the parameters varied: Fading, Round Trip Time(RTT), Signal-to-Interference-plus-Noise Ratio(SINR), and the number of users. In [63] a cross-layer downlink scheduling approach is proposed, to maximize network resource utilization and QoE. The parameters that were varied are number of users, random mobility model, and different modulation schemes. In [64] a single video file was simulated under 24 different conditions using NS-3 by varying Number of UE, Transmission Mode, and Resource Blocks to create impaired video dataset. Subjective MOS scores were calculated for these 24 video sequence, and a multi-layer feedforward ANN model was build using delay, jitter, and loss. The work was concluded that there was no statistically significant difference in the scores of collected and predicted MOS justified by t-test. This work is taken as a reference for this thesis work. The next step in this dissertation is to select the list of LTE network parameters that can be used to created different networking condition for the video transmission.

Chapter 4

Design and Implementation

The aim of this research is to design and implement a Machine Learning based video Quality of Experience monitoring system. Previous chapters discussed some of the approaches used earlier to achieve it. From background research, it was decided to build a supervised Machine Learning Model using the Network QoS parameter for predicting the chosen benchmark QoE, VMAF. This requires a labeled dataset consisting of the network QoS parameters and their respective VMAF values. The high-level architecture of the system is shown in Figure 4.1. The design and development are broadly divided into two phases, the Data Gathering and Model Building.

4.1 Data Gathering phase

4.1.1 Video Preparation

Video Dataset

One of the primary motivation for this work is the increasing popularity of HD videos[3]. There are very few previous works where High-Definition(HD) videos have been used in a network simulation environment for QoE prediction. Hence, only HD 1080p video are considered for this work. Also, the video used for experimentation has to be from different categories. The impact of same levels of packet loss on a steady frame video(newsreader) is not the same as in an animation video. This could also be validated from the work done in [49], where different types of video yield different QoE

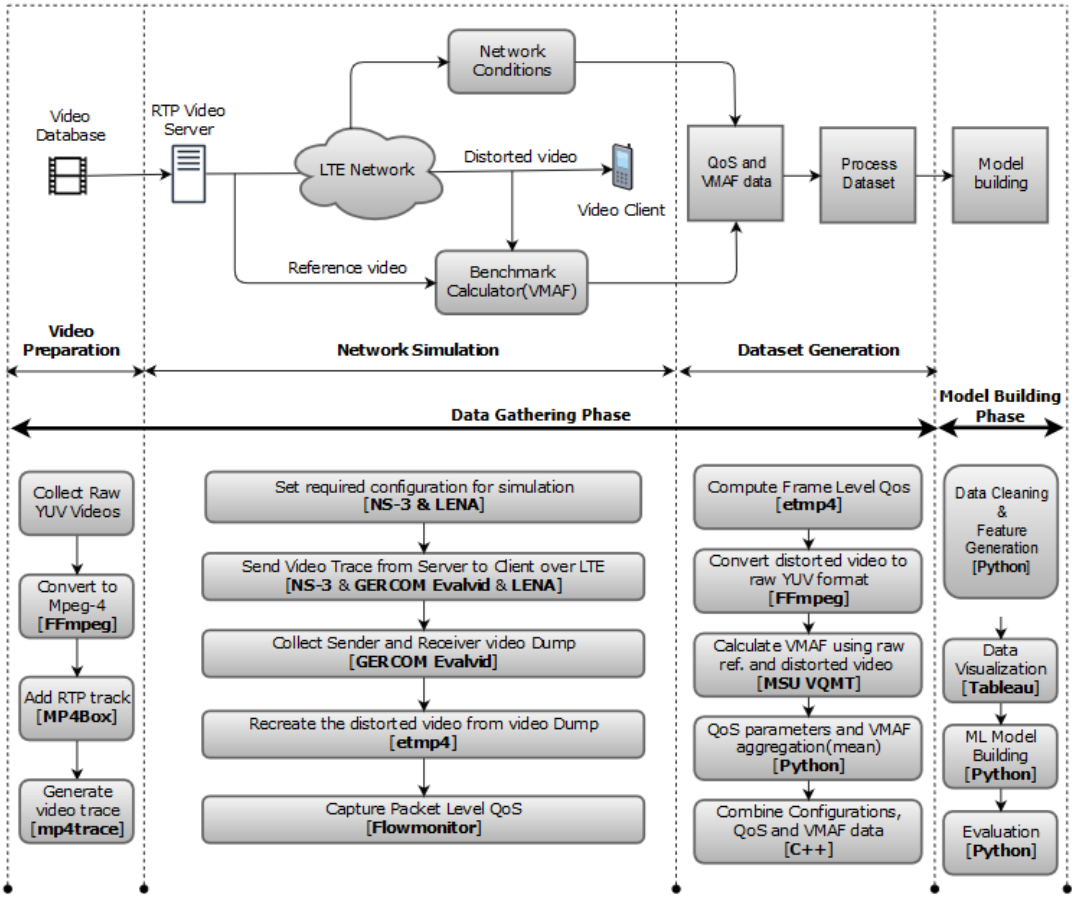

Figure 4.1: High Level Architecture for Data Gathering and Model Building Phases

under same loss conditions. For this research work public HD videos from Netflix Public Data set[65] and public TUM video dataset[66] are used. Table 4.1 describes the videos in the dataset, and all the videos are downloaded in uncompressed raw YUV format.

Video Conversion

For HD videos the ideal bitrate settings vary from 4Mbps to 8Mbps[67]. Each of the chosen video files are compressed using FFmpeg into MPEG-4 format with bitrates 512Kbps, 4Mbps, and 8Mbps at 25fps. In addition to the recommended bitrate for HD videos, 512Kbps was chosen to create a lower quality video for experimentation purpose. Now each of these videos files has to be converted into a trace file for simulation. Firstly,

Video Id	Name	Description	Duration
1	FoxBird	Animation	6
2	Seeking	Slow sliding shot of people moving	6
3	BigBuckBunny	Animation	6
4	OldTownCross	Drone shot of City	6
5	CrowdRun	Steady shot of people running	6
6	Shield	Slide and Zoom of shields	10
7	Stockholm	Pan shot of a city	10
8	Tree	Drone shot of a tree	10
9	Parkjoy	Fast sliding shot of people running in park	10
10	Parkrun	Pan shot of a single person walking	10

Table 4.1: Video Dataset

a hint track of MTU 1460 is added to the encoded video file, which describes how to packetize the video frames for the transport with RTP. Then, this video file is converted to a trace file using evalvid etmp4. Figure 4.2 shows the stages of video files conversion.

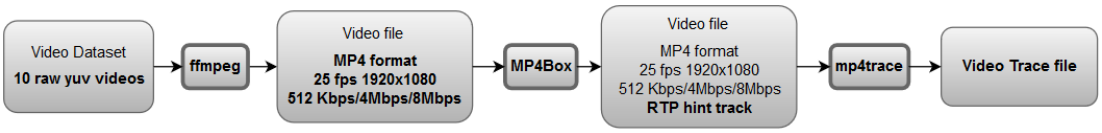

Figure 4.2: Video Trace File Generation

4.1.2 Network Simulation

The network simulation forms the heart of the system(Figure 4.1). The video files prepared as explained in the previous section has to be sent over a network for creation of different distorted videos. LTE is the chosen means for video transmission in this research work. Considering the difficulty in the creation of a real physical LTE network for experimentation a network simulation approach is taken. It is extremely difficult to model a network simulation that behaves exactly similar to a real-world network. NS-3 and Lena LTE module are used for the creation of a simulation environment. The LTE network simulation is divided into three parts: Network Topology, Traffic design, and Configuration variability

Network Topology

The LTE network topology designed is very similar to LTE-EPC architecture(Figure 2.6) discussed as part of LTE LENA module in chapter 1. The topology consists of UE, eNodeB, EPC, and remote hosts. Although the EPC consists of additional components namely MME, SG-W, and P-GW, this research is just focused on the radio link between UE and eNodeB. The remote host, EPC, and eNodeB are connected using a Point-to-point link, and the UE and eNodeB are connected using a radio link. Data can flow in both the uplink and downlink directions.

For the sake of simplicity, only one eNodeB is modeled, and it can be connected to multiple UE. It was identified by manual experimentation, that under the default setting there was no communication between the UE and eNodeB for a distance parameter greater than 50km. So, it was decided to distribute the UE randomly with a minimum distance of 0.5Km and a maximum distance of 50Km as shown in Figure 4.3. Each UE is assigned a particular application, for example, one of the UE will be streaming the video file that is sent from the Video Server. Also, each UE is attached with a default bearer and dedicated bearer depending on the application data they request as shown in Figure 4.4.

Network Traffic Design

The Internet consists of a mixture of different traffics. To build and evaluate a QoE system that can be deployed in real-world, it is necessary to recreate a similar traffic model in the simulation environment. In [68], a traffic generator that matches the statistical properties of real-life IP networks for NS-3 is designed and evaluated. Though the implementation of the traffic generator was made available, it was not compatible with the recent NS-3 version that was used in this research work. Alternatively, a manual approach was taken to overcome the shortcoming of unavailability of a realistic internet traffic generator. The internet mostly consists of either TCP or UDP transport layer protocol, and NS-3's off-the-shelf libraries can either generate TCP or UDP traffic. Hence, the network simulation environment was designed to consists of 50% TCP traffic and 50% UDP traffic.NS-3 provides the below list of applications[6] to be modeled inside the simulation environment:

- **BulkSendApplication**: This application can send data as fast as possible up

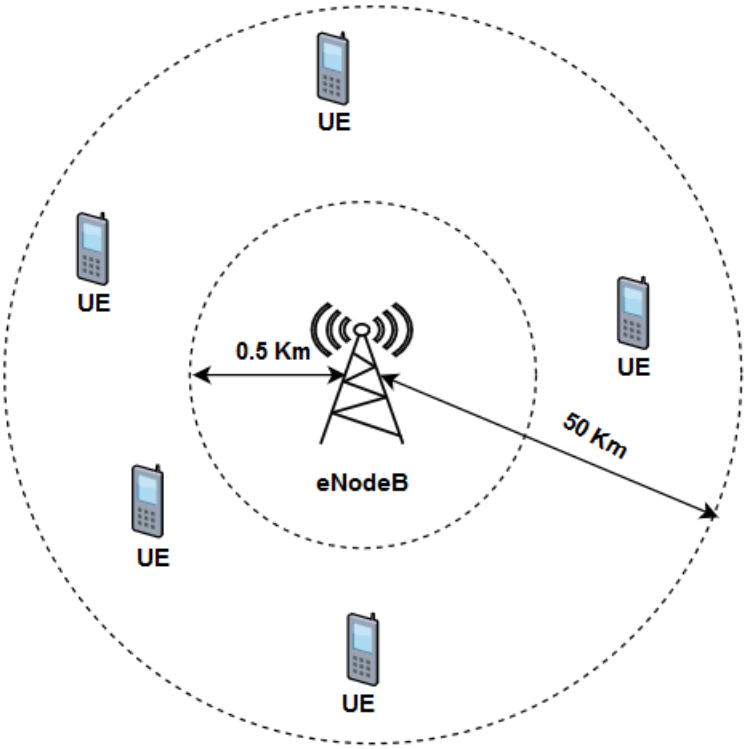

Figure 4.3: Topology of UE and eNodeB

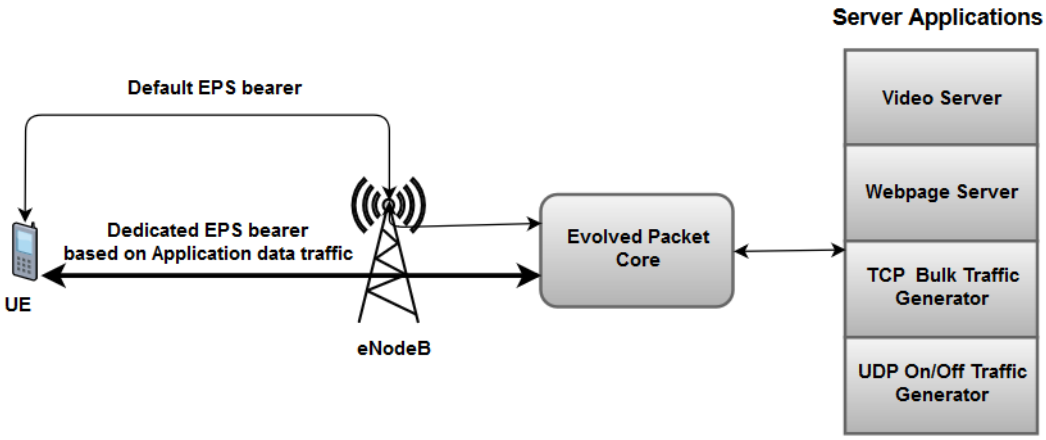

Figure 4.4: Topology of UE, eNodeB, EPC, and Server Applications

to a Maximum byte limit or until it is stopped. It can send both UDP and TCP traffic.

- **OnOffApplication**: This can send traffic in an ON and OFF pattern. For a random time period, the application sends data and for a random time period, no traffic is generated. This supports both TCP and UDP traffic.
- **ThreeGppHttpClientServer**: This is used to create a traffic similar to web-pages. The webpage is sent to the client and then there is waiting period called reading time, before sending the next set of data packets. This is based on HTTP over TCP.
- **UdpTraceClient**: This application can send UDP packets based on a trace file of an MPEG4 stream.
- **PacketSink**: It is used to receive and consume traffic generated, both TCP and UDP. Ideally used as a Client applications for BulkSendApplication and OnOffApplication.

Initially, it was assumed that NS-3's UdpTraceClient can be used for sending the video trace files, but it was found that it was impossible to recreate the distorted video from the received dump, which is essential for calculating the Benchmark QoE. This problem was overcome by using GERCOM Evalvid for NS-3 package. GERCOM Evalvid has a Server application which can be configured to act as Video Server and a Client application which can be installed in the UE to stream the video. GERCOM Evalvid generates a sender video dump and a receiver video dump file(Table 2.3) during the simulation and the trace file is used to recreate the distorted video file. As mentioned earlier the simulation contains equal proportions of TCP and UDP traffics. Each UE installed with one application and a dedicated EPS bearer. Table 4.2 shows the different applications used in the simulation, the type of traffic they generate, the dedicated bearer assigned(QCI), and the Number of UE with the mentioned applications are installed.

LTE Configuration

The goal of the simulation is to send video files under different networking conditions. Since this research is focused primarily on the Radio Link between eNodeB and UE, only the associated parameters are varied. The parameters have to be suitably selected

NS-3 Application	**Traffic Type**	**QCI**	**Installation (No. of UE)**	**Installation (No. of UE) when totalNoUe=8**
EvalvidClientServer	UDP	4	1	1
ThreeGppHttpClient	TCP	9	$totalNoUe-$ $(totalNoUe-1)/2-$ $(totalNoUe-1)-((totalNoUe-1)/2)$	2
OnOffApplication	UDP	2	$(totalNoUe-1)/2$	3
BulkSendApplication	TCP	8	$((totalNoUe-1)-((totalNoUe-1)/2))/2$	2

Table 4.2: UE application QCI and Installation

to create distributed variation in network QoS Parameters such as loss, delay, and jitter. Below is the list of parameters configured:

- **Number of UE**: This is the total number of UE attached to the base station(eNodeB). Increasing the number of UE will result in increased load on the network causing congestion. After multiple test iteration the values 4,8,12, and 16 were chosen.

- **Transmission Mode**: LENA LTE allows to use 7 different transmission model. Each transmission mode represents a type of RF control systems, as discussed in section 2.1.3. SISO is an old technique compared to MIMO. Test simulations showed a significant difference in throughput with MIMO outperforming.

- **Resource Blocks(RB)**: A resource block is defined as the smallest unit of resource that can be allocated to a user. Number of RBs represents the transmission Bandwidth. Higher the resource blocks allocated higher will be the throughput. RBs can take values 6(1.4MHz), 15(3MHz), 25(5Mhz), 50(10Mhz), 75(15Mhz), 100(20Mhz). 6, 25, and 100 are chosen for this work. The same value is set for both uplink and downlink traffic.

- **Max.Size of the Transmission Buffer**: It is the maximum buffer size assigned for the transmission. During test simulation, the default value of 10Kb assigned for buffer size was not sufficient to stream a complete HD video due to heavy packet loss. Low values of buffer size caused heavy packet loss but timely delivery. A high value of buffer size results in very less packet loss, but a larger delay. After some test experimentation 100Kb, 512Kb, and 5Mb were chosen.

Configuration Parameters	Values
Number of UE	Variable: 4, 8, 12, 16
Transmission Mode	Variable: SISO, MIMO Tx Diversity
Resource Blocks	Variable: 6, 25, 100
Max. Size of the Transmission Buffer	Variable : 100Kb, 512Kb, 5Mb
Number of eNodeBs	1
UE Placements	Randomly between 500m - 30Km
EARFCN Downlink	100
EARFCN Downlink	18100
UE Transmission power	10 dBm
eNodeB Transmission power	30 dBm

Table 4.3: Configuration Parameters

- **E-UTRA Absolute Radio Frequency Channel Number(EARFCN)**: EARFCN is channel number for the frequency, different numbers correspond to different frequency value. The default value of 100(2120MHz) for downlink and 18100(1930) uplink was configured

- **Transmission Power**: Transmission power is nothing but the strength of the signal the UE or eNodeB during transmitting. This value is assigned the default.

Work done by Tarik et al[64] is very similar to the objective of this research work, but there are few limitations here namely lack of sufficient data points, no variation in transmission buffer size, constant placement of UE, use of just one ML algorithm, additionally the ML algorithm were not sufficiently evaluated using robust evaluation metrics. This research work is designed to overcome these limitations by generating a reasonably larger dataset by varying the LTE parameters, building multiple ML models, and performing through evaluation. Now, each video file is encoded into 3 different bitrates and each of them is simulated using the 72 combinations of parameters(table 4.3) resulting in a total of 216 simulations per video.

4.1.3 Dataset Generation

For each simulation, the Network QoS parameter and Benchmark VMAF has to be collected as shown in Figure 4.1.

Network QoS Parameters

Network QoS parameters are collected at the packet level and frame level granularity. The packet level QoS parameters are obtained using the Flowmonitor module(section 2.1.4). The flowmonitor is installed in the UE that is streaming the video, to capture the total number of Packets, Total Jitter Sum, Total Delay Sum, Packets lost, and throughput(Kbps). The frame level QoS parameters are calculated using Evalvid etmp4(Section 2.2.3). It captures Total Number of Frames, Sender Inter Frame delay, Receiver Inter Frame delay, Cumulative Jitter, Loss ratio of I-frame, loss ratio of P-frame, and loss ratio of B-Frame.

Benchmark VMAF

The impaired video obtained after the simulation is compared with the original video file to compute the VMAF value. MSU Quality Measurement Tool discussed in Section 2.3.4 is employed in this research to calculate the VMAF of each distorted video.

An initial attempt was to retain the QoS for every frame rather than computing the average, such that a video file with 150 frame will generate 150 rows of data. Then for each frame, the VMAF value can be computed. This approach was not suitable because under some harsh networking condition the video got heavily distorted causing many frame drops. Lack of frames resulted in no corresponding VMAF value, and there is no suitable solution to identify the lost frame number to map a VMAF value of 0. Hence, this approach was dropped and an aggregation approach was implemented, such that each video simulation will generate mean QoS parameters and mean VMAF. Simulation of 10 video files encoded into 3 different bitrates under 72 different networking condition will result in a total of 2160 data points.

4.1.4 Implementation

The Design and implementation of code for Data collection is divided into three modules as show in the Figure 4.5.

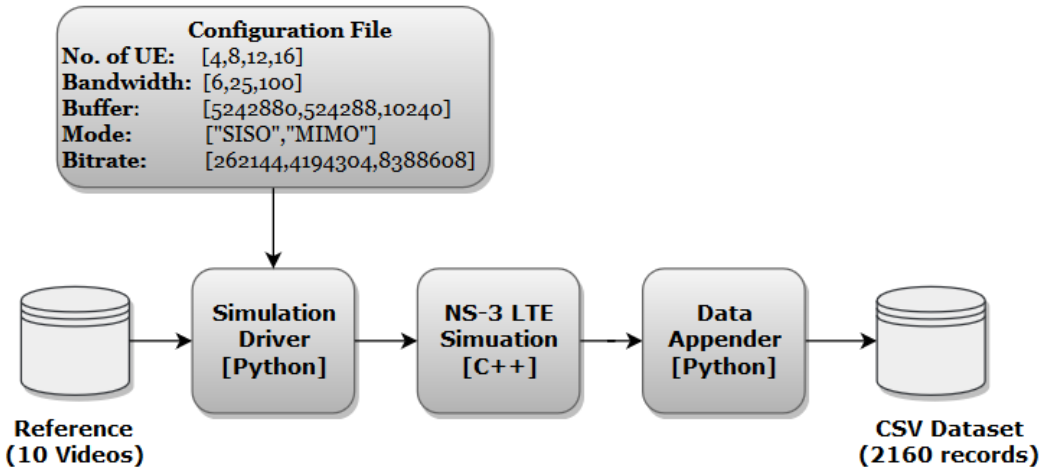

Figure 4.5: Data Collection Phase - Implementation

NS-3 LTE Simulation Module

NS-3 LTE Simulation module is written in C++ and is run using the command line. This module's main design consideration is extensibility and automation. As explained in the previous sections that it was decided to conduct a total of 2160 different simulations. Each simulation requires a distinctive set of operations involving a number of tools and frameworks, as shown in Figure 4.1. Manual execution of these steps is laborious, hence the NS-3 LTE simulation module was created as a generic piece of code which can be executed externally by passing run-time parameters. This module was designed keeping options for extending the research work. In this work, only a few network configuration parameters are decided to be varied and capture only the Network layer QoS parameters, but some previous works have explored other QoS parameters such as video bitrate, Framerate, and resolution. The NS-3 LTE Simulation module is implemented with a wide variety of run-time parameter. The Figure 4.6 shows the list of parameters that can be configured in the current implementation and their default values. This module requires the network configuration and video workspace(directory) containing raw YUV video, MPEG-4 video file and its corresponding trace file. This module outputs a CSV data file containing a single row of the columns. The data dictionary is shown in Table 1(Appendix).

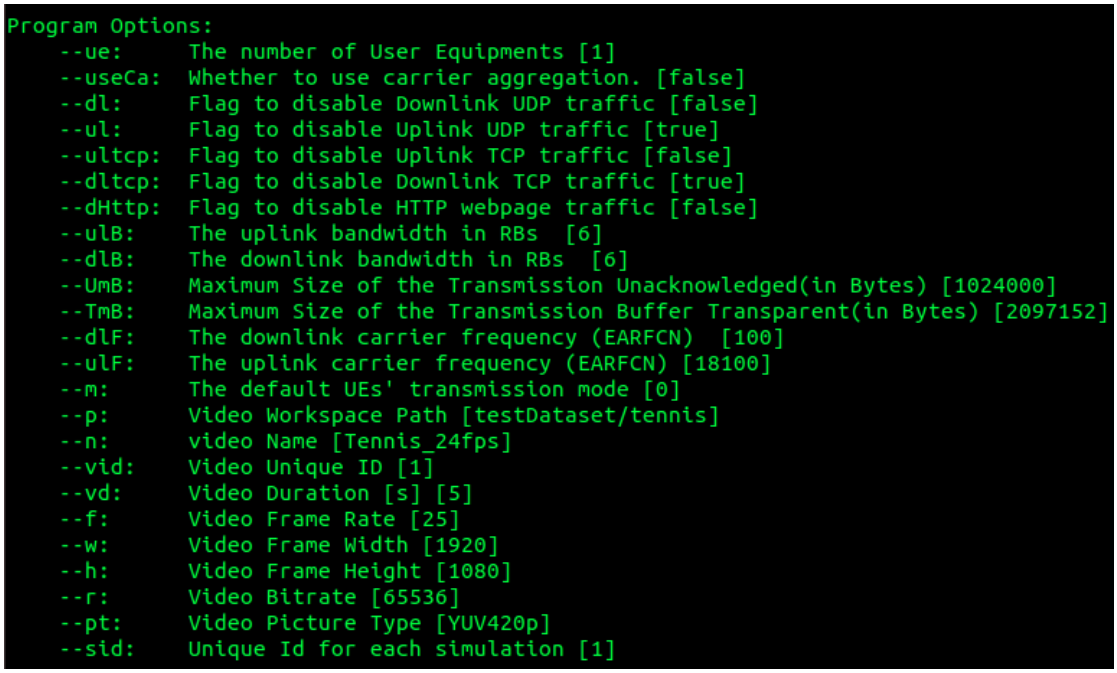

Figure 4.6: NS-3 LTE Simulation Module Run time Parameters

Simulation Driver Module

Simulation Driver Module is written in Python. This module is responsible for automating the simulation process for data generation. A configuration file consisting of the different networking parameter and bitrates along with the video dataset is the input to this module, as shown in figure 4.5. Apart from the data mentioned in the configuration files this module generates values videoId and SimulationId based on counter variable. For each video sequence, this module creates a workspace(directory) and converts video into three bitrates. These three video files are then converted into a trace file as explained earlier. The NS-3 LTE simulation module is executed for each of this newly encoded video for the 72 different networking condition specified in the configuration file. This module iterates through a combination all the properties in the configuration file against all the video files resulting in total of 2160 simulations. At the end of all the simulations, the Data Appender Module is executed.

Data Appender Module

This is a simple module written in python, which is responsible for iterating through all the dataset generated by the simulation to create a single CSV file. As mentioned earlier each simulation will create a CSV file containing a single row, but for further processing, it will be easier to have a single file containing all the data points. This module will result in the final dataset in CSV format which form the base for the Model Building Phase.

4.2 Model Building Phase

The Model Building is the final phase of this research work. As shown in Figure 4.1 this phase is subdivided into four stages.

4.2.1 Data Cleaning and Feature Generation

The dataset generated from the Data Collection Phase has to be prepared for model building. Unclean data will result in incorrect results. In this stage, the generated data is carefully observed for the commonly occurring data issues discussed in [69]. Below are the steps taken in this research for ensuring data integrity of the collected data.

- **Fix row and Columns**: Data engineering process sometimes results in blank unnecessary extra rows or misaligned columns. Misaligned columns might occur when a column value is made of the delimited value, in our case comma. During the data cleaning process, the dataset did not have any misaligned columns, but extra blank rows were found. since this issue was found during the early stage of simulation a bug fix was applied in the simulation code to avoid this scenario.
- **Missing Values**: Missing values are a very common issue in the data collection process. Some machine learning model such as Linear regression do not support missing values for model building. When computing the mean VMAF and mean QoS parameters, an initial approach used was a built-in function, but it resulted in null values for scenarios when there was complete frame loss. An alternative approach of dividing the total number of Frame or Packet was used to overcome this issue.

- **Standardize Numbers**: NS-3 outputs memory in bytes and for easy understanding they were converted to Kilobytes or Megabytes based on the variable.
- **Fix Invalid Values**: Columns jitterSum and delaySum(appendix 1), the output from Flowmonitor was concatenated with its unit "ns". This will not allow numeric computation on this column, hence the non-numeric part was removed. Additionally, all the rows and columns imported into the python environment validated for the datatype consistency.
- **Filter Data**: In order to build a parsimonious model, it is important to remove unnecessary columns. Columns that had no variation such as BframeLossPerc, width, height, and frameRate was removed from the dataset. Id columns and Name columns were dropped because they were just for identification and serves no purpose for further processes. Finally, in the simulation data there were some simulation scenarios caused the received video distorted to an extent of no motion, that is the received number of frames were less than 25. There were about 230 such scenarios and these records were removed from the final dataset for model building.

As part of feature generation, variables JitterSum and DelaySum were converted to their respective mean values.Packets loss was converted to Packet loss ratio. At the end of this stage, only the columns necessary for data visualization was retained.

4.2.2 Data Visualization

Data visualization will give more insight into the dataset. The underlying assumption of data exploration is that, the more one knows about the data, the more effectively the data can be put to use[70].In this research Tableau[71], a powerful GUI based desktop visualization software is used to create various data visualization. Additionally, matplotlib package[72], a program based visualization tool, is also used.In this research, two types of data visualization are performed namely univariate and bivariate analysis.

Univariate Analysis

Data analysis performed on a single variable is called as univariate analysis. Histograms are used to find the distribution of continuous variables such as Frame Loss ratio,

VMAF, and Packet loss ratio, while Frequency bar plots are used for dimensional variables such as bandwidth, transmission mode, and number of User equipment.

Bivariate analysis

Data analysis performed on two variable is called as bivariate analysis. Bar plots are employed when there is one dimension variable and one categorical variable. For example, Packet loss ratio with VMAF value. Also, Pearson correlation plots are used to find the relationship between two continuous variables.

4.2.3 Machine Learning

Six White-box regression algorithms specifically Regression Trees, Random Forrest regression, gradient boosting regression, linear regression, lasso regression, and ridge regression were picked for implementation. Variables that are unnecessary for model building were removed from the dataset post data visualization stage. The final dataset contains Network QoS: throughput, Overall Frame loss ratio, P-Frame loss ratio, I-Frame loss ratio, frame end-to-end delay, Sender Inter Frame Delay, Receiver Inter Frame Delay, Frame Cumulative Jitter, Packet loss ratio, Packet jitter, and Packet delay and QoE: VMAF.

Regression Evaluation Metric

In Machine Learning, the model evaluation metric is used to find the predicting power of the model. There are a variety of evaluation metrics and in this research, for model building and evaluation these two metrics are used:

- **Mean Squared Error(MSE)**: MSE is the mean square of errors in prediction.Error(e) is the difference between the predicted and actual value, as shown in figure 4.7. Equation 4.1 is the formula to compute MSE. MSE is inversely proportional to the model performance. A good model should have a smaller MSE value.

$$MSE = 1/n \sum_{i=1}^{n} (y_i - \hat{y})^2 \tag{4.1}$$

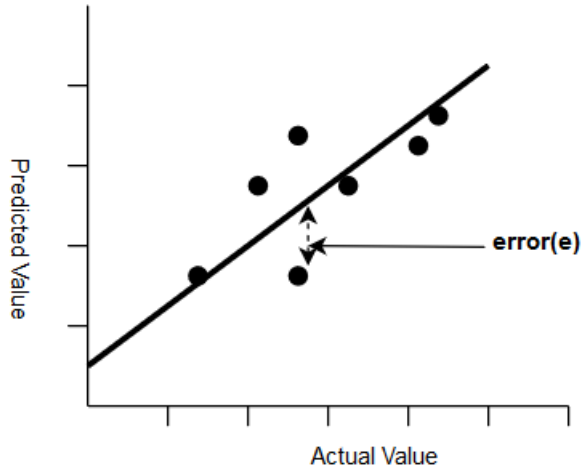

Figure 4.7: Prediction error in regression

- **Coefficient Of Determination**: Coefficient of Determination is also know as R^2. It measures the variance in the output variable(y) that can be explained by the model. The formula for calculating R^2 is given below. R^2 varies from 0 to 100. Higher R^2 value implies high model predicting power. R^2 is approximately equal to the the square of correlation. R^2 is a robust metric and very much suitable for this thesis work.

$$R^2 = 1 - \sum(y - \hat{y})^2 / \sum(y - \bar{y})^2 \tag{4.2}$$

Model Evaluation

The goal of a supervised machine learning models is to learn from a know labeled data(training set) and predict the label values for future data. During model building the future data will not be available, so to overcome this problem the collected dataset is split into training and test sets. The test set will act as the proxy for future/unknown data. Now, the training dataset will be used for model training, but the model evaluation on the just training set will yield bias results. There are two popular approaches to overcome this problem :

- **Hold-out**: In hold-out technique, the training dataset is further split into the training set and validation set. Now the ML model is build on the training set

and evaluated on the validation set. Apart from evaluation, the validation set is used for tuning the hyper-parameters.

- **Cross-validation**: The main drawback of the hold-out method is that a good proportion of data(validation set) cannot be used in training the model. A more powerful approach is cross-validation or K-fold cross-validation. In K- fold cross-validation, the model is trained on one group and scored on the left out group and this process is repeated until each group is used as a test set. For example, in 10-fold cross-validation, the dataset will be split into 10 different groups and the model is built 9 separate times and test on the group that was not part of training.10-fold cross-validation is shown in the figure 4.8. . K-fold cross-validation can be configured to output the mean evaluation metric with which confidence intervals can be computed.

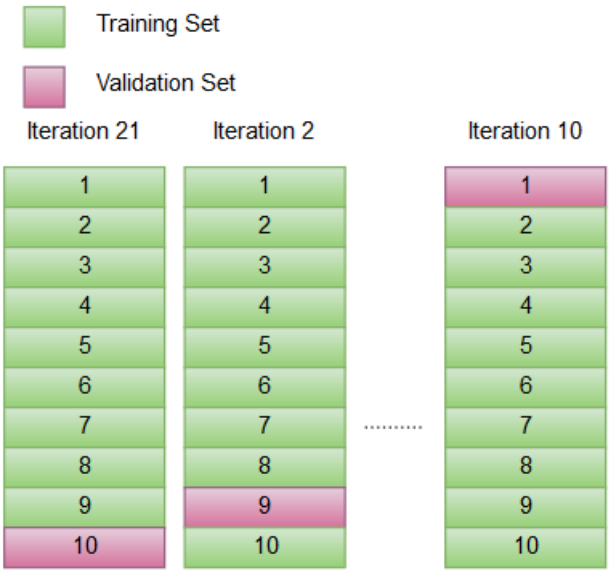

Figure 4.8: 10-Fold Cross validation

Grid Search with Cross Validation

Machine Learning algorithm can have a set of parameters called hyperparameters that are not directly optimized by the algorithm. For example for ridge regression

λ(equation 2.3) is the hyper-parameter. In ML choosing these hyperparameters are done by trial and error method. Sklearn[73] gives an automated option called GridSearchCV, where a grid of possible values for the hyperparameters are set. Then GridSearchCV will build models using cross-validation by iterating through all the possible combination of the hyperparameter. GridSearchCV inputs an evaluation metric based on which the best set of hyper-parameters for the model is given as a result. In this research, this technique is used for tuning the hyperparameters.

Implementation

Python sklearn[73] libraries are used for model building in this work. As decided earlier six algorithms were trained and evaluated. Except for linear regression, all other models have hyper-parameters. Below are the steps followed in model building

1. All the input features are standardized by removing the mean and scaling to unit variance, using sklearn[73] StandardScaler. This aids in quicker optimization of the algorithm.

2. Set the seed value for achieving repeatability while model building. If seed value is not set then the model performance will change for every run even under the same conditions.

3. The dataset is divided into training and test sets in 80:20 ratio.

4. ML model is built on the training set. Depending on the type of model a grid of hyper-parameters is selected based on some initial test trials. Table 4.4 shows the list of variables tuned and the final parameter values that yielded maximum performance.

5. For each combination of the hyperparameter in the grid, 10-fold cross-validation is performed resulting in mean and standard deviation of R^2 of both training and validation set.

6. For each ML algorithm, the model that resulted in highest R^2 value for cross-validation is chosen as the final model.

7. This final model is now tested against the test set that was initially kept aside.

8. Finally, feature importance for tree based algorithms and coefficients(θ) for linear models are analyzed.

4.3 Challenges

These are the challenges faced during the designing and implementation of the research work.

- NS-3 is an open-source software developed and maintained by the research community. The code base is huge and the creation of simulation requires putting together pieces of abstract code. This made code debugging extremely difficult and there are very few resources on the internet. It took significant time to build a consistent code for LTE and video simulation.

- During simulation tests, it was found that HD video did not reach the designation completely even when the bandwidth was kept the maximum and UE placed near the base station. For this issue the packet traces had to be analyzed at the various stage of data flow from server to UE. After rigorous effort, the issue was found to be with the setting of RLC buffer.

- While debugging the above issue, a bug in GERCOM Evalvid package was identified. This bug causes the last packet to be dropped in scenarios when frame size equals the multiple of the MTU setting. This bug was fixed by appropriately modifying the code logic.

- Getting the right combination of network parameter setting(table 4.3) was the hardest as it required multiple batch simulation runs to identify the exact combination that results in a decent distribution of the target VMAF values.

- Initially each simulation took around 6 minutes to create a single data point. At this rate to generate 2160 data points would have required 9 days of continuous run. After some research, it was identified that NS3 provides an optimized configuration using which an optimized version of the build can be created. The drawback of this build is it doesn't generate lod and cannot be used for debugging. Once the code was tested thoroughly the optimized version of NS-3 was

Algorithm	Hyper-Parameter	Best Parameter
Linear Regression	-	-
Lasso Regression	λ = [start=0.001; end=1; step= 0.005]	λ=0.066
Ridge Regression	λ = [start=1; end=10; step= 0.02]	λ=8.4
Random Forrest	criterion = mse, mae max_features = sqrt, log2 min_samples_leaf = 1, 3, 5 min_samples_split = 2, 8, 12,16,20 n_estimators = 50, 100, 200, 400, 800	criterion = mse max_features = log2 min_samples_leaf = 1 min_samples_split = 2 n_estimators = 400
Regression Tree	criterion = mse, mae max_features = auto, sqrt, log2 min_samples_leaf = 1, 3, 5 min_samples_split = 2, 8, 12,16,20 max_depth = 2, 4, 6, 8, 12	criterion= mse max_depth = 12, max_features = auto min_samples_leaf = 5 min_samples_split = 2
Gradient Boosting	criterion = mse, mae max_features = sqrt, log2 min_samples_leaf = 1, 3, 5,8 min_samples_split = 2, 8, 12,16 n_estimators = 100, 200, 400, 800 max_depth = 2, 4, 6, 8, 12 learning_rate = 0.01, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2	criterion = mse, mae max_features = log2 min_samples_leaf = 3 min_samples_split = 2 n_estimators = 100 max_depth = 6 learning_rate = 0.1

Table 4.4: Parameter Tunning

used which resulted in the same simulation to run under 2 minutes. It took 2.5 days for the simulation to complete.

- Lastly, putting all the different pieces of tools and technology into a single simulation environment was the biggest challenge of all.

4.4 Summary

In this chapter, the design and implementation were discussed which mainly consists of the Data gathering phase and Model building phase. In the data gathering phase, 10 different videos encoded into 3 bitrates was run under 72 different networking conditions resulting in a dataset containing the metadata, QoS parameters, and QoE metric. In the Model building phase, the generated data is cleaned, analyzed, and six ML algorithms were tuned and built using a grid search cross-validation technique. In the next chapter, the results are discussed.

Chapter 5

Results and Evaluations

This chapter presents the results and observations obtained from the Model Building Phase discussed in the previous chapter.

5.1 Data Exploration

Understanding the data is important before getting into the machine learning part. This section covers some of the observations from visualizing the input data. Figure 5.1 presents a histogram of collected VMAF values. It is interesting to note that most of the values are near maximum and the reason for it that VMAF is designed to output higher values for HD video and lower values for lower resolution video[46]. This is the main reason for the left-skewed data distribution. For building good performing ML models the data in the training set should be from the same distribution of the population. The key understanding here is that the model build on this dataset will not perform well on a video of lower resolution. Figure 5.2 shows the relationship between the network parameters that were modified in the simulation with the VMAF. The intuition that increasing the load(No.of UE) on the network will reduce the performance, and increasing the bandwidth allocations will increase the performance can be confirmed from the bar plots. Even though in the background research, it was explained that MIMO improves the throughput, the graphs do imply it. One possible reason is during data cleaning some video simulations that didn't have sufficient frames were removed from the dataset, there is a good chance that all those records were simulated under

SISO. Buffer size doesn't seem to have any direct relationship which VMAF. Previous work[54, 55, 56] had identified that Packet loss is not linearly related to the video quality metric and it can be confirmed from the Figure 5.3. Packet level QoS metrics do not have a linear relationship, but the frame level QoS, especially Frame Jitter and frame loss ratio are well correlated with the VMAF values. It is likely for frame level QoS to be good predictors in our ML models.

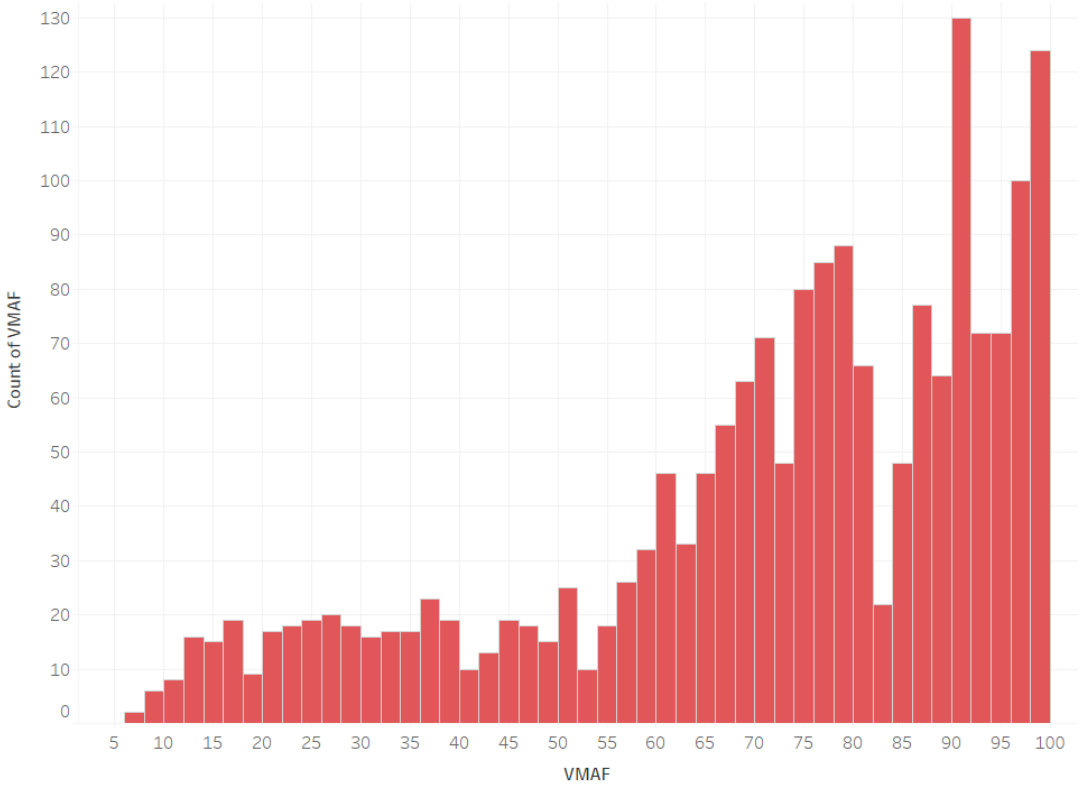

Figure 5.1: VMAF - Histogram

5.2 Model Evaluation Results

Six Machine learning algorithms were trained using a grid search K-fold Cross-validation technique on the training set. The best set of hyper-parameter for each model is identified which were discussed in the previous chapter. R^2 is the chosen evaluation metric for assessing the performance of the algorithms. For each model, Table 5.1 presents the

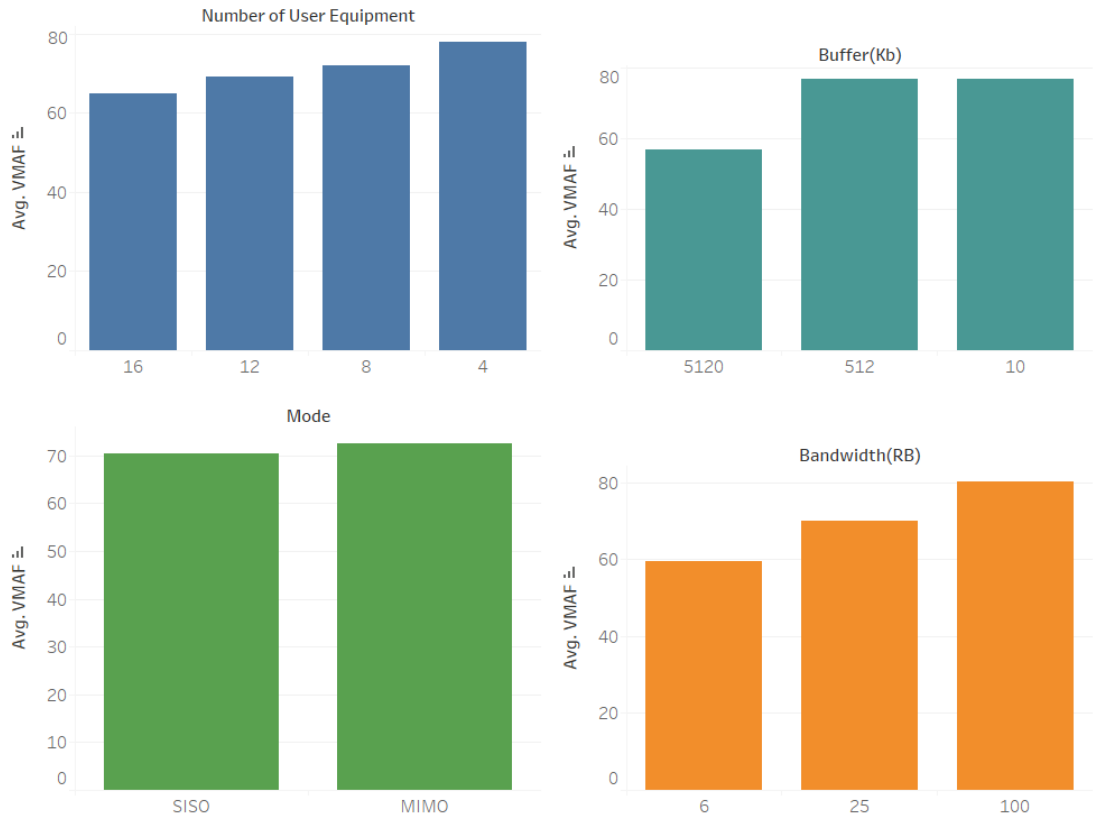

Figure 5.2: Network Parameters Vs Average VMAF

validation set R^2 expressed in 95% confidence interval and Test Set R^2. The confidence interval is an interval that is likely to have the true value of the estimation(R^2) of the population. In all the models the CI is very narrow which is a good sign that the model is very precise. This can be further confirmed from comparing the interval with the Test set R^2 value. It could be observed that the validation and test set accuracy are very similar, this suggests that the models are not overfitting and have a good bias-variance trade-off. The tree-based algorithms perform better than the linear model algorithms. Random Forrest and gradient boosting are the top performers with an R^2 value of 0.956 and 0.951, however Compared to linear models they take more time to be trained. Linear models coefficients can be useful in understanding the relationship with the QoE.

Figure 5.3: Scatter Plot of QoS Vs VMAF

5.3 Feature Importance

One of the main reason for choosing the white-box algorithm's is to interpret the model by understanding the predictive power of the QoS parameters. Tree based algorithms outputs feature importance which determines how influential the input parameters are while splitting the feature space. Figure 5.4 shows the feature importance obtained from the Regression trees, Random Forrest, and Gradient boosting algorithms. It could be observed that Frame level QoS parameter are very influential, specifically Frame loss ratio. From tree based models we could conclude that Frame level QoS parameter strongly influences the predictive power of these models. Figure 5.5 shows the coefficient of Linear models(θ). Linear regression model's coefficients are particularly interesting because the Total Frame loss ratio and P-Frame loss ratio are pulling the value of predictions on both sides. The reason for this could probably due to overfitting and multicollinearity. Ridge and Lasso regression models have overcome this problem by the regularization. Table 5.2 shows the numeric values of linear model coefficients

Algorithm	**Validation Set R^2(95% CI)**	**Test Set R^2**
Linear Regression	[0.8527, 0.8633]	0.866
Lasso Regression	[0.8535,0.8635]	0.885
Ridge Regression	[0.8534,0.8636]	0.886
Random Forrest	[0.9471,0.9523]	0.956
Regression Tree	[0.9291,0.9342]	0.943
Gradient Boosting	[0.9588,0.9652]	0.951

Table 5.1: Machine Learning algorithm evaluation

Features	Linear Regression	Lasso Regression	Ridge Regression	Random Forrest	Regression Tree	Gradient Boosting
I-Frame Loss Ratio	-22.3827	-7.71295	-7.2859	0.18441411	0.02303032	0.12847977
P-Frame Loss Ratio	-166.855	-13.6253	-7.22366	0.27378945	0.75195008	0.0949947
Total Frame Loss Ratio	166.5241	0	-7.21245	0.28183588	0.11339894	0.09217117
Frame End-To-End Delay	-2.13377	-1.64732	-1.89067	0.01832017	0.00230968	0.04002756
Frame Sender Inter Delay	-2.39492	-2.23746	-2.33473	0.04271061	0.07623482	0.34554995
Frame Receiver Inter Delay	2.590243	2.810569	2.737919	0.04703855	0.01483257	0.11361849
Frame Cummilative Jitter	1.340536	0	0.774171	0.09446977	0.00268994	0.04942373
Packet Jitter	-0.29608	-0.16457	-0.28698	0.01493624	0.00392697	0.04912157
Packet Delay	-0.13483	0	-0.20558	0.01724347	0.00378587	0.01670737
Packet Loss Ratio	-0.82404	-0.76571	-0.90748	0.02524174	0.00784083	0.06990569

Table 5.2: Feature Importance and Coefficients of ML model

and tree model importance. Lasso regression model has eliminated some of the features by assigning 0 for the coefficients. In conclusion, Frame level QoS parameter have higher predictive power when compared to packet level QoS , this can be seen in both Linear models and tree model.

5.4 Summary

The machine learning models built are able to predict the QoE effectively using the QoS parameters. The models are precise and are well fit which can be understood

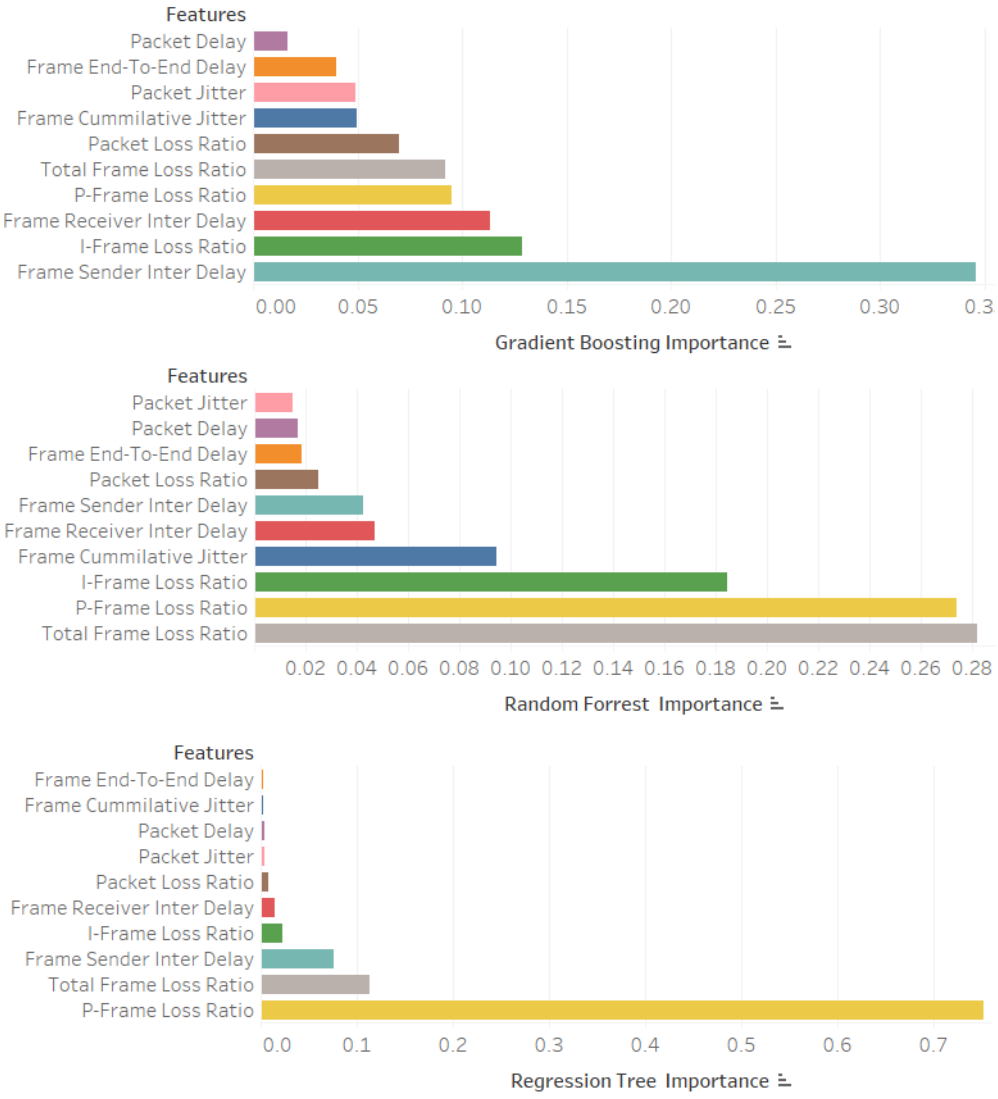

Figure 5.4: Feature importance of Tree based algorithms

by comparing the validation set and test set accuracy. Compared to the Packet level QoS parameters Frame level QoS parameters have higher predicting power. Tree based algorithms out perform the linear based models.

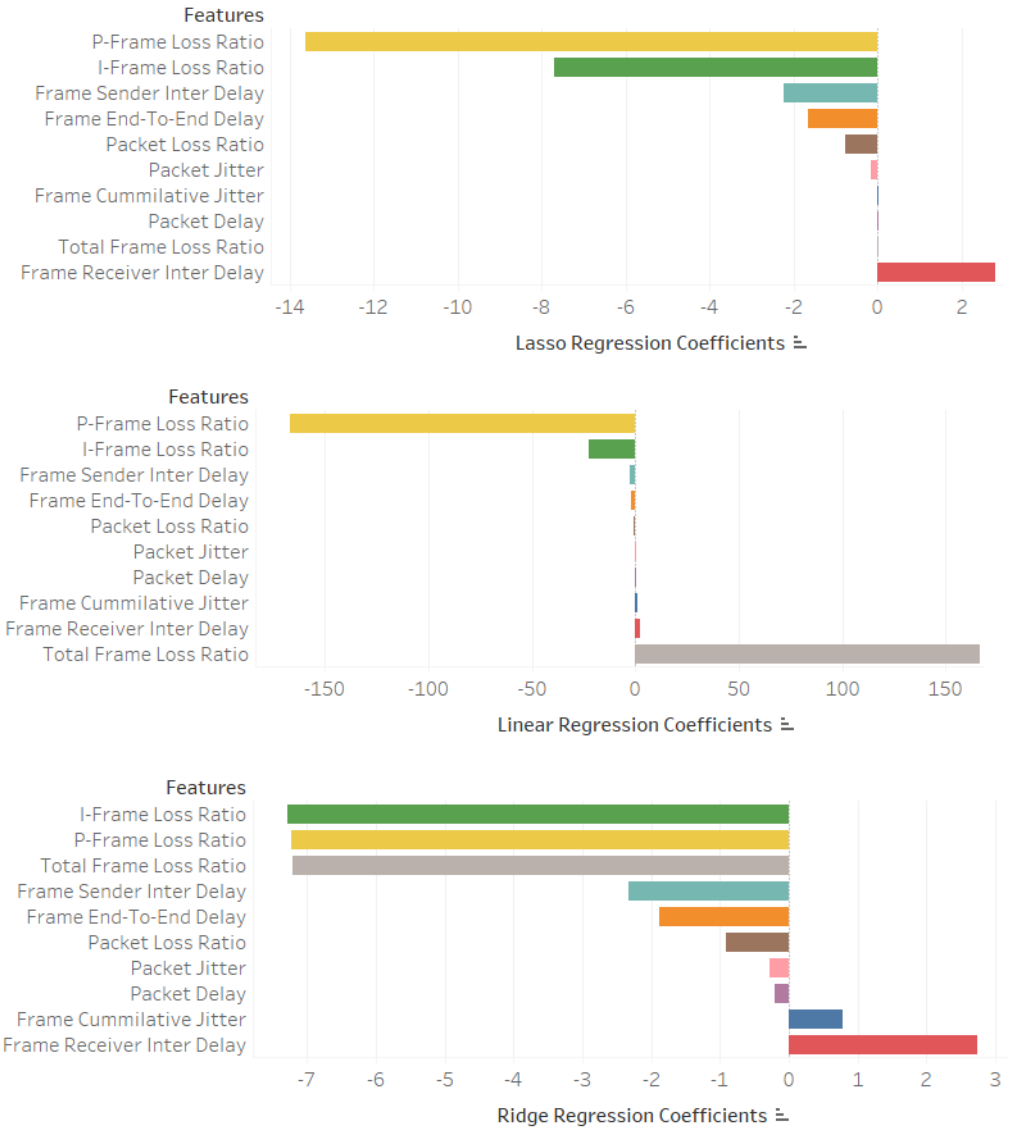

Figure 5.5: Coefficents of Linear models

Chapter 6

Conclusion

In this dissertation, it is identified that machine learning models can be used to predict the video QoE objectively using the network QoS with reasonably good accuracy. Frame level QoS parameter has higher predictive power when compared to packet level QoS. The models trained do not require any reference video signal, hence can be used for real-time prediction. They are lightweight and require low computation power for prediction, however, they require sufficient computation power for training the model. Lastly, this research concludes that Tree based algorithms perform well compared to linear regression models.

6.1 Future Work

There are few drawbacks in the current methodology used namely the benchmark metric chosen is another objective metric. The objective metric might not capture all the Influence Factors. If time permits the distorted video has to be assessed for subjective scores. Building a model on top subjective scores will give realistic results. Alternatively, multiple objective tests could have been taken rather than just VMAF which was considered in this work. Even though some effort was put in simulating realistic traffic into the simulation, it cannot match the real-world traffic. The testbed can be created to capture much realistic video transmission losses. Some of the variable in the dataset were not varied for example the video resolution and frame rate. In the future, these QoS parameters could be varied and a hybrid model could be built.

Specifically, network QoS parameters do not use the information in the payload. This research work focused on white-box algorithms, and there are a wide variety of complex algorithms such as deep learning and Reinforcement learning which could be explored further to build better models.

Bibliography

[1] T. ETSI, "123 203 v10. 5.0 (jan. 2012)digital cellular telecommunications system (phase 2+)," *Universal Mobile Telecommunications System (UMTS)*.

[2] A. Takahashi, D. Hands, and V. Barriac, "Standardization activities in the itu for a qoe assessment of iptv," *IEEE Communications Magazine*, vol. 46, no. 2, pp. 78–84, 2008.

[3] V. Cisco, "Cisco visual networking index: Forecast and trends, 2017–2022," *White Paper*, vol. 1, 2018.

[4] S. Nashwan, "Sak-aka: A secure anonymity key of authentication and key agreement protocol for lte network.," *International Arab Journal of Information Technology (IAJIT)*, vol. 14, no. 5, 2017.

[5] S. Sesia, I. Toufik, and M. Baker, *LTE-the UMTS long term evolution: from theory to practice*. John Wiley & Sons, 2011.

[6] NS-3 Developers, "ns-3 manual," 2018.

[7] N. Baldo, "The ns-3 lte module by the lena project," *Center Tecnologic de Telecomunicacions de Catalunya*, 2011.

[8] B. Ferster, "Digital Video 101: Understanding How Digital Video Works ." `https://elearningindustry.com/how-digital-video-works-digital-video-101`, 2019. [Online; accessed 03-August-2019].

[9] Y. Chen, K. Wu, and Q. Zhang, "From qos to qoe: A tutorial on video quality assessment," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 1126–1165, 2014.

[10] M. T. Vega, C. Perra, F. De Turck, and A. Liotta, "A review of predictive quality of experience management in video streaming services," *IEEE Transactions on Broadcasting*, vol. 64, no. 2, pp. 432–445, 2018.

[11] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*, vol. 112. Springer, 2013.

[12] S. Bennett, "The Experience Age has Arrived." `https://www.batimes.com/articles/the-experience-age-has-arrived.html`, 2019. [Online; accessed 03-August-2019].

[13] K. Brunnström, S. A. Beker, K. De Moor, A. Dooms, S. Egger, M.-N. Garcia, T. Hossfeld, S. Jumisko-Pyykkö, C. Keimel, M.-C. Larabi, *et al.*, "Qualinet white paper on definitions of quality of experience," 2013.

[14] U. Reiter, K. Brunnström, K. De Moor, M.-C. Larabi, M. Pereira, A. Pinheiro, J. You, and A. Zgank, "Factors influencing quality of experience," in *Quality of experience*, pp. 55–72, Springer, 2014.

[15] A. Sackl, P. Zwickl, and P. Reichl, "The trouble with choice: An empirical study to investigate the influence of charging strategies and content selection on qoe," in *Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013)*, pp. 298–303, IEEE, 2013.

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

[17] J. Pokhrel, *Intelligent quality of experience (QoE) analysis of network served multimedia and web contents*. PhD thesis, 2014.

[18] M. Nohrborg, "LTE." `https://www.3gpp.org/technologies/keywords-acronyms/98-lte`, 2008. [Online; accessed 03-August-2019].

[19] R. K. Singh and R. Singh, "4g lte cellular technology: Network architecture and mobile standards," *International Journal of Emerging Research in Management &Technology*, vol. 5, no. 12, 2016.

[20] S. M. Bilalb, M. Othmana, *et al.*, "A performance comparison of network simulators for wireless networks," *arXiv preprint arXiv:1307.4129*, 2013.

[21] NS-3 Developers, "ns-3 tutorial," 2018.

[22] G. Carneiro, P. Fortuna, and M. Ricardo, "Flowmonitor: a network monitoring framework for the network simulator 3 (ns-3)," in *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, p. 1, ICST (Institute for Computer Sciences, Social-Informatics and , 2009.

[23] P. Read and M.-P. Meyer, *Restoration of motion picture film*. Elsevier, 2000.

[24] M. Yuen and H. R. Wu, "A survey of hybrid mc/dpcm/dct video coding distortions," *Signal processing*, vol. 70, no. 3, pp. 247–278, 1998.

[25] F. Developers, "ffmpeg tool (version n-94030-gcaabe1b495)[software]," 2018.

[26] M. Wilbert, "Which Video Streaming Protocol Should You Use?." `https://www.dacast.com/blog/video-streaming-protocol`, 2017. [Online; accessed 03-August-2019].

[27] J. Klaue, B. Rathke, and A. Wolisz, "Evalvid–a framework for video transmission and quality evaluation," in *International conference on modelling techniques and tools for computer performance evaluation*, pp. 255–272, Springer, 2003.

[28] S. Chong, S.-Q. Li, and J. Ghosh, "Dynamic bandwidth allocation for efficient transport of real-time vbr video over atm," in *Proceedings of INFOCOM'94 Conference on Computer Communications*, pp. 81–90, IEEE, 1994.

[29] M. Wu, R. A. Joyce, H.-S. Wong, L. Guan, and S.-Y. Kung, "Dynamic resource allocation via video content and short-term traffic statistics," *IEEE Transactions on Multimedia*, vol. 3, no. 2, pp. 186–199, 2001.

[30] H. Luo, M.-L. Shyu, and S.-C. Chen, "An optimal resource utilization scheme with end-to-end congestion control for continuous media stream transmission," *Computer Networks*, vol. 50, no. 7, pp. 921–937, 2006.

[31] Y. Liu, Z. G. Li, and Y. C. Soh, "A novel rate control scheme for low delay video communication of h. 264/avc standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 1, pp. 68–78, 2006.

[32] G. J. Sullivan, T. Wiegand, *et al.*, "Rate-distortion optimization for video compression," *IEEE signal processing magazine*, vol. 15, no. 6, pp. 74–90, 1998.

[33] I. T. S. Sector and O. ITU, "Quality of experience requirements for iptv services," *ITU-T Recommendation G*, vol. 1080, 2008.

[34] R. I.-R. BT, "Methodology for the subjective assessment of the quality of television pictures," 2002.

[35] S. Chikkerur, V. Sundaram, M. Reisslein, and L. J. Karam, "Objective video quality assessment methods: A classification, review, and performance comparison," *IEEE transactions on broadcasting*, vol. 57, no. 2, pp. 165–182, 2011.

[36] M. Vranješ, S. Rimac-Drlje, and K. Grgić, "Review of objective video quality metrics and performance comparison using different databases," *Signal Processing: Image Communication*, vol. 28, no. 1, pp. 1–19, 2013.

[37] S. Akramullah, *Digital video concepts, methods, and metrics: quality, compression, performance, and power trade-off analysis*. Apress, 2014.

[38] T.-L. Lin, S. Kanumuri, Y. Zhi, D. Poole, P. C. Cosman, and A. R. Reibman, "A versatile model for packet loss visibility and its application to packet prioritization," 2010.

[39] A. Rehman and Z. Wang, "Reduced-reference image quality assessment by structural similarity estimation," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3378–3389, 2012.

[40] K. Yamagishi and T. Hayashi, "Qrp08-1: Opinion model for estimating video quality of videophone services," in *IEEE Globecom 2006*, pp. 1–5, IEEE, 2006.

[41] T. Hayashi, K. Yamagishi, T. Tominaga, and A. Takahashi, "Multimedia quality integration function for videophone services," in *IEEE GLOBECOM 2007-IEEE Global Telecommunications Conference*, pp. 2735–2739, IEEE, 2007.

[42] C. A. Mello, M. M. Saraiva, D. P. Menor, and R. Nishihara, "A comparative study of objective video quality assessment metrics.," *J. UCS*, vol. 23, no. 5, pp. 505–527, 2017.

[43] C. Lee, S. Cho, J. Choe, T. Jeong, W. Ahn, and E. Lee, "Objective video quality assessment," *Optical engineering*, vol. 45, no. 1, p. 017004, 2006.

[44] A. Rehman, K. Zeng, and Z. Wang, "Display device-adapted video quality-of-experience assessment," in *Human Vision and Electronic Imaging XX*, vol. 9394, p. 939406, International Society for Optics and Photonics, 2015.

[45] R. Rassool, "Vmaf reproducibility: Validating a perceptual practical video quality metric," in *2017 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. 1–2, IEEE, 2017.

[46] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, "Toward a practical perceptual video quality metric," *The Netflix Tech Blog*, vol. 6, 2016.

[47] J. Ozer, "Buyers' guide to video quality metrics," *VIDEO PROFESSIONALS MANUAL*, 2019.

[48] A. Khan, L. Sun, and E. Ifeachor, "Qoe prediction model and its application in video quality adaptation over umts networks," *IEEE Transactions on Multimedia*, vol. 14, no. 2, pp. 431–442, 2011.

[49] M. T. Vega, D. C. Mocanu, S. Stavrou, and A. Liotta, "Predictive no-reference assessment of video quality," *Signal Processing: Image Communication*, vol. 52, pp. 20–32, 2017.

[50] P. Gastaldo, S. Rovetta, and R. Zunino, "Objective quality assessment of mpeg-2 video streams by using cbp neural networks," *IEEE Transactions on Neural Networks*, vol. 13, no. 4, pp. 939–947, 2002.

[51] M. Shahid, J. Panasiuk, G. Van Wallendael, M. Barkowsky, and B. Lövström, "Predicting full-reference video quality measures using hevc bitstream-based no-reference features," in *2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX)*, pp. 1–2, IEEE, 2015.

[52] N. Staelens, G. Van Wallendael, K. Crombecq, N. Vercammen, J. De Cock, B. Vermeulen, R. Van de Walle, T. Dhaene, and P. Demeester, "No-reference bitstream-based visual quality impairment detection for high definition h. 264/avc encoded video sequences," *IEEE Transactions on Broadcasting*, vol. 58, no. 2, pp. 187–199, 2012.

[53] N. Staelens, D. Deschrijver, E. Vladislavleva, B. Vermeulen, T. Dhaene, and P. Demeester, "Constructing a no-reference h. 264/avc bitstream-based video quality metric using genetic programming-based symbolic regression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 8, pp. 1322–1333, 2013.

[54] O. Issa, F. Speranza, T. H. Falk, *et al.*, "Quality-of-experience perception for video streaming services: Preliminary subjective and objective results," in *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, pp. 1–9, IEEE, 2012.

[55] M. Fiedler, T. Hossfeld, and P. Tran-Gia, "A generic quantitative relationship between quality of experience and quality of service," *IEEE Network*, vol. 24, no. 2, pp. 36–41, 2010.

[56] T. Hoßfeld, D. Hock, P. Tran-Gia, K. Tutschku, and M. Fiedler, "Testing the iqx hypothesis for exponential interdependency between qos and qoe of voice codecs ilbc and g. 711," in *Proceedings of the 18th ITC Specialist Seminar on Quality of Experience*, pp. 105–114, 2008.

[57] M. Elkotob, D. Grandlund, K. Andersson, and C. Ahlund, "Multimedia qoe optimized management using prediction and statistical learning," in *IEEE Local Computer Network Conference*, pp. 324–327, Ieee, 2010.

[58] H. Du, C. Guo, Y. Liu, and Y. Liu, "Research on relationship between qoe and qos based on bp neural network," in *2009 IEEE International Conference on Network Infrastructure and Digital Content*, pp. 312–315, IEEE, 2009.

[59] V. A. Machado, C. N. Silva, R. S. Oliveira, A. M. Melo, M. Silva, C. R. Francês, J. C. Costa, N. L. Vijaykumar, and C. M. Hirata, "A new proposal to provide estimation of qos and qoe over wimax networks: An approach based on computational

intelligence and discrete-event simulation," in *2011 IEEE Third Latin-American Conference on Communications*, pp. 1–6, IEEE, 2011.

[60] Y. He, C. Wang, H. Long, and K. Zheng, "Pnn-based qoe measuring model for video applications over lte system," in *7th International Conference on Communications and Networking in China*, pp. 58–62, IEEE, 2012.

[61] T. Hori and T. Ohtsuki, "Qoe and throughput aware radio resource allocation algorithm in lte network with users using different applications," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–6, IEEE, 2016.

[62] N. Alliance, "Service quality definition and measurement," 2013.

[63] M. Nasimi, M. Kousha, and F. Hashim, "Qoe-oriented cross-layer downlink scheduling for heterogeneous traffics in lte networks," in *2013 IEEE 11th Malaysia International Conference on Communications (MICC)*, pp. 292–297, IEEE, 2013.

[64] T. Begluk, J. B. Husić, and S. Baraković, "Machine learning-based qoe prediction for video streaming over lte network," in *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pp. 1–5, IEEE, 2018.

[65] "Netflix Public Dataset." `https://drive.google.com/drive/u/0/folders/0B3YWNICYMBIweGdJbERlUG9zc0k`. [Online; accessed 03-August-2019].

[66] "TUM public video Dataset." `ftp://ftp.ldv.ei.tum.de/videolab/public/SVT_Test_Set/1080i/`. [Online; accessed 03-August-2019].

[67] "Recommended Encoding Settings." https://support.video.ibm.com/hc/en-us/articles/207852117-Internet-connection-and-recommended-encoding-settings. [Online; accessed 03-August-2019].

[68] D. Ammar, T. Begin, and I. Guerin-Lassous, "A new tool for generating realistic internet traffic in ns-3," in *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, pp. 81–83, ICST (Institute for Computer Sciences, Social-Informatics and , 2011.

[69] E. Rahm and H. H. Do, "Data cleaning: Problems and current approaches," *IEEE Data Eng. Bull.*, vol. 23, no. 4, pp. 3–13, 2000.

[70] F. Hartwig and B. E. Dearing, *Exploratory data analysis*. No. 16, Sage, 1979.

[71] A. Nandeshwar, *Tableau data visualization cookbook*. Packt Publishing Ltd, 2013.

[72] P. Barrett, J. Hunter, J. T. Miller, J.-C. Hsu, and P. Greenfield, "matplotlib–a portable python plotting package," in *Astronomical data analysis software and systems XIV*, vol. 347, p. 91, 2005.

[73] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

Appendix

.1 Data Dictionary

Column	Description
SimulationId	Each simulation is given a unique Id
videoId	Each video file is given a unique id
Name	Video Name
noOfUe	Number of User Equipments
Bandwidth	Network Bandwidth configuration(RB)
rlcBufferSize	RLC Buffer size(Bytes)
transmissionMode	Transmission mode(0-SISO,1-MIMO)
duration	Video Duration(s)
width	Video Frame Width
height	Video Frame Height
frameRate	Video Frame Rate(fps)
bitrate	Video Bitrate(Kbps)
frameCount	Total Number of Frames transmitter
noOfPackets	Total Number of Packets transmitted
jitterSum	Packet Jitter Sum(nS)
delaySum	Packet Delay Sum(nS)
lostPackets	Packets Lost
throughput	Video Throughput(Kbps)
IframeLossPerc	I-Frame Loss ratio
PframeLossPerc	P-Frame Loss ratio
BframeLossPerc	B-Frame Loss ratio
OverallFrameLossPerc	Total Frame Loss ratio
meanVmaf	VMAF for the video
meanEndToEndDelay	Frame End-To-End Delay
meanSenderInterFrameLag	Sender Inter Frame Delay
meanReceiverInterFrameLag	Receiver Inter Frame Delay
meanCumulativeJitter	Frame Cumulative Jitter

Table 1: Simulation Dataset Data Dictionary