Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

Using Machine Learning to Predict Quality of Experience of Video in LTE Networks

Daria Tsaregorodtseva
Master in Computer Science

Supervisor: Meriel Huggard

Submitted to the University of Dublin, Trinity College, April, 2019

Declaration

I, Daria Tsaregorodtseva, declare that the following dissertation, except where otherwise stated, is entirely my own work; that it has not previously been submitted as an exercise for a degree, either in Trinity College Dublin, or in any other University; and that the library may lend or copy it or any part thereof on request.

Signed: ________________________ Date: ________________________
Daria Tsaregorodtseva

Summary

The prediction of Quality of Experience of video in mobile networks can help with QoE adaptation, and thus is important to service providers and researches. This dissertation applies machine learning algorithms to QoE prediction and uses network Quality of Service parameters as features for the machine learning models. This work also investigates the role of wireless-specific QoS parameters in QoE prediction.

In current research QoE prediction is typically done in two main ways: with media-layer parameters and with network QoS parameters. Media-layer parameters are features and impairments of the output video signal, the computation of which would make such models too resource-intensive for in-service usage. Works that focus on network QoS parameters typically limit their feature space to a few QoS parameters and rarely feature any wireless-specific parameters in their models.

To collect the QoS parameters an LTE simulation was created that attempts to accurately emulate the current and future LTE background traffic landscape and the general environment of video transmission. Quality of Service parameters are collected during the simulation for the target video being transmitted. After the simulation, the transmitted video is objectively evaluated for QoE by the VQM tool. This tool has high correlation with subjective QoE MOS and is widely used in research.

Quality of Service parameters collected and the Quality of Experience scores are used as features and labels respectively in four machine learning models. These models are Support Vector Machines, Random Forest, Gradient Boosted Trees and a Feedforward Neural Network. The performance of these algorithms was evaluated, and all the algorithms achieved RMSE of around 0.1, which is a tolerable error with respect to the MOS scale. SVM performed the poorest on the task of QoE prediction, with Random Forest and Gradient Boosted Trees performing well. Feature importance of wireless-specific parameters was evaluated, and it was found that the number of UE connected to an eNB and the % of the UE streaming video have high importance, while the other wireless-specific parameters have a small influence on QoE prediction. New models of the four algorithms were created which used either delay, jitter and packet loss to predict QoE or the wireless-specific parameters alone, and both of the models performed reasonably well, with the RMSE being only slightly larger for both than in the original model.

It was discovered that machine learning can be successfully applied to QoE prediction. Some wireless-specific parameters were found to have a large impact in QoE prediction, and it was also found that they could be used alone to predict QoE. However, that would not provide and advantage to using delay, jitter and packet loss in QoE prediction models. The potential future works include conducting a subjective study to verify the accuracy of the models, extending the work to TCP video and applying incremental learning techniques for potential in-service deployment of machine learning for QoE prediction.

Using Machine Learning to Predict Quality of Experience of Video in LTE Networks

Daria Tsaregorodtseva

Abstract

With the rapid growth in mobile network usage and video streaming being the most popular service, Quality of Experience of video in mobile networks is of extreme importance to both service providers and their customers. The ability to effectively predict Quality of Experience of video is key for QoE adaptation and higher levels of customer satisfaction.

In this work machine learning algorithms were used to create models that predict QoE with network QoS parameters, including wireless-specific and LTE-specific parameters. An LTE simulation that reflects the current mobile traffic landscape was created to obtain the data set for training. An objective tool for video QoE evaluation was used to gather QoE data necessary to train the prediction models. Support Vector Machines, Random Forest, Gradient Boosted Trees and Neural Networks were chosen as the machine learning algorithms for Quality of Experience prediction, and it was shown that they achieve high accuracy. Influence of wireless-specific parameters on QoE prediction was also investigated, and it was discovered that they are suitable for use in Quality of Experience prediction models.

Acknowledgements

I would like to thank all of my family and friends, but especially my Mother, whose love and hard work made it possible for me to write this dissertation and who will forever remain my main inspiration. I would also like to thank my Grandfather, Yuri Nikolayevich, who has been very supportive and encouraging throughout my academic career but especially during the writing of this dissertation.

I would also like to thank my supervisor Meriel Huggard for providing indispensable guidance and advice during the writing of this work.

Contents

List of Figures

List of Tables

1 Introduction

Quality of Experience is a metric that describes user satisfaction with a given service. The ability to predict and monitor Quality of Experience of video streaming in LTE networks could help achieve higher overall levels of mobile video Quality of Experience by way of timely QoE adaptation. It could also potentially give an insight into which parameters affect QoE the most.

1.1 Motivation

In recent years it has become impossible to ignore the rapid increase in global mobile traffic. It has been estimated that mobile traffic data in 2018 equaled over 225 exabytes, which is an almost 60% increase from 2017. By the end of 2022 that figure is predicted to grow to nearly a zettabyte [11]. In North America and Western Europe alone, the compound annual growth rate of mobile traffic is predicted to amount to 40 % from 2017 to 2022 [11].
With the rapid increase in demand of mobile traffic comes the challenge for service providers to be in step with the growth and to consistently ensure a high quality service. This is particularly crucial for video streaming as it is currently responsible for 58% of all mobile traffic and is also an especially demanding service. It is estimated that the percentage of mobile traffic used for video streaming will grow to 79% by 2022 [12].
Not only is it difficult to provide a service that ensures high quality video delivery, but it has also been shown that users are especially impatient when it comes to online video.
According to recent findings, it only takes 2 seconds of playback stalls for 25% of viewers to

abandon a video, and over 50% of viewers abandon a video after 30 or more seconds of buffering [13], which illustrated just how important Quality of Experience of video is for both service providers and their customers.

In the wake of all of these challenges service providers need to be able to keep up and keep customer satisfaction rates high with high Quality of Experience of all services, but especially the predominant service of video streaming.

Quality of Experience prediction could be one of the methods to help service providers improve video QoE in LTE networks. Machine learning algorithms are frequently used to predict QoE of video by using either media-layer or network-layer Quality of Service parameters. Models that use network QoS parameters are more practical for in-service usage, however wireless-specific parameters are rarely included in these models. The usage of such parameters for QoE prediction could give a better insight into adaptation possibilities for QoE of video in LTE networks

1.2 Research Objectives

The main objective of this work is to apply machine learning techniques to video Quality of Experience prediction in an LTE network. A few machine learning algorithms have been previously applied to QoE prediction in wireless networks, but the works are very limited and typically only focus on a few parameters for QoE prediction. This work aims to train several machine learning models on an adequately large data set of network QoS parameters and their respective video QoE scores. The data used for QoE prediction should competently reflect real-world data, both in terms of QoS parameters and QoE scores, to ensure robust and accurate prediction models. Another objective is to infer the influence of wireless-specific parameters on prediction of QoE of video in LTE networks. This aspect is often neglected in related research despite how tied it is to the setting of the research.

1.3 Dissertation Structure

Chapter 2 of this dissertation provides the necessary background and discussion of related research. In Chapter 3 the design and implementation of this work are presented in detail. Chapter 4 provides the results and evaluation of the work and Chapter 5 discusses the conclusions and potential for future work.

2 Background

Unmet expectations of quality can be frustrating for customers and devastating for service providers, especially when it comes to the most popular service. Being able to successfully monitor and predict Quality of Experience of Video in mobile networks is a critical need for the network operators. Many approaches to QoE prediction in wireless networks have been developed, including intrusive and non-intrusive models as well as models that use machine learning or other techniques.

In this chapter Quality of Experience of mobile video and its assessment methods are described, related work on QoE prediction is presented and background on LTE networks and the machine learning algorithms used in this work is given.

2.1 Quality of Experience

Quality of Experience is a metric used to describe the users' satisfaction with a service and their perception of the service's quality. The essence of Quality of Experience is purely subjective as it is only concerned with the users perspective and not with any technical quality metrics. However, Quality of Experience can be measured both subjectively and objectively. For a subjectively measured QoE, user surveys are required to gather subjective evaluations of a given service. Such surveys are generally costly and time-consuming, and require a variety of demographics and context scenarios to be as accurate as possible. In addition, they typically provide feedback on a post hoc basis.

The most commonly used metric for subjective multimedia QoE evaluation is Mean Opinion

MOS	Quality
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

Table 2.1: MOS scale

Score (MOS). MOS is an ITU standardized 5-point scale [14], on which the values 1-5 correlate to bad, poor, fair, good and excellent, as shown in Table 2.1

The usefulness and accuracy of Mean Opinion Score is debated, mainly in terms of how it is obtained from the users and how it should be interpreted [15]. It still remains the most widely metric used in research and also in industry, for example by Skype, which uses a MOS post-service to measure perceived quality of the call.
Alternatively, objective models are also used to estimate QoE and attempt to do so without human interaction. Objective models are reproducible, more predictable and also are more suited for in-service usage for real-time service monitoring and adaptation, however due to their nature they are likely to be less accurate than subjective models.

2.1.1 Objective Video QoE Assessment

Objective video QoE assessment techniques are necessary to objectively evaluate human perception of video quality and produce a reliable QoE score that would reasonably correlate with a subjectively obtained QoE score. The most popular methods most commonly rely on the input and the output video to assess QoE. A few such methods are currently in use by researchers who study objective Quality of Experience to avoid having to resort to surveys or crowdsourcing especially if large data sets of videos are involved. These include:
Peak Signal to Noise Ration (PSNR) is a dated but traditional metric for evaluating objective QoE. It can be mapped to MOS through the ITU-T J.144 [16] standardized formula and is quite simple to understand and compute. However, it is widely accepted in current research that PSNR does not accurately reflect subjective QoE scores [17, 18] as it only evaluates the changes in the output video compared to the input video and does not

take into account any aspects of human perception.

Structural Similarity Index Metric (SSIM) [19] is a video quality assessment metric that assesses video quality based upon brightness, contrast and structure. Originally SSIM was developed for images, however it was later extended to video [20]. MultiScale SSIM is an extension of the original metric which incorporates analysis of image details at different resolutions and performs better than standard SSIM [21, 22, 23].

Video Quality Metric (VQM) was developed by the National Telecommunications and Information Administration (NTIA) to measure perceptual video quality [24]. It has been standardized by ITU-T [16] and works by performing feature extraction and video quality assessment through parameters such as blurriness, block distortion and noise. Studies have shown that VQM has a high correlation with subjective scores [25, 26].

Video Multimethod Assessment Fusion (VMAF) is a perceptual video quality metric developed by Netflix [1]. The tool relies heavily on machine learning, namely Support Vector Machines (SVM) to maximize its correlation with subjective MOS and does so by using image fidelity metrics as features for the algorithm. Independent studies have shown that VMAF does correlate strongly with subjective MOS scores, but regularly overestimates MOS [27].

Netflix has conducted its own comparison of PSNR, SSIM, VQM_VFD (a version of VQM that employs Neural Networks), and VMAF on four public video datasets: NFLX-TEST [28], LIVE database [29], the VQEGHD3 collection of the VQEG HD database[30] and LIVE Mobile database [31]. In this comparative study it was shown that VQM performs similarly to VMAF, sometimes outperforming it. The root-mean-squared error of the 4 methods applied to the 4 data sets is shown in Figure 2.1. As a result of this study and other similar works it was decided to use VQM in this work to objectively asses QoE of transmitted videos and use them as labels for the machine learning models.

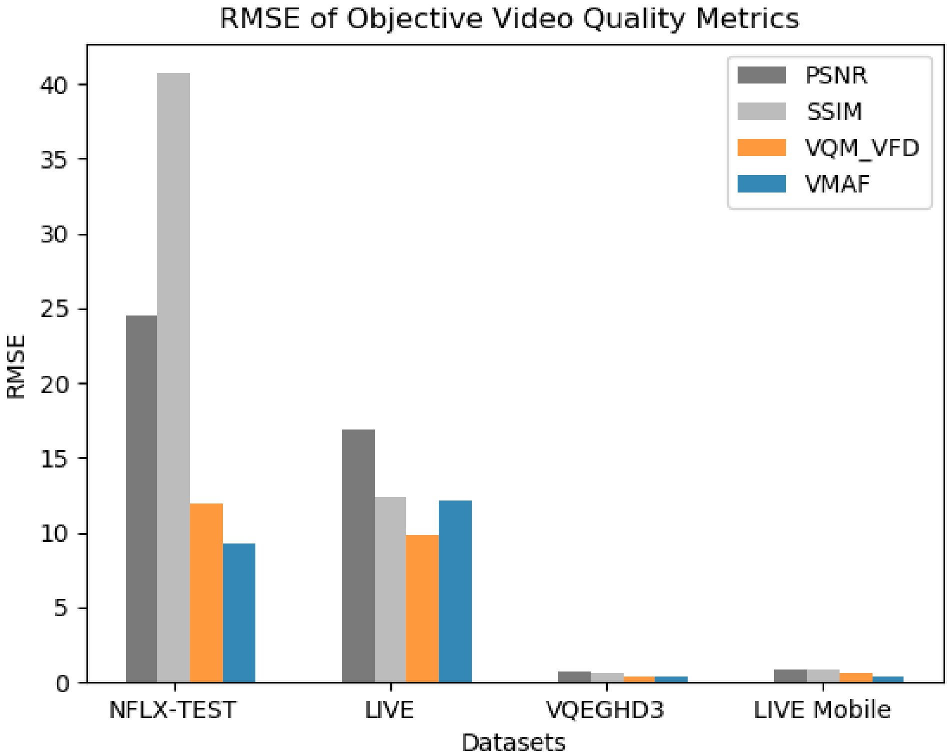

Figure 2.1: RMSE Comparison of four tools for objective QoE evaluation[1]

2.1.2 QoE Evaluation With VQM

The VQM tool was developed by the National Telecommunications and Information Administration (NTIA), which is a US agency, to most accurately evaluate human perception of video quality [32]. The General VQM model is included as a normative method in the ITU J.144 recommendation [16]. There are a few steps performed in VQM to estimate the QoE:

1. Reduced-reference calibration of the target video is performed. This is achieved by estimating the valid region of the video to prevent non-picture areas such as borders from affecting the QoE estimation, determining and correcting spacial and temporal shifts, as well as gain and level offset.

2. Extraction of quality features is implemented by first enhancing particular properties of perceived quality of both the original and the target video streams by applying perceptual filters. Then various mathematical functions are used to extract the

features from spatial-temporal sub regions of the target video. The quality features extracted in this step contain information on noise, unnatural motion and jerkiness, blurring, blocking, colour distortion and other similar elements that affect human perception of video quality. A threshold is applied to prevent measurement of imperceptible impairments.

3. Quality parameters are calculated from the quality features to represent overall video distortion by comparing quality features of the original and the target videos. The General VQM model calculates seven independent quality parameters.

4. VQM General Model takes the quality parameters calculated in the previous step to compute VQM. The score produced by the model ranges from 0, which signifies no perceptible quality impairments, to 1, which represents the maximum perceived quality impairment.

Experiments to validate the VQM General model were conducted with 1536 subjectively measured video sequences, and resulted in an overall Pearson linear correlation coefficient of 0.948 between the subjective scores and objective scores obtained with the VQM model. The VQM score maps linearly to MOS.

Seven years after the General VQM model was developed, NTIA came out with a VQM Variable Frame Delay model [33]. This model is of particular interest due to the use of Neural Networks in the model and its increased accuracy.

The implementation of the VQM VFD model is very similar to the General model. The steps detailed above are all present, with the first main difference being that there is an extra quality parameter calculated which represents the impact repeated and dropped frames have on perceived video quality. The other, arguably more impactful change is the application of Neural Networks in mapping the calculated quality parameters to the final VQM value. The Neural Network used was implemented in Matlab and was trained on 9000 video clips of different sizes and their subjective ratings.

This model generally achieves a higher degree of accuracy in predicting subjective scores [1, 23, 33], which is the reason why it was chosen as the objective QoE evaluation tool for this work.

2.2 Related Work in QoE Prediction

The ability to predict the customers' Quality of Experience would enable service providers to carry out efficient and timely resource allocation and adaptation to ensure a higher quality service with reduced cost. In particular, it would allow them to focus only on parameters that affect the users perception of the service and reduce the need for maximization of all QoS. Most recent approaches for QoE prediction rely on statistical and probabilistic techniques as well as machine learning and artificial intelligence, and use different types of features for QoE prediction.

Quality of Experience prediction models can be intrusive and non-intrusive, where intrusive models predict QoE by extracting features from the output signal, either on its own or by comparing it with the input signal [34], while non-intrusive models rely on network and application parameters [14, 35].

The most common models of multimedia QoE prediction in wireless networks in current research can be roughly split into Media-Layer models and Parametric models, which are two of the five types of QoE evaluation models specified in ITU J.144 [14]. Media-Layer models are intrusive models which typically rely on output multimedia signal features for Quality of Experience prediction in mobile setting. In these works most popular parameters used to predict Quality of Experience are buffering ratios [36, 37, 38], video playback stalls [39, 40] and ratio of uninterrupted viewing [37, 39, 40]. These models frequently employ small scale studies to get subjective QoE scores, and Decision Trees and Random Forest seem to be very popular machine learning algorithms in Media-Layer QoE prediction [36, 37, 38, 39]. These models do achieve high accuracy in predicting QoE, but are predictively too resource intensive for in-service deployment. They also do not take into account the wireless nature of the multimedia signal transmission.

The Parametric Models use network Quality of Service parameters for Quality of Experience

prediction. They are more suited for in-service QoE monitoring and prediction and give a better insight into possible QoE adaptation.
This dissertation focuses on network Quality of Service parameters for Quality of Experience prediction, and thus can be considered a Parametric model, so it is be beneficial to describe related work in more detail.

2.2.1 Parametric Models

As mentioned above, parametric models rely on network Quality of Service parameters for Quality of Experience prediction. In the context of telecommunications, Quality of Experience (QoE) and Quality of Service (QoS) of networks are closely related but are recognizably separate from each other. Quality of Service is a term that characterizes the technical aspects of the service's performance while Quality of Experience only describes the users perception of a service's quality.
QoS cannot be exclusively used in assessing a service's end user experience since QoS does not linearly map to the users perception of quality as it does not take into account any human-related factors. Different Quality of Service might end up resulting in the same QoE due to the context, device, type of service or human inability to distinguishing small changes in quality.
Network Quality of Service can be more thoroughly described as a set of methods, parameters and characteristics that manage a network flow. QoS is the main instrument to reaching a desired QoE and is what QoE relies on the most.
Jitter, delay, latency and packet loss are some of the most popular parameters that are used to measure and describe network Quality of Service and are relied on heavily for QoE prediction in wireless networks, however most often only a few Quality of Service parameters are used and wireless-specific parameters especially are used very rarely.
Works that use QoS parameters for QoE prediction commonly use Machine Learning and can be split by the type of algorithm they choose to use.

Non-Machine Learning approaches

There are some recent works on QoE prediction in wireless networks that do not use machine learning. Khan *et al.* [41] proposes a model for QoE estimation based on content clustering and linear regression. This is a hybrid model, using both media-layer and network QoS. The prediction focuses mainly on video attributes, namely the video content type, which is extracted with content clustering. Then linear regression is used to design an equation which calculates MOS based on content type, the sending bitrate, the frame rate and the packet error rate. According to the results presented in the paper, video content type has a significant effect on Quality of Experience. However, very few QoS parameters were used in the model which can prove to be ineffective as it is not sufficiently exhaustive. In the case of [41] the addition of other QoS parameters might offer an improvement in QoE prediction. CaQoEM is an approach developed by Mitra *et al.* [42] that uses Bayesian Networks, which is a type of probabilistic model, and utility theory to predict QoE in mobile networks. The work focuses heavily on context variables, such as user location and the type of device the video is viewed on. Context can be an influential factor in the user's Quality of Experience [43], but it has not been shown to be more important than the QoS. In [42], Quality of Service parameters such as jitter and delay were reduced to three states: 'good', 'fair', and 'poor' and their influence on Quality of Experience was not widely explored, as they were not the main focus of the research.

Reinforcement Learning Approaches

Reinforcement Learning (RL) is a machine learning technique that employs an agent which learns how to act in an environment based on the rewards and punishments it receives for its actions. Reinforcement learning is particularly suitable for problems that require decision making and has been successfully applied to several networking problems [44, 45, 46]. Canale *et al.* [47] applied Reinforcement Learning to Quality of Experience prediction by having the RL agent make changes to QoS affected parameters, and then receive a reward based on the resulting QoE, which was measured by a MOS. QoS is represented by a function that maps the usual QoS parameters (bandwidth, delay, jitter etc.) to a value in the range of [0, 1], with a higher QoS corresponding to values close to 1. This study provides a way to model the QoS/QoE relationship based on user behavior, but does not provide a deeper look into the effect of specific QoS parameters on QoE.

Neural Networks Approaches

A Neural Network is a Machine Learning method that is loosely inspired by the structure of the human brain. Neural Networks learn to transform inputs into higher level features useful for the task they are being trained on. Of particular relevance for this work is the fact that they are suitable for mapping non-linear relationships, such as the one between QoS and QoE.

Begluk, Husić and Baraković [48] used a FeedForward Neural Network to create a model for predicting MOS for wireless video transmission. The NN approach has also been one of the components of the MLQoE tool [49], which has a high performance rate in the VoIP QoE calculation in IEEE802.11. Neural Networks have also been used in non-wireless QoE prediction [50], [51]. In [48] The LENA NS3 module was used to simulate the network, and the EvalVid framework was utilized to evaluate the MOS of each transmitted video. The EvalVid framework relies on PSNR for MOS calculation. Delay, jitter and packet loss were chosen as features for training the Neural Network, as they are the most common parameters for QoE prediction in research. To verify the MOS predictions, a small-scale subjective study was also performed. The main contribution of [48] is the implementation of real-time optimization of MOS prediction by the NN, which was shown by this study to be reasonably effective in the conditions of this study. Further research is needed to verify the effectiveness of the proposed prediction approach when expanding MOS prediction to include more QoS parameters and using a more reliable MOS calculation method.

He *et al.* [52] proposed using a Probabilistic Neural Network(PNN) to estimate the QoE of Video transmission over an LTE network. A PNN is a Feedforward Neural Network with two hidden layers and is commonly used in classification problems. The network was simulated in

OPNET, and MOS was used to quantify the Quality of Experience. In the work, end-to-end delay, delay jitter, packet loss rate and mean loss burst size were calculated for each transmitted video, and were used as features for the PNN. The Mean Opinion Scores were collected by a small-scale subjective study. PNNs are fitting for the task as they are well suited to pattern recognition and non-linear mapping problems. In addition, their fast training times make them a good candidate for online deployment. This approach seems to have highly accurate results, however, it is heavily dependant on MOS scores collected from a small group and only uses a few QoS parameters for QoE estimation.

Decision Trees

Decision Trees have also been successfully applied to modelling the QoS/QoE relationship. Menkovski *et al.* [53, 54] have considered Support Vector Machines and Decision Trees for QoE prediction, and have ultimately decided that decision trees are more suitable for the task. Other work in the wired domain seems to support this choice [55]. Menkovski *et al.* have developed an Online Learning model based on Hoeffding Option Trees, which are a type of Decision Tree that enables Online Learning due to the training data being processed one datapoint at a time with no need for the training set to remain in memory. The method uses real-time user feedback in its online learning, and is one of the few works on Quality of Experience prediction that takes a fully online approach. However, they only use video received bitrate, audio received bitrate and framerate as features for their algorithm, as well as using a binary score of "acceptable" and "unacceptable" for QoE measurement, which is a very simplified way of measuring how real users view QoE.
Decision Trees are also used as one of the methods in MLQoE [49], which is a modular approach to QoE assessment that utilizes multiple machine learning algorithms to assess QoE of VoIP in wireless networks. Average Delay, packet loss, average jitter and other similar metrics are used as features for the models, and in the work Decision Trees generally performed at a similar level with Neural Networks. This work is quite unique due to creating a modular system, and is geared towards VoIP and WiFi.

Other Supervised Learning Approaches

In the work by Aggarwal *et al.* [56] a tool called *Prometheus* was developed. It is a prototype that predicts QoE and playback stall periods of mobile video and VoIP based on network parameters such as packet loss rate and throughput. The video quality parameters are collected from applications installed on the user device. A slightly modified MOS scale is used for QoE evaluation, and LASSO regression is the technique chosen for QoE prediction. A comprehensive number of network QoS parameters is used in the model,however the methods of collecting video quality parameters from apps would be impractical for service providers due to the ever-growing number of apps providing video-on-demand and VoIP services, as well as the possible unwillingness of these apps to provide access to their data.

2.3 LTE Overview

2.3.1 LTE Mobile Network Overview

Long-Term Evolution (LTE) is a 4th Generation (4G) mobile communications standard developed by 3GPP. It provides high peak data rates, low latency and flexible bandwidth operation. The architecture of the LTE Network is shown in Figure 2.2.

The Evolved Packet System (EPS) is the central part of the LTE network that consists of the Evolved Packet Core (EPC) and the Evolved Universal Terrestrial Radio Access Network (E-UTRAN). The EPC is the core network of LTE and consists of various telephony switches, which connect the mobile devices and the mobile network to the Internet [57]. The EPC consists of the following five main nodes:

- The **Serving GateWay (SGW)** connects the E-ULTRAN to the EPC and is responsible for connecting the user terminals to the external IP networks. It is connected to the PGW and they are commonly combined in the same physical network device.

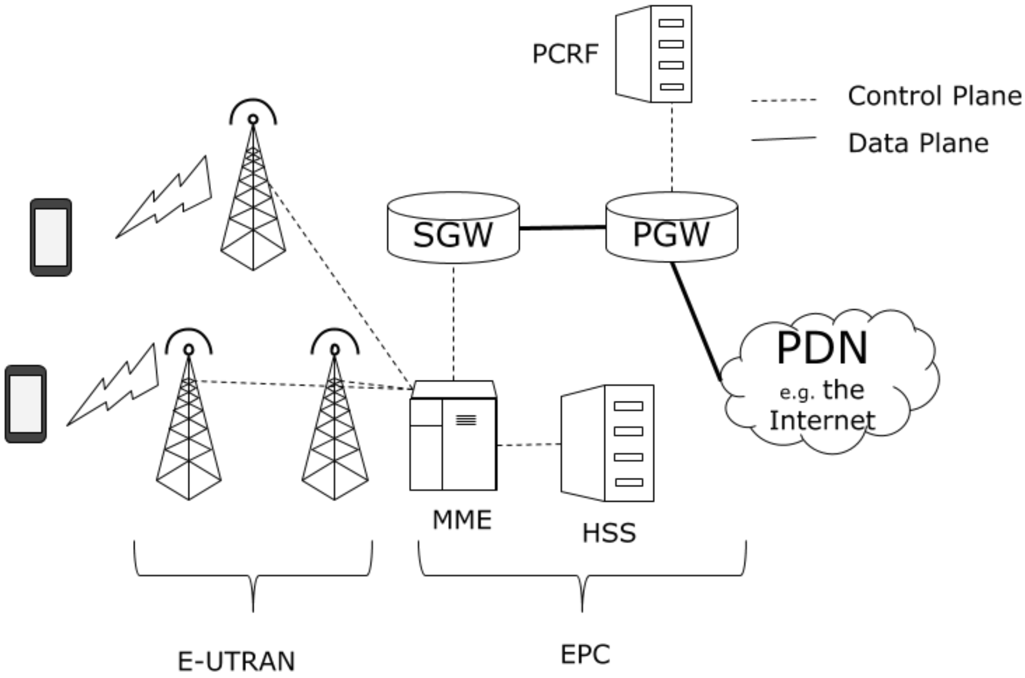

Figure 2.2: LTE Architecture
Based on materials from 3GPP [2] and figures in [3]

- The **PDN GateWay (PGW)** connects the EPC to the external IP networks, or Packet Data Networks, and is responsible for routing the traffic between them. It also deals with tasks such as IP address allocation, packet filtering and policy enforcement.

- The **Policy and Charging Rules Function (PCRF)** deals with charging in the EPC and real-time policy rules control.

- The **Home Subscriber Server (HSS)** is the database that contains subscriber-related data and other data relating to mobile users. It also performs actions such as user authentication and call and session setup.

- The **Mobility Management Entity (MME)** is the core control node in the EPC and performs mobility, roaming and tracking management functions among others.

The E-UTRAN controls the radio connections between the Evovled Node B (eNB or EnodeB) and the user terminals, or UE (User Equipment). The eNBs' are the base stations that connect the users' mobile devices to the EPC via a radio interface.

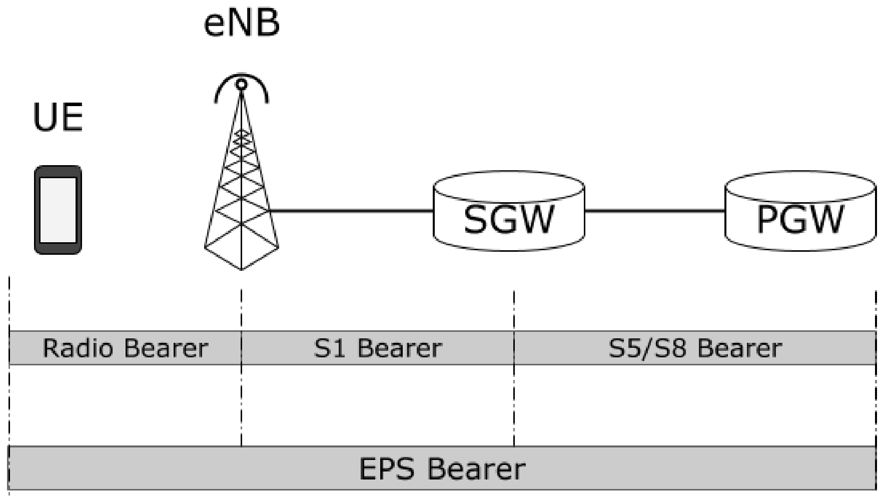

Figure 2.3: LTE QOS Bearers, Recreated with reference to [4]

Quality of Service in LTE

The Quality of Service mechanism in LTE attempts to provide seamless connectivity by means of prioritizing various packets in the network based on their type of service. For example, Voice over LTE (VoLTE) and mobile gaming get higher priorities due to them being more sensitive to delay, jitter, etc. than Email or web browsing. The LTE QoS is implemented between the UEs and the PDN and is achieved with the help of bearers. A bearer is a concept that describes a set of configurations for the transmission of a particular traffic flow between LTE network interfaces.
An EPS bearer refers to a bearer between PDN and UE and is a concatenation of bearers that exist between User Equipment and eNB (Radio Bearer), eNB and SGW(S1 bearer), and SGW and PGW (S5/S8 bearer) [57]. A QoS Class Identifier (QCI) is assigned to each bearer and determines the priority of the bearer's traffic, its packet delay and packet loss budgets as well as whether the bearer has a Guaranteed Bit Rate (GBR) or not. There are 9 QCI's in total, each with a different priority [10]. These are detailed in Table 2.2.

When an LTE UE first attaches to the network a default bearer is established, and it has QCI 9 and all of this QCI's related parameters, including the lowest priority and no GBR. A dedicated bearer can be assigned on top of the default bearer for services that require a higher priority, like Voice over LTE or video streaming. 4 out of 9 QCI's provide a

QCI	Resource Type	Priority	Packet Delay Budget	Packet Error Loss Rate	Example Services
1	GBR	2	100 ms	10^{-2}	Conversational Voice
2		4	150 ms	10^{-3}	Conversational Video (Live Streaming)
3		3	50 ms	10^{-3}	Real Time Gaming
4		5	300 ms	10^{-6}	Buffered Non-Conversational Video
5	Non-GBR	1	100ms	10^{-6}	IMS Signalling
6		6	300 ms	10^{-6}	Buffered Video, TCP-based services
7		7	100 ms	10^{-3}	Voice, Live Video, Interactive Gaming
8		8	300 ms	10^{-6}	Buffered Video, TCP-based services
9		9	300 ms	10^{-6}	TCP-based services

Table 2.2: QoS Class Identifier Values [10]

Guaranteed Bit Rate. The mapping between QCI, its priority and the services it is assigned to has been determined by 3GPP [10]. Guaranteed Bit Rate values are not determined by 3GPP, and instead are set by the service provider.

2.4 Machine Learning Overview

Machine learning algorithms are powerful tools in creating prediction models. To choose the algorithms to use in this work, four popular supervised learning algorithms that are suited to non-linear mapping problems were chosen. These algorithms were Support Vector Machine (SVM), Random Forest, Gradient Boosting and Neural Networks. Each of these algorithms and its application is going to be described in detail in this section.

2.4.1 Support Vector Machines

Support Vector Machines (SVM) is a popular supervised learning algorithm. SVMs are maximum margin classifiers. In particular, linear SVMs seek to find a hyperplane in the dataspace that separates the data into its respective classes and maximizes the distance between the data points of different classes that are closest to this separating hyperplane. For example, when there are two classes and the data is two-dimensional, this would consist of finding a line which separates the data into the two classes and where the two vectors of

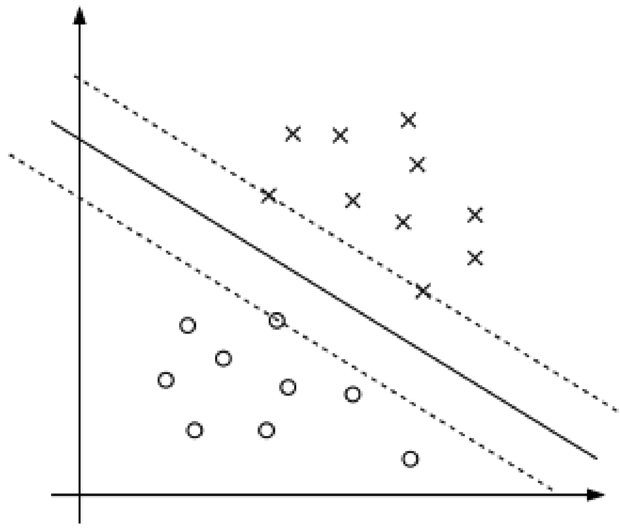

Figure 2.4: Linear SVM [5]

different classes closest to the line are furthest away from each other [58]. This is shown in Figure 2.4.

Most real world data sets cannot be linearly separated. To tackle these data sets with SVMs, the data can be projected into a higher dimensional space and then the separating hyperplane can be learned in this space. Figure 2.5 shows an example of projecting data into multi-dimensional space so that it becomes linearly separable. However, finding this separating hyperplane in high dimensional space would quickly become computationally intractable when done naively. This can be avoided by using what is called a kernel trick, where the algorithm is reformulated so that there is no need to explicitly represent the higher dimensional space [59].

A separating hyperplane can be efficiently learned with the kernel trick. In order to use the kernel trick, a kernel, which is a function of two vectors $k(x, y)$ and which gives a measure of distance between two vectors must be specified. A popular kernel is the RBF kernel which enables learning an infinite dimensional separating hyperplace while only ever computing dot products in the original dataspace [60].

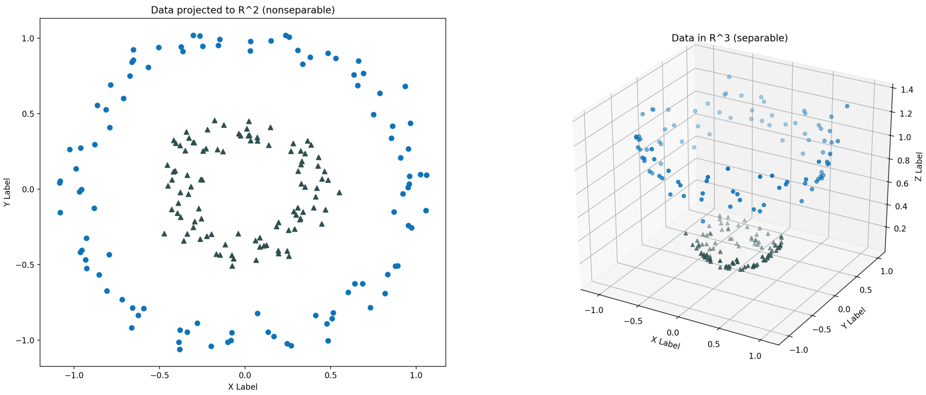

Figure 2.5: Example of data projected into 3 dimensions for classification with SVM. Recreated from [6]

2.4.2 Random Forest

Random forest is a supervised learning algorithm where the model is created by an ensemble of Decision Trees. In short, a Decision Tree works by formulating a set of rules to use for prediction from the features and labels of the training data set. It can be described as a flowchart of 'yes' or 'no' questions that eventually lead to a predicted class or continuous value. The specifics of how the questions, or splits of nodes, are elected is dependant on the type of Decision Tree. Most commonly in cases of classification, the splits of nodes are chosen to maximize the reduction in Gini Impurity of their answers. Gini Impurity is a relatively simple mathematical concept that represents the probability of a randomly chosen element of the set being incorrectly labelled if it was labeled by a distribution of samples in the set. In cases of regression, mean squared error (MSE) is commonly used to measure quality of a split [61]. In a Decision Tree, at each node the algorithm searches through all of the possible features to find the one which would result in the greatest Gini Impurity or MSE reduction, and then chooses it to split on. This splitting procedure is repeated recursively until the tree reaches maximum depth, which is when each node only contains samples of one class and is completely pure or has the lowest possible MSE.

An issue with decision trees is that they are high variance methods and can fit noise in the dataset well, resulting in very different trees being learned for moderately different splits in the dataset. This results in severe overfitting to the training data and poor generalization

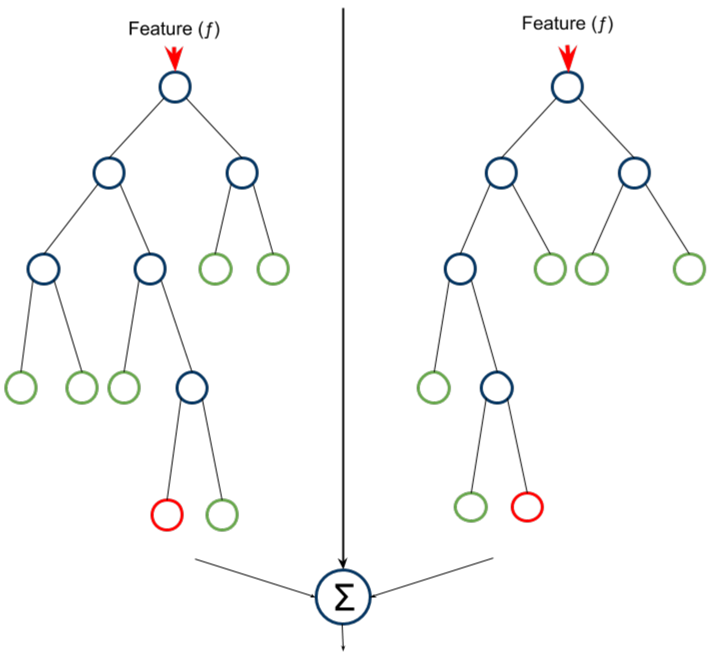

Figure 2.6: Random Forest with two estimators. Created with refernce to [7]

performance [62].

An approach to countering overfitting for high variance machine learning models is bagging, where an ensemble of models are trained on different random samples of the dataset. Random Forests is the application of bagging to decision trees. The algorithm selects a random subset of training data for each Decision Tree, and selects a random subset of features for splitting nodes. When a tree in a Random Forest picks a random sample of training data points they are drawn with replacement, which is known as bootstrapping, and the predictions of each tree in the Random Forest are averaged at test time. This procedure is known as bootstrap aggregation, or bagging [63]. An illustration of Random Forest with two estimators is shown in Figure 2.6

2.4.3 Gradient Boosted Trees

Gradient boosting is a general technique similar to bagging that can be used to create an ensemble of models. While bagging is used to reduce overfitting of high variance models, Gradient Boosting is used to increase the power of high bias i.e. weak models that fail to fit the data well when used individually.

Unlike in bagging, for Gradient Boosting the ensemble of models is trained sequentially rather than in parallel. In the case of Gradient Boosted Trees, which is the algorithm used in this work, Decision Trees are used as the weak model [64].

What sets Gradient Boosted Trees apart from the Random Forest algorithm is that the trees are not random and independent of each other, but rather they are built sequentially, and each new tree attempts to minimize the loss function, for example MSE, of all the trees combined. It is often the case that individual models in the ensemble become good at explaining data in a particular subspace of the dataspace anda good fit to the full dataspace can be achieved by combining all of these specialized models. Gradient Boosted Trees are quite efficient and do not use a lot of memory.

2.4.4 Neural Networks

Neural Networks (NN) are a long established machine learning algorithm that have recently become very popular as it has become possible to train very large Deep Neural Networks. A Neural Network can be made up of several layers, which are in turn made of nodes. The nodes are supposed to roughly model human brain neurons in their function. A node in the Neural Network receives inputs, which have associated weights that represent the input's importance relative to the other inputs to this node and then the node computes its output based on its Activation Function(e.g. a sigmoid) using inputs and their weights [65].

This work uses a simple Feedforward Neural Network. A fully-connected Feedforward Neural Network, or a Multi-Layer Perceptron, usually consists of an input layer, hidden layer(s) and an output layer. The input layer does not perform any computation and just passes the information onto the hidden layer. Hidden layers and output layers do perform computation, with the last hidden layer's nodes passing their outputs to the output layer, which produces the final result value. In a Feedforward Neural Network nodes from adjacent layers are linked by weighted connections, or edges, and the information only goes in the

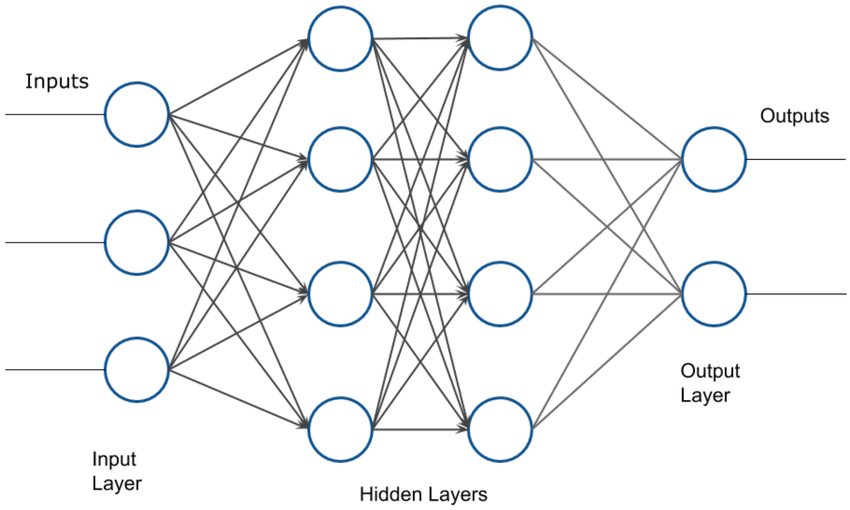

Figure 2.7: Feedforward Neural Network. Recreated with reference to [8]

forward direction, from one layer to another, hence the name Feedforward Neural Network. A simple illustration of a Feedforward Neural Network is show in Figure 2.7. Neural Networks typically use Back Propagation to learn the weights of the Network. In Back Propagation, the weights start off being random. Every input in the training data set is propagated through the NN, and the output is compared with the corresponding label[66]. Then, based on the error the weights are adjusted using the gradient descent optimization algorithm. This process repeats until the error is low enough, and after it terminates the NN has learned all of its weights and can be used for its intended purpose .

2.4.5 Background Summary

Quality of Experience is a complex metric that can be measured both objectively and subjectively. Prediction and monitoring of QoE in wireless networks is a relatively popular topic in research due to its potential usefulness in improving customer satisfaction, especially in multimedia streaming. Machine Learning algorithms have been applied to QoE prediction to aid in modelling of the relationship between QoS parameters and QoE. Supervised

Learning algorithms are mainly used, especially Neural Networks and Decision Trees. Most works on the topic that follow the non-intrusive approach only use a few QoS parameters and do not usually include any wireless-specific QoS parameters, and very few works elaborate on feature impact, both of which could be important factors in QoE in wireless networks.
LTE and the LTE QoS mechanism was explained in this chapter, along with an overview of the machine learning algorithms used in this work, which are Support Vector Machines, Random Forest, Gradient Boosted Trees and Neural Networks.

3 Design and Implementation

Quality of Experience prediction is a complex task. There are a wide variety of ways to design and implement a QoE prediction model, depending on the domain, application type, goals and the type of data chosen for the prediction. In this chapter the approach chosen for use in this work will be presented and detailed.

3.1 Design Overview

Since Video Streaming has been recognized as the key domain for Quality of Experience prediction, it was necessary to be mindful of this when designing the evaluation system at the core of this work. Overall, the system consists of the following components:

1. **Wireless Network Simulation**: The chosen wireless environment was a 4G cellular technology, LTE. In order to be able to train a Machine Learning model a data set of features and labels is required. In this work a data set of QoS parameters and respective QoE scores was obtained in an LTE simulation environment.

2. **Output Video Quality Evaluation**: After the video is transmitted over the simulated network, its has to be evaluated to produce a reference QoE score to have the ability to train the Machine Learning Model. For this work, the NTIA VQM QoE evaluation metric was chosen.

3. **Machine Learning Models**: Several Machine Learning algorithms were applied to QoE prediction, namely Support Vector Machines (SVM), Random Forest, Gradient Boosted Trees and a Feedforward Neural Network. The performance of these

algorithms was then compared and evaluated.

3.2 LTE Simulation

For this project a simulation of an LTE wireless network was created to gather the data necessary for predicting video Quality of Experience, however it is important to point out that simulation is not the only method of getting the data necessary for QoE prediction. The works that fall under media-layer models of QoE prediction described in the previous chapter frequently employ public video databases and subjective studies to evaluate the QoE of the videos from those databases [38, 39, 40]. As mentioned in the previous chapter, this type of intrusive approach is not particularly realistic for in-service deployment, and also does not explore the effect network parameters and configurations have on QoE.

It is also possible to acquire access to a public data set of QoE and its corresponding Influence Factors, however using a public data set would set strict boundaries in terms of the QoS parameters it has available and the type of QoE metric it provides.

The choice of wireless network was motivated by was the desire to keep the findings topical and relevant. LTE was ultimately chosen due to several factors. Firstly, according to several US studies cellular networks have overtaken WiFi in popularity. This is attributed to the prevalence of affordable unlimited data plans [67]. LTE also has the primary share of all cellular traffic, and it is predicted to stay this way for the next five years at the least. In contrast, 5G is predicted to only account for 3.4 percent of all mobile connections by 2022 [11].

There are a number of network simulation frameworks available [68, 69] that provide tools for LTE simulation, however NS-3 [70] was chosen for this work as it includes all of the necessary features and models. In addition it has a longstanding reputation within the networking community. The LENA LTE module was used with NS-3 for LTE simulation.

3.2.1 NS-3 Overview

NS-3 is a discrete-event network simulator that is mostly used for research and educational purposes. It is open source and is available for commercial and non-commercial use under the GNU GPLv2 license. NS-3 provides a simulation platform for networking and is designed as a set of software libraries that can be linked to the user program to create the simulation [71]. The user programs should be written in C++, or, if all the used libraries support it, it can also be written in Python.

NS-3 is the successor of the very popular NS-2 network simulation tool. NS-3 is not an extension of NS-2, but rather a tool written from scratch to better adhere to the requirements of network modelling for research purposes. NS-3 was designed with a goal of making its elements similar in operation and implementation to the real thing, and to make sure that simulation results diverge as little as possible from experimental results [68].

NS-3 is composed of various modules which consist of models of network elements that can be found in computer networks. Some notable examples of such models are Network Nodes, Network Devices, Communication Channels and Communication Protocols. It also provides helper objects, such as attributes for network element configuration, random variables and trace object facilities to aid in the creation of the simulation and result analysis. The Applications module provides resources to create traffic on the network and can be installed on nodes and configured to provide the desired traffic pattern.

The creation of a C++ NS3 simulation begins with creating the network topology. This involves creating and configuring all of the elements in the desired simulation such as nodes, devices and channels by instantiating the corresponding C++ objects. Creation of data demand on the network is the next step. To achieve this the necessary network application models should be created to simulate sending and receiving information as well as the creation and processing of packets. Next, the simulation can be assigned a *stop time* and then executed.

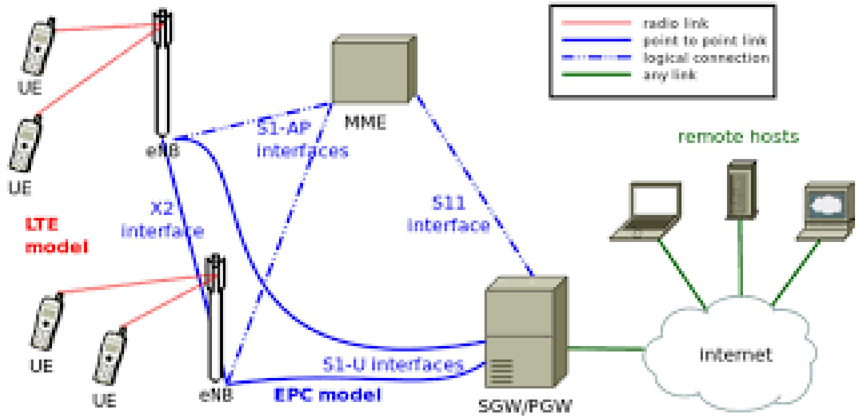

Figure 3.1: LENA LTE Model [9]

3.2.2 LENA NS-3 LTE Module

The LENA LTE Module is an open-source software module for NS-3 that provides the library to simulate the LTE-EPC model. The simulation model consists of two components, the LTE Model which exists within the UE and eNB nodes, and the EPC model which simulates the core network and its entities, interfaces and protocols [9].

The LTE model allows scaling up to tens of eNBs and hundreds of UE, as well as provides a Resource-Block level granularity at the radio level. In general, the level of detail of the LTE model allows for correct evaluation of QoS-aware Packet Scheduling and Radio Resource Management. It can be used alone in the simulation, or together with the EPC model.

The EPC model allows for simulation of an end-to-end IP network over the LTE model and allows Internet connectivity for UEs. The LENA LTE Model is presented in Figure 3.1.

There are some simplifications of a typical LTE core network in the EPC model. Namely, only IPV4 PDN is supported, the SGW and PGW functionality has been combined into one node and the EPC control plane is modeled in a simplified way by implementing direct interaction between some simulated entities through just one pair of interfaces. This was done, for example, for the MME and SGW. However, the EPC model provides the means for simulating end-to-end performance of realistic applications working on top of UDP or TCP and provides the ability for UEs to have application with different QoS profiles. The data plane of the EPC in general is implemented in great detail.

The LENA module also provides several models to achieve more realism in the simulation, such as the Propagation model which allows the addition of buildings and more realistic path loss and fading to the LTE simulation.

3.2.3 LTE Simulation Implementation

The LTE simulation for this work was written in C++ using the NS-3 network simulator and the LENA LTE-EPC Module described above. It was designed with the goal of closely modelling the current mobile network landscape and to most realistically emulate the environment in which a video is streamed over an LTE Network. The simulation was required to produce the QoS parameters and the video traces of the network flow that transmitted the target video.

Topology

The topology of the simulation is relatively simple. The EPC and the LTE models are instantiated and linked. One eNB is created. Then, a random number of UE nodes, from 3 to 70 is created and they are attached to the eNB. All of the UE nodes are placed various distances away from the eNB and are configured to be mobile and be moving in random directions.

Network Traffic

After the topology is created, traffic needs to be generated on the network. This includes transmitting the video for which the QoS data will be collected and creating background traffic.

First, the UEs are split into groups based on their application type. One UE, which can be considered the main UE, is going to be streaming the video which will be later evaluated for QoE, and the QoS data of the flow will be collected during the simulation. The rest of the UE are creating background traffic.

A random percentage of the UE are streaming video. The percentage range is between 30%

and 80% to reflect the current and future mobile traffic landscapes [12]. 0% to 20% are VoLTE nodes, and the rest are generating other regular web traffic.

To create all of this traffic remote hosts are created for each UE. Internet connectivity is created between the remote hosts and the EPC PGW node, and the IP stack is installed on all UEs. For the main UE's video transmission the Gercom's Evalvid model for NS-3 is used [72]. The Evalvid client is installed on the UE, and the Evalvid server is installed on a remote host, specifying the video trace file to be transmitted to the main UE.

The NS-3 Evalvid module works in a similar manner to a typical NS-3 Application and facilitates the transmission of video over the network from a remote host to a UE through UDP sockets. The video is transmitted in the form of a frame trace file. The process of creating the trace file from a viewable video is detailed later in this chapter. The Evalvid NS-3 module was chosen over other NS-3 tools for LTE main video transmission since it produces the receiver frame trace file which makes recreating the received video straightforward.

The goal of the background traffic in the simulated network was to create a realistic simulation scenario of video streaming in a cellular network. For this reason the traffic on background UEs was created to be as realistic as possible. For the background video streaming nodes, the NS-3 UdpTraceClient class was used to enable video streaming for every node. UdpTraceClient dynamically changes its generation rate based on a frame trace file of an MPEG4 video to emulate a video streaming application and is perfect for scenarios that do not require the transmitted video to be recreated. The frame trace file chosen was from a video frame trace library created by the Telecommunication Networks Group at the Technische Universität Berlin [73]

VoIP traffic in NS-3 can be modelled by the OnOffApplication, which generates traffic in an On/Off pattern, where traffic is generated during the On times and no traffic is generated during the Off times. The implementation was based on the model by Hassan *et al.* [74] which states that voice traffic typically has an active ON period with mean duration of 0.352 seconds and and inactive OFF period of 0.65 seconds. We assume that the codec used is G.711, which produces output of 64kb and that packetization delay is 20ms, which would

QoS parameters	Possible Values
Number of UE	3 - 70
% Video UE	30-80 %
% Voice UE	0-10
GBR	40 - 400 kbps
QCI	4, 6, 8
Lost Packets	N/A
Delay (seconds)	N/A
Cumulative Jitter (seconds)	N/A

Table 3.1: QoS parameters collected during the simulation

result in a packet with 160 byte payload and 12 RTP header bytes, overall 172 bytes [75]. This is used as the payload attribute in the OnOFF Application.

The traffic on the UEs is supposed to loosely emulate web traffic, where packets are send at random intervals simultaneously in the uplink and in the downlink directions.

The Lena module allows for the configuration of bearer-lever QoS. It allows for the creation of a bearer with a specified GBR, MBR and QCI. In Table 2.2 it was shown that 3GPP allows for 3 possible QCI priorities for Buffered Video Streaming, only one of which provides a Guaranteed Bit Rate. A dedicated bearer is created for the main UE, which is then configured to be randomly assigned one of the 3 QCI's. In the case of it being QCI 4, the GBR is also randomly assigned in the range of 40kbps to 400kbps, which is an extreme range not quite representative of actual industry GBR values, however it was decided to have this range mainly to see the full effects of GBR on QoE, especially since GBR specification is determined by the LTE service providers.

The FlowMonitor module was used to collect some of the Quality of Service parameters. These include packet loss, the cumulative jitter and delay of the main video streaming application. All of the parameters collected during the simulations were streamed into a CSV file. For each simulation the parameters collected are presented in Table 3.1.

Jitter, delay and packet loss are very obvious parameters to use for QoE prediction and are always present in QoE prediction models. The other parameters are wireless-specific and their effect on QoE is one of the aspects investigated in this work. The number of UE per eNB and the percentage of those using popular high-priority services allows an insight into how possible congestion and different levels of priority per eNB affect Quality of Experience

of a particular user. The GBR parameter is especially intriguing since it is set by service providers and thus is fully under their control and can be used for QoE adaptation. The QCI affects the priority with which the flow is treated and thus can provide an insight into how that affects video QoE, since buffered video streaming can be assigned one of 3 possible QCI. Some other parameters were also considered at different stages of the design process. One of them was Maximum Bit Rate (MBR), which is the maximum allowed bitrate of a dedicated GBR bearer. It was found that its presence did not affect QoE in several iterations of the QoS parameter set through both simple elimination and feature importance extraction from Random Forest and Gradient Boosted Trees. Some media-layer parameters were also considered early on, but were dropped to maintain the non-intrusive approach.
Certain simulation parameters were configured for increased realism. The propagation loss model was configured to follow the Okumura-Hata propagation loss model in an urban mode. The Okumura-Hata model is considered one of the most accurate models for path loss prediction in urban areas [76].
The creation of the simulation was a particularly challenging part of this work. Arguably NS-3 has a very steep learning curve, and together with a lack of a lot of detailed documentation or community resources getting the simulation right was a difficult challenge. The simulation went through a lot of iterations before it was finally in a state where it performed as planned.

3.3 Video QoE Evaluation

To use QoE scores as labels for the machine learning models and to compare their effectiveness, it is necessary to obtain QoE scores for all of the videos that were transmitted in the simulation. QoE can be evaluated subjectively and objectively. Subjective evaluations combined with the LTE simulation would be very restrictive in terms of data set size. It is extremely challenging to survey enough people to get a comprehensive data set, let alone while doing that in a controlled environment as employing remote crowdsourcing can add further impairments to the videos and therefore may produce erroneous results. The studies that have used surveys to get QoE scores usually end up with less than 50 subjects, most often with just over 20 [37, 38, 39]. This number of subjects is not realistic for a data set with over 3000 videos, therefore it was decided to go with objective QoE evaluation.

There are many tools for objective QoE evaluation. The four most popular ones in literature are PSNR, SSIM and VQM, as well as VMAF which is a newer tool that is steadily gaining popularity. These tools were described in Chapter 2, where it was shown based on related works that PSNR is considered to be the least accurate technique, while VQM and VMAF have relatively high correlation with subjective QoE scores.

3.3.1 Initial QoE Evaluation Approach

Early on during the implementation the Evalvid [77] tool was used to evaluate video QoE. Though this tool has the same name as the NS-3 module used in the simulation, they are not formally connected or distributed together, with the NS-3 module being developed much later than the original Evalvid tool. The original Evalvid tool is a simple command line tool which compares the original video to the transmitted video and provides data on lost frames, delays and jitter as well as the MOS. It also provides a few other capabilities which are described later in the chapter. This tool was initially chosen due to its seeming prevalence in research and due to its ease of use.

The MOS provided from the Evalvid tool is based on peak signal-to-noise Ratio (PSNR) and its direct mapping to MOS specified in ITU J.144 [16]. This mapping is presented in the table below

PSNR(dB)	MOS
≥ 37	5 (Excellent)
31 - 37	4 (Good)
25-31	3 (Fair)
20-25	2 (Poor)
≤ 20	1 (Bad)

The Evalvid tool was a good starting point, however after some further consideration it was decided to move on to a more accurate tool. PSNR works by simply performing a byte-to-byte comparison of the original video with the target video and is measured in decibels, with a higher value corresponding to a higher degree of similarity between the two videos. No viewer bias is taken into account when calculating PSNR. While this might be enough for some applications that exclusively deal with objective measures of video quality, PSNR has been shown to be significantly inaccurate when compared to subjective QoE scores [78, 79]. Therefore, it was decided that a different methodology for QoE evaluation was needed, and VQM was eventually chosen due to its accuracy.

3.3.2 QoE Evaluation Process

To send a video over the network it is first necessary to pick the videos to send. The videos chosen were videos of animals from the Consumer Digital Video Library [80]. The videos are in Common Intermediate Format (CIF), which signifies that the size of the video is 352x288 pixels. The frame rate of the videos are 30 fps, the lengths are around 20 seconds, they are of excellent quality and have no scene cuts or audio.
While CIF is not the maximum possible resolution for mobile video, it is reasonable enough and is commonly used in mobile video evaluation. A higher resolution would also negatively impact the simulation execution times as well as the QoE evaluation times which would have resulted in a much smaller data set. It was important for the original video to be of excellent quality to make sure the resulting QoE score would be adequately computed since VQM requires a reference original video.
To transmit the video over the wireless network it was necessary to create the trace file of the video due to the requirements of sending it over the NS-3 simulated network. The frame trace file contains four columns: frame index, frame type, frame size, number of segments in the case of frame segmentation and the time the frame was generated by the encoder. An excerpt from the trace file is presented in Table 3.2

MPEG4 [81] is one of the codecs supported by the Evalvid NS-3 module, so it was chosen

Frame Index	Frame Type	Frame Size	Segment Number	Time Sent
1	H	15535	16	0.019
2	P	12305	13	0.036
3	P	13548	14	0.081
4	P	13111	13	0.115
5	P	14077	14	0.134
6	P	15105	15	0.166
7	P	14250	14	0.216

Table 3.2: Video Trace Frame File

for trace file generation. The videos, originally in AVI format, were converted to MP4 with an MPEG4 codec by the FFMPEG command line tool [82], saving all of the original quality parameters, like size, frame rate and bit rate.

Using the trace generation tool from the Evalvid toolset [77], the trace of the original video was obtained. Then, using the FFMPEG tool, the source MP4 video was converted to raw YUV since it is the format used for video evaluation.

When the video is transmitted, the Evalvid NS-3 module produces a receiver trace file, which consists of three columns: the time received, the frame id and the payload size. With the Evalvid toolset the transmitted video is recreated in MP4 format by reference to the original frame trace and the original MP4 video. It is then also converted to raw YUV with FFMPEG. This routine had to be performed for each simulation, so it was automated with bash scripts.

To evaluate the videos the VQM VFD tool that was described in Chapter 2 was used. NTIA VQM software is free for commercial and non-commercial use and can be downloaded from their website. The batch processing tool can only be used with a GUI where the YUV clips to be evaluated are selected, the original clip is specified and finally the type of VQM model is chosen, which for this work was VQM VFD. Through experiments it was found that the tool is unable to handle more than 500 clips, and took around five to six hours to complete. The output provided was a comma-separated file with VQM scores for every clip.

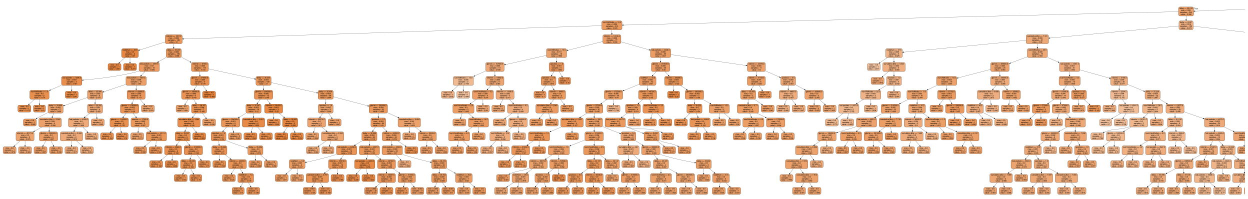

Figure 3.2: Left Half of Example Random Forest Tree

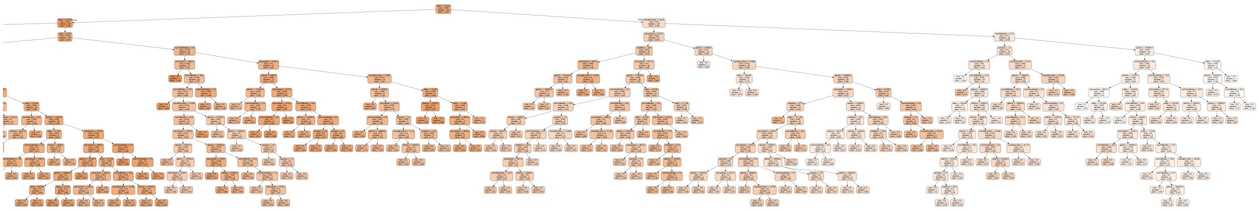

Figure 3.3: Right Half of Example Random Forest Tree

3.4 Application of Machine Learning

The four Machine Learning Algorithms picked for this work were Support Vector Machines, Random Forest, Gradient Boosted Trees and Neural Networks. The details of these algorithms are described in Chapter 2.

The first three models were implemented using the sci-kit learn Python library[83]. The QoS data and the corresponding QoE MOS were split randomly into 70% for training and 30% for testing, and the QCI classes were one-hot encoded and data was normalized. For the SVM implementation, the Support Vector Regression model was used with the RBF kernel. While classification might seem like a good fit since the MOS scale is discrete and has 5 possible values, this problem is more suited to regression algorithms since the MOS values were left fractional to keep them more granular and descriptive. The model was trained on the training data set and then tested.

For the Random Forest implementation feature scaling or one-hot encoding is not important so the data was left as is. To tune the model a grid search was performed to find the most optimal number of estimators (number of trees) and the maximum depth of a tree. An example of a tree created by the Random Forest algorithm after training on the QoS parameter data set is presented in Figure 3.2 and Figure 3.3.

A similar process was performed for Gradient Boosted trees, except that the learning rate

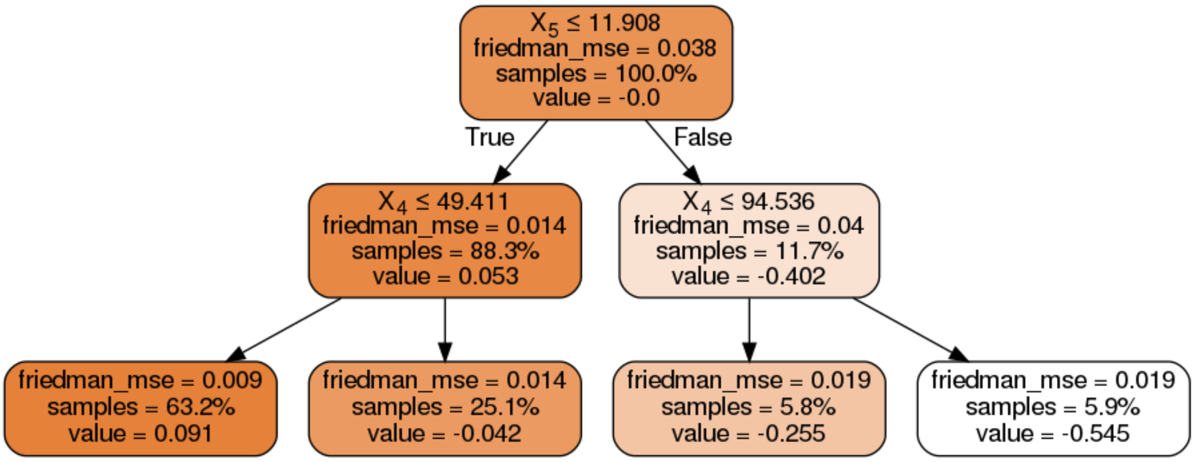

Figure 3.4: Example GBT estimator

parameter also had to be tuned. An example of a GBT estimator from training on the QoS parameter data set is presented in Figure 3.4. It should be noted that no significant boost in error minimization was achieved through parameter tuning, but neither was it expected. It was mainly performed to make the model perform as well as possible, even if it was just a gain in a couple of percent of accuracy.

For the Neural Network implementation Tensorflow [84] and Keras API [85] were used. All of the data was first normalized to aid the training process. Then, with the help of the Keras API a Neural Network with four layers was built, including two densely connected hidden layers using the Rectified Linear Unit (ReLU) activation function, which is a simple and computationally fast activation function [86], and an output layer with one node. The Mean Squared Error loss function was specified, which is commonly used for regression problems. The model was then trained with the normalized training data set, which only took a few minutes since the data set is reasonably small, which resulted in a Feedforward Neural Network model ready to be tested.

3.5 Design and Implementation Summary

An LTE simulation with the NS-3 simulation framework and the LENA LTE module was created, which aims to recreate a realistic video transmission scenario. QoS parameters were collected during the simulation, and the transmitted videos were objectively evaluated for

QoE with the NTIA VQM tool. Four machine learning algorithms models, namely Support Vector Machines, Random Forest, Gradient Boosted Trees and a Feedforward Neural Network were created, tuned and trained for video QoE prediction. The results and analysis are presented in the next chapter.

4 Results and Evaluation

This chapter provides a detailed discussion of the results of QoE prediction of the four machine learning models. The importance of wireless-specific features in QoE prediction is also examined. The evaluation of the results and the work in general is also provided in this chapter.

4.1 Results

The four algorithms used for QoE prediction in this work are Support Vector Machines, Random Forest, Neural Networks and Gradient Boosted Trees.

The models were trained on the data set that contained network QoS data of video streaming collected during LTE simulations and the QoE MOS of each streamed video. All of the four models created for this work performed reasonably well on the task of Quality of Experience prediction and showed high degrees of accuracy. In Figure 4.1 the Mean Absolute Percentage Error (MAPE) scores of the models are shown. Mean Absolute Percentage Error is a metric that provides relative error and is calculated by dividing the absolute error by the target value [87]. MAPE is a metric that allows simple comparison of performance of different machine learning models. Shown in Figure 4.1 is the Mean Absolute Percentage Error subtracted from 100 to show the accuracy of the models. Only the Neural Network managed to score over 90%, however other models also performed decently, with Support Vector Machines performing the poorest with less than 85% accuracy.
In Figure 4.2 the Root Mean Square Error(RMSE) scores are shown. RMSE is the sample

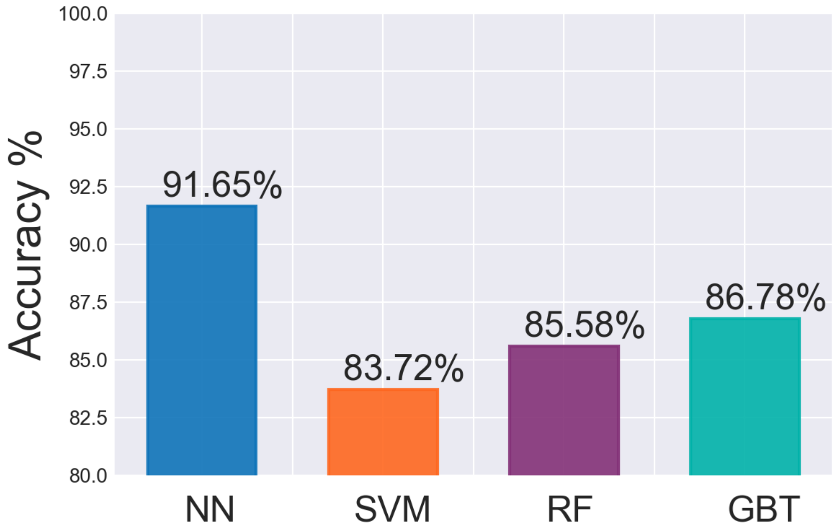

Figure 4.1: Accuracy Comparison of the four ML models

Figure 4.2: RMSE Comparison of the four ML models

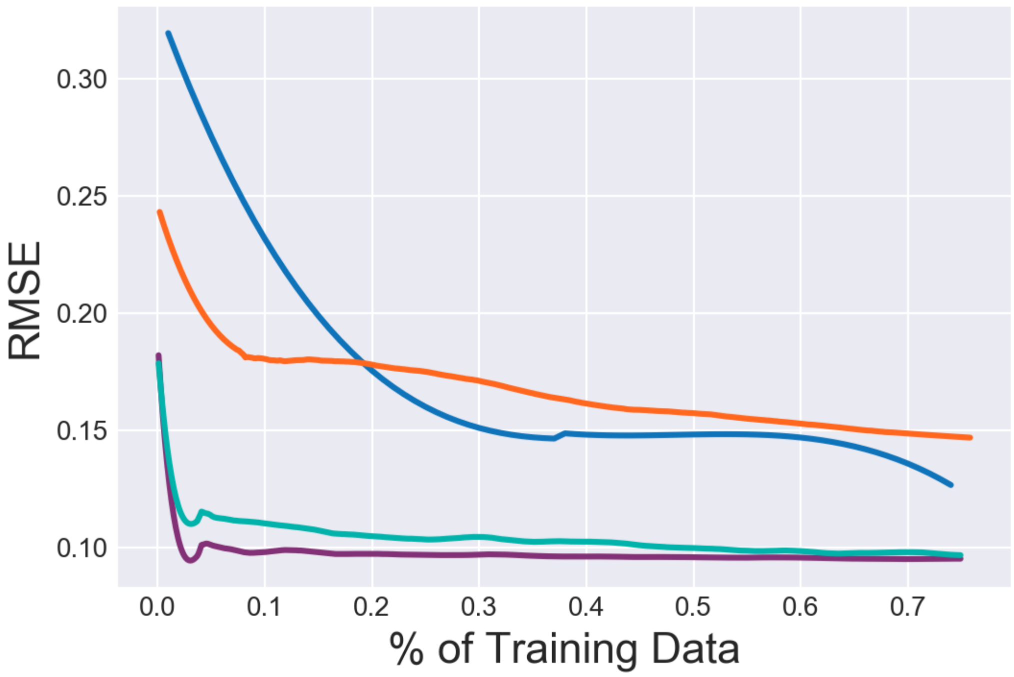

Figure 4.3: Influence of number of training samples on RMSE

standard deviation of the predicted values and target values. RMSE is measures in the same units as the label data and is very commonly used to evaluate regression model performance. The lower the RMSE, the better the model is at prediction. It shows a slightly different picture than MAPE due to the fact that RMSE punishes larger errors more severely than smaller ones. It is evident that SVM performs the worst out of the four algorithms. Random Forest and Gradient Boosted Trees achieve just under 0.1 RMSE, which is very reasonable in terms of the MOS scale. The RMSE of Neural Networks is not critical, however it is not fully desirable either.

In total 5000 simulations were run, and each model was trained on 75% of the total data set. Figure 4.3 shows the relationship between the number of training samples and Root Mean Square Error. The number of training samples has dramatically decreased error in the case of Support Vector Machines and Neural Networks, however has not affected the other two quite as much. In general, Neural Networks and SVM require larger training data sets for more accurate prediction.

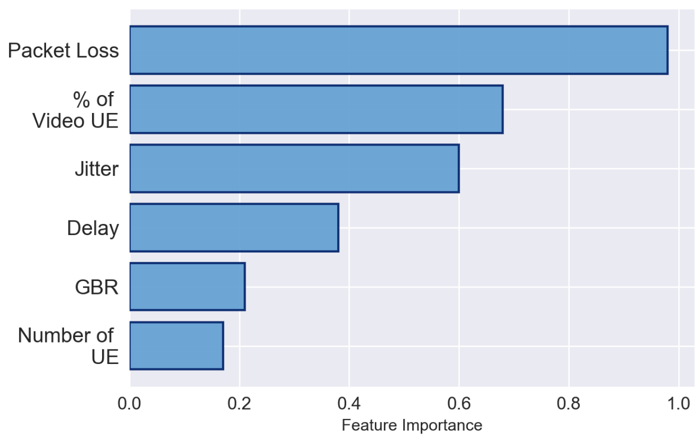

Figure 4.4: GBT Feature Importance

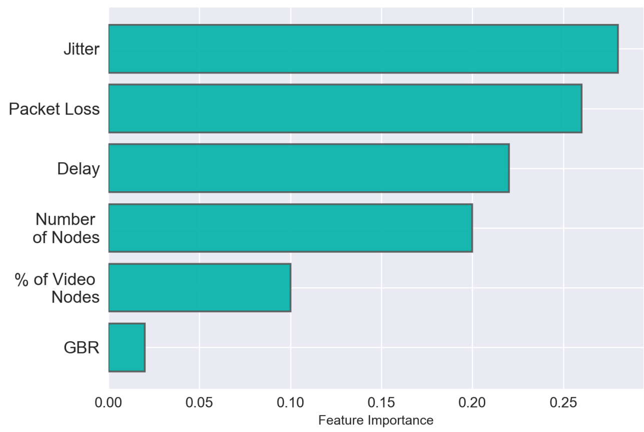

Figure 4.5: RF Feature Importance

4.2 Feature Importance

Feature Importance extraction is provided by both Random Forest and Gradient Boosted Trees. The feature importances provide a score for how influential a particular feature was when determining splits in the model. Figure 4.4 and Figure 4.5 show the feature importances of the Gradient Boosted Trees and the Random Forest models respectively.

The feature importances are quite different in the two models, which is most likely due to the different methods of split determination in the two models. Packet loss, delay and jitter seem to be influential in both models. Out of the background traffic parameters, the influence of the number of VoIP users was too negligible to even feature on the graph. Even

	Jitter	Delay	Packet Loss
Number of UE	0.95	0.90	0.91
% of Video UE	0.9	0.82	0.83
GBR	0.22	0.23	0.24

Table 4.1: Pearson Correlation of wireless-specific parameters and delay, jitter and packet loss

though VoIP does have a higher priority than video, the low ratio of background VoIP traffic must have not affected QoE of video streaming in the same cell. QCI seems to have also not been important in QoE prediction.

The total number of UE and the video streaming UE feature as having importance in both models. Particularly, in Gradient Boosted Trees, percentage of UE streaming video seems to have a lot of influence over QoE. In Random Forest, the number of UE's connected to the eNB is more influential than other wireless-specific parameters. Based on the two figures GBR seems to have had some influence in QoE prediction, but not a considerable amount. Number of UE per eNB and the percentage of UE streaming video seem to clearly be the most influential out of the wireless-specific QoS parameters.

One thing this model have not considered is the correlation between some of the parameters. In particular, delay, jitter and packet loss can be affected by the number connections and the percentage of video streams in those connection. To assess this, Pearson correlation was calculated between delay, jitter and packet loss, and number of UE, % of UE streaming video and GBR. The values are shown in Table 4.1. Number of UE and % of Video UE have quite high correlations, which would imply that there is a potential linear relationship between the values, which in turn could mean that only one set is necessary for QoE prediction.

To assess this hypothesis, all the models were re-trained on the two separate feature sets, one featuring delay, jitter and packet loss and the other all of the wireless-specific parameters. The RMSE of the results are shown in Figure 4.6.

None of the RMSE have increased dramatically from the model with all of the features. This would indicate that two sets could potentially be used for QoE prediction independently from each other.

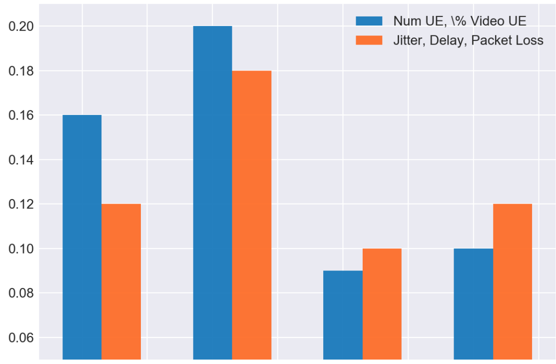

Figure 4.6: RMSE of Wireless vs Delay, Jitter, Packet Loss Parameters

4.3 Evaluation

The results presented are promising and do indicate that machine learning techniques can be used for QoE prediction. Out of the wireless-specific parameters the number of UE per eNB and the number of these UE streaming video seem to be the most influential in QoE prediction, however they are not more important than delay, jitter and packet loss. The finding that a reasonably performing model is possible without delay, jitter and packet loss parameters and using only wireless-specific parameters, in particular number of UE per eNB and the percentage of the UE per eNB streaming video, is helpful in cases where only specific types of data is available or is more easily obtained. However, high accuracy QoE prediction of video is possible without wireless-specific parameters. There is no clear winner among the machine learning algorithms used, however Random Forest and Gradient Boosted Trees have the lowest RMSE values, as well as fast training times.

The fact that LTE QoS-specific parameters did not have much effect on the performance of the QoE models is a big disappointment. GBR in particular is set by the service providers, and thus could have been effectively used in QoE adaptation .

One particular note about the simulation is that the MOS of the videos transmitted in the simulation seem to be distributed around two values as shown in Figure 4.7. It is also of note that no MOS of 1 were collected, which would be an issue if this model is presented with data that was not created by this simulation. It is also an indicator that the current LTE simulation parameters cannot cause extreme distress to a video. However, even in

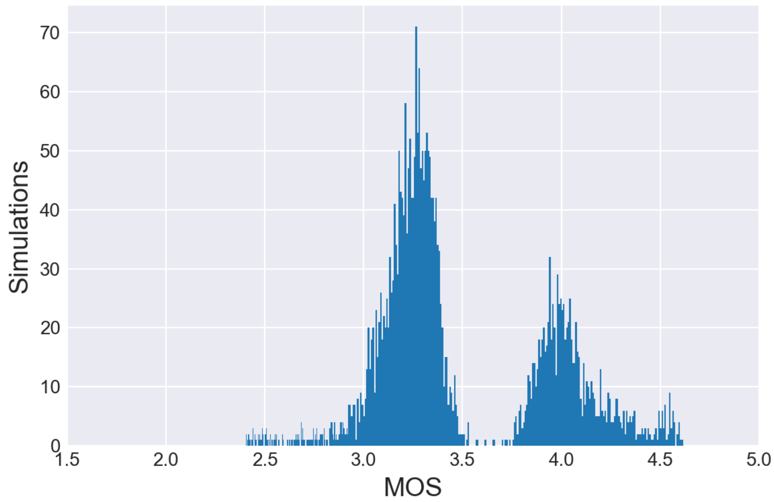

Figure 4.7: Distribution of MOS scores

subjective studies the amount of true MOS of 1 is quite low. [88] created a data set of impaired videos specifically for subjective MOS evaluation, but only 20 videos out of over 1200 were given a MOS of 1, so the absence of these scores might not be a particularly critical issue. In turn, the absence of many values of MOS 2 is generally slightly concerning in terms of the ability of this model to predict all QoE. This indicates that more focus should have been placed on getting the edge cases when creating the simulation. However, this distribution also speaks to the non-linear relationship between QoS and QoE.

It is also important to point out that the models rely fully on an external tool to objectively evaluate QoE. Despite VQM being widely used in research and highly regarded, the use of objective QoE scores could put the model at a disadvantage when used in real-world scenarios, and subjective QoE scores are needed to verify the accuracy of the models presented in this work.

Finally, in terms of the work in general a lot of time was spent on creating the LTE simulation. While this is helpful for model performance in terms of feature quality, in hindsight using a public data set would allow for more time for a deeper exploration of machine learning algorithms, specifically for incremental and online learning.

Overall, machine learning algorithms were successfully applied to QoE prediction. It was discovered that Support Vector Machines are the least suited out of the four for QoE prediction and that Random Forest and Gradient Boosted Trees achieve the lowest RMSE scores. Wireless-specific parameters were found to affect QoE prediction, and Number of UE per eNB and % of UEs Video streaming could be used for relatively precise QoE prediction if needed, however such models do not provide an advantage over just using delay, jitter and packet loss for QoE prediction.

5 Conclusion

In this work an approach was developed that uses network Quality of Service parameters for video Quality of Experience prediction with machine learning algorithms. An LTE simulation was created to provide the means for collecting a data set of network QoS parameters of video streaming. The LTE simulation was designed to reflect a realistic scenario of video transmission, including realistic background traffic and transmission environment. The parameters collected during each simulation included QoS parameters that are typically used to describe a service, such as delay, jitter and packet loss, but also wireless-specific and LTE-specific parameters that are often neglected in related research.
The data set of video QoE scores was gathered by objectively evaluating each video transmitted in each simulation, with a tool that has high correlation with subjective QoE scores. The data together was used for training four Machine Learning models: Support Vector Machines, Random Forest, Gradient Boosted Trees and Feedforward Neural Network. After evaluating the results of these models, it was discovered that all but Support Vector Machines perform adequately and achieve relatively low error. These results can still be achieved even when the models are trained exclusively on wireless-specific parameters, thus showing that these parameters have effect on video QoE and should be considered for Quality of Experience prediction in wireless networks. In particular, the number of UE connected to an eNB and the percentage of them streaming video are two wireless-specific parameters that are especially influential in QoE prediction. However, using the wireless-specific parameters alone for QoE prediction does not provide an advantage over using delay, jitter and packet loss alone

5.1 Future Work

In the current implementation, the LTE simulation only allows for the evaluation of UDP video streaming. It might be of interest to extend the simulation to TCP video streaming since Quality of Experience of TCP and UDP video is affected differently by QoS impairments [89], and a lot of modern mobile video is TCP-based.

The models presented in the work would benefit from subjective QoE scores. A reasonably-sized data set of QoE scores would allow for proper model accuracy verification. A large-scale study for subjective QoE evaluation of all videos in the set would be extremely beneficial for the models' accuracy and potential real-world application, however it might be particularly ambitious and potentially unachievable due to the number of videos in the data set.

Extending the models to be used for online QoE prediction and monitoring in its present state could be possible if the pre-trained models are used to predict QoE. However, this approach would not be considered fully online since it would not be learning real-time and would be reliant on data it was initially trained on for QoE prediction. It would be of interest to explore the use of incremental learning models, such as Incremental Support Vector Machines [90] or Gaussian Process regression [91] for more insight on in-service deployment.

Bibliography

[1] Ioannis Katsavounidis Anush Moorthy Megha Manohara Zhi Li, Anne Aaronet. "Toward a practical perceptual video quality metric". https://medium.com/netflix-techblog/toward-a-practical-perceptual-video-quality-metric-653f208b9652, 2016.

[2] 3GPP. "UTRAN Long Term Evolution (LTE) and 3GPP System Architecture Evolution (SAE)", 2006.

[3] A Bradai, Tinku Rasheed, Toufik Ahmed, and Kamal Singh. "Cellular Software Defined Network – a Framework". *IEEE Communications Magazine*, 53, 06 2015. doi: 10.1109/MCOM.2015.7120043.

[4] "Quality of Service (QoS) in LTE ". www.simpletechpost.com/2013/01/quality-of-service-qos-in-lte.html, 2013. [Last accessed 02-April-2019].

[5] Jeremy Jordan. "Support Vector Machines". https://www.jeremyjordan.me/support-vector-machines/, 2017. Last Accessed: 2019-04-08.

[6] Eric Kim. "Everything you wanted to know about the kernel trick (but were too afraid to ask)". http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html. Last Accessed: 2019-04-08.

[7] Niklas Donges. "The Random Forest algorithm". https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd, 2018. Last Accessed: 2019-04-08.

[8] Yash Upadhyay. "Introduction to FeedForward Neural Networks". https://towardsdatascience.com/feed-forward-neural-networks-c503faa46620, 2018. Last Accessed: 2019-04-08.

[9] Nicola Baldo. "The NS-3 LTE module by the LENA project", 2011.

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

[11] Cisco. "Cisco visual networking index: Global mobile data traffic forecast update, 2017–2022", 2017.

[12] Ericsson. "Ericsson mobility report". White paper, 2018. URL https://www.ericsson.com/assets/local/mobility-report/documents/2018/emr-interim-feb-2018.pdf.

[13] Akamai. "Maximizing audience engagement: How online video performance impacts viewer behavior". White paper, 2015. URL https://www.akamai.com/us/en/multimedia/documents/white-paper/maximizing-audience-engagement-white-paper.pdf.

[14] Akira Takahashi, David Hands, and Vincent Barriac. "Standardization activities in the ITU for a QoE assessment of IPTV". *IEEE Communications Magazine*, 46(2):78–84, 2008.

[15] Robert C Streijl, Stefan Winkler, and David S Hands. "Mean opinion score (MOS) revisited: methods and applications, limitations and alternatives". *Multimedia Systems*, 22(2):213–227, 2016.

[16] ITUT Rec. J. 144:"Objective perceptual video quality measurement techniques for

digital cable television in the presence of a full reference". *International Telecommunication Union, Telecommunication standardization sector*, 2004.

[17] Zhou Wang, Alan C Bovik, and Ligang Lu. "Why is image quality assessment so difficult?". In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (CASSP*, pages IV–3313. IEEE, 2002.

[18] Quan Huynh-Thu and Mohammed Ghanbari. "Scope of validity of psnr in image/video quality assessment". *Electronics letters*, 44(13):800–801, 2008.

[19] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. "Image quality assessment: from error visibility to structural similarity". *IEEE transactions on image processing*, 13(4):600–612, 2004.

[20] Zhou Wang, Ligang Lu, and Alan C Bovik. "Video quality assessment based on structural distortion measurement". *Signal processing: Image communication*, 19(2): 121–132, 2004.

[21] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. "Multiscale structural similarity for image quality assessment". In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

[22] Richard Dosselmann and Xue Dong Yang. "A comprehensive assessment of the structural similarity index". *Signal, Image and Video Processing*, 5(1):81–91, 2011.

[23] Jacob Søgaard, Lukáš Krasula, Muhammad Shahid, Dogancan Temel, Kjell Brunnström, and Manzoor Razaak. "Applicability of existing objective metrics of perceptual quality for adaptive video streaming". *Electronic Imaging*, 2016(13):1–7, 2016.

[24] Video Quality Experts Group et al. "Final report from the video quality experts group on the validation of objective models of video quality assessment, phase ii". *2003 VQEG*, 2003.

[25] Margaret H Pinson and Stephen Wolf. "Comparing subjective video quality testing methodologies". In *Visual Communications and Image Processing 2003*, volume 5150, pages 573–583. International Society for Optics and Photonics, 2003.

[26] Manish Narwaria, Matthieu Perreira Da Silva, and Patrick Le Callet. "HDR-VQM: An objective quality measure for high dynamic range video". *Signal Processing: Image Communication*, 35:46–60, 2015.

[27] Reza Rassool. "VMAF reproducibility: Validating a perceptual practical video quality metric". In *2017 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 1–2. IEEE, 2017.

[28] Chris Fetner and Danny Sheehan. "The Netflix HERMES Test: Quality Subtitling at Scale". *Netflix Tech Blog*, 2017.

[29] Kalpana Seshadrinathan, Rajiv Soundararajan, Alan Conrad Bovik, and Lawrence K Cormack. "Study of subjective and objective quality assessment of video". *IEEE transactions on Image Processing*, 19(6):1427–1441, 2010.

[30] Video Quality Experts Group et al. "Report on the validation of video quality models for high definition video content". *http://www. its. bldrdoc. gov/media/4212/vqeg_hdtv_final_report_version_2. 0. zip*, 2010.

[31] Anush Krishna Moorthy, Lark Kwon Choi, Alan Conrad Bovik, and Gustavo De Veciana. "video quality assessment on mobile devices: Subjective, behavioral and objective studies". *IEEE Journal of Selected Topics in Signal Processing*, 6(6):652–671, 2012.

[32] Margaret H Pinson and Stephen Wolf. "A new standardized method for objectively measuring video quality". *IEEE Transactions on broadcasting*, 50(3):312–322, 2004.

[33] Stephen Wolf and MH Pinson. "Video quality model for variable frame delay (VQM-VFD)". *National Telecommunications and Information Administration NTIA Technical Memorandum TM-11-482*, 2011.

[34] Patrick Le Callet, Christian Viard-Gaudin, Stéphane Péchard, and Emilie Caillault. "No reference and reduced reference video quality metrics for end to end qos monitoring". *IEICE transactions on communications*, 89(2):289–296, 2006.

[35] ITUT Rec. "p. 1401, methods, metrics and procedures for statistical evaluation, qualification and comparison of objective quality prediction models". *International Telecommunication Union, Geneva, Switzerland*, 2012.

[36] Athula Balachandran, Vyas Sekar, Aditya Akella, Srinivasan Seshan, Ion Stoica, and Hui Zhang. "Developing a predictive model of quality of experience for internet video". In *ACM SIGCOMM Computer Communication Review*, volume 43, pages 339–350. ACM, 2013.

[37] Weiwei Li, Petros Spachos, Mark Chignell, Alberto Leon-Garcia, Leon Zucherman, and Jie Jiang. "Understanding the relationships between performance metrics and QoE for over-the-top video". In *2016 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2016.

[38] Christos G Bampis and Alan C Bovik. "Learning to predict streaming video QoE: Distortions, rebuffering and memory". *arXiv preprint arXiv:1703.00633*, 2017.

[39] Pedro Casas and Sarah Wassermann. "Improving QoE prediction in mobile video through machine learning". In *2017 8th International Conference on the Network of the Future (NOF)*, pages 1–7. IEEE, 2017.

[40] Yun Gao, Xin Wei, Wenqin Zhuang, Ruochen Huang, and Mengwen Diao. "QoE prediction for IPTV based on BP_Adaboost Neural Networks". In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 32–37. IEEE, 2017.

[41] Asiya Khan, Lingfen Sun, and Emmanuel Ifeachor. "Content clustering based video quality prediction model for MPEG4 video streaming over wireless networks". In *2009 IEEE International Conference on Communications*, pages 1–5. IEEE, 2009.

[42] Karan Mitra, Arkady Zaslavsky, and Christer Åhlund. "Context-aware QoE modelling, measurement, and prediction in mobile computing systems". *IEEE Transactions on Mobile Computing*, 14(5):920–936, 2015.

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

[44] Hongzi Mao, Mohammad Alizadeh, Ishai Menache, and Srikanth Kandula. "Resource management with deep reinforcement learning". In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, pages 50–56. ACM, 2016.

[45] Keith Winstein and Hari Balakrishnan. "TCP ex machina: computer-generated congestion control". 2013.

[46] J Boyan and M Littman. "A distributed reinforcement learning scheme for network routing". In *Proceedings of the international workshop on applications of neural networks to telecommunications*, pages 55–61. Psychology Press, 2013.

[47] S Canale, F Delli Priscoli, S Monaco, L Palagi, and V Suraci. "A reinforcement learning approach for QoS/QoE model identification". In *2015 34th Chinese Control Conference (CCC)*, pages 2019–2023. IEEE, 2015.

[48] Tarik Begluk, Jasmina Baraković Husić, and Sabina Baraković. "machine learning-based QoE prediction for video streaming over LTE network". In *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pages 1–5. IEEE, 2018.

[49] Paulos Charonyktakis, Maria Plakia, Ioannis Tsamardinos, and Maria Papadopouli. "On user-centric modular QoE prediction for VoIP based on machine-learning algorithms". *IEEE Transactions on mobile computing*, 15(6):1443–1456, 2016.

[50] Pablo Frank and Jose Incera. "A neural network based test bed for evaluating the quality of video streams in ip networks". In *Electronics, Robotics and Automotive Mechanics Conference (CERMA'06)*, volume 1, pages 178–183. IEEE, 2006.

[51] Haiqing Du, Chang Guo, Yixi Liu, and Yong Liu. "Research on relationship between QoE and QoS based on BP neural network". In *2009 IEEE International Conference on Network Infrastructure and Digital Content*, pages 312–315. IEEE, 2009.

[52] Yuan He, Chao Wang, Hang Long, and Kan Zheng. "PNN-based QoE measuring model for video applications over LTE system". In *7th International Conference on Communications and Networking in China*, pages 58–62. IEEE, 2012.

[53] Vlado Menkovski, Adetola Oredope, Antonio Liotta, and Antonio Cuadra Sánchez. "Predicting quality of experience in multimedia streaming". In *Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia*, pages 52–59. ACM, 2009.

[54] Vlado Menkovski, Georgios Exarchakos, and Antonio Liotta. "Online QoE prediction". In *2010 Second International Workshop on Quality of Multimedia Experience (QoMEX)*, pages 118–123. IEEE, 2010.

[55] M Sajid Mushtaq, Brice Augustin, and Abdelhamid Mellouk. "Empirical study based on machine learning approach to assess the QoS/QoE correlation". In *2012 17th European Conference on Networks and Optical Communications*, pages 1–7. IEEE, 2012.

[56] Vaneet Aggarwal, Emir Halepovic, Jeffrey Pang, Shobha Venkataraman, and He Yan. "Prometheus: toward quality-of-experience estimation for mobile apps from passive network measurements". In *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, page 18. ACM, 2014.

[57] Christopher Cox. *"An introduction to LTE: LTE, LTE-advanced, SAE and 4G mobile communications"*. John Wiley & Sons, 2012.

[58] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. "A practical guide to support vector classification". 2003.

[59] Bernhard Scholkopf and Alexander J Smola. *"Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond"*. MIT press, 2001.

[60] Bernhard Scholkopf, Kah-Kay Sung, Christopher JC Burges, Federico Girosi, Partha Niyogi, Tomaso Poggio, and Vladimir Vapnik. "Comparing Support Vector Machines with gaussian kernels to radial basis function classifiers". *IEEE transactions on Signal Processing*, 45(11):2758–2765, 1997.

[61] S Rasoul Safavian and David Landgrebe. "A survey of decision tree classifier methodology". *IEEE transactions on systems, man, and cybernetics*, 21(3):660–674, 1991.

[62] Tom Dietterich. "Overfitting and undercomputing in machine learning". *ACM computing surveys*, 27(3):326–327, 1995.

[63] Andy Liaw, Matthew Wiener, et al. "Classification and regression by Random Forest". *R news*, 2(3):18–22, 2002.

[64] Jane Elith, John R Leathwick, and Trevor Hastie. "A working guide to boosted regression trees". *Journal of Animal Ecology*, 77(4):802–813, 2008.

[65] Simon Haykin. *"Neural networks: a comprehensive foundation"*. Prentice Hall PTR, 1994.

[66] Robert Hecht-Nielsen. ""theory" of the backpropagation neural network". In *Neural networks for perception*, pages 65–93. Elsevier, 1992.

[67] Ian Fogg. "The state of Wifi vs mobile network experience as 5G arrives", 2018.

[68] Klaus Wehrle, Mesut Günes, and James Gross. *"Modeling and tools for network simulation"*. Springer Science & Business Media, 2010.

[69] Johannes Lessmann, Peter Janacik, Lazar Lachev, and Dalimir Orfanus. "Comparative study of wireless network simulators". In *Seventh International Conference on Networking (icn 2008)*, pages 517–523. IEEE, 2008.

[70] George F Riley and Thomas R Henderson. "the NS-3 network simulator". In *Modeling and tools for network simulation*, pages 15–34. Springer, 2010.

[71] Thomas R Henderson, Mathieu Lacage, George F Riley, Craig Dowell, and Joseph Kopena. "Network simulations with the NS-3 simulator". *SIGCOMM demonstration*, 14(14):527, 2008.

[72] GERCOM. "NS-3 Evalvid module by GERCOM", 2013. URL `https://gitlab.com/gercom/evalvid-ns3`.

[73] Frank HP Fitzek and Martin Reisslein. "MPEG-4 and H.263 video traces for network performance evaluation". *IEEE network*, 15(6):40–54, 2001.

[74] Hassan Hassan, Jean Marie García, and Brun. "Bandwidth allocation and session scheduling using sip". *Journal of Communications*, 1, 08 2006.

[75] Sanjay Ramroop. "A diffserv model for the ns-3 simulator". *En ligne]. http://www.eng.uwi.tt/depts/elec/staff/rvadams/sramroop/index.htm*, 2011.

[76] Yuvraj Singh. "Comparison of okumura, hata and cost-231 models on the basis of path loss and signal strength". *International journal of computer applications*, 59(11), 2012.

[77] Jirka Klaue, Berthold Rathke, and Adam Wolisz. "Evalvid–a framework for video transmission and quality evaluation". In *International conference on modelling techniques and tools for computer performance evaluation*, pages 255–272. Springer, 2003.

[78] Olivia Nemethova, Michal Ries, Eduard Siffel, and Markus Rupp. "Quality assessment for H. 264 coded low-rate and low-resolution video sequences". In *Communications, Internet, and Information Technology*, pages 508–512, 2004.

[79] D Hernando, JE López de Vergara, D Madrigal, and F Mata. "Evaluating quality of experience in iptv services using mpeg frame loss rate". In *2013 International Conference on Smart Communications in Network Technologies (SaCoNeT)*, volume 3, pages 1–5. IEEE, 2013.

[80] Margaret H Pinson. "The consumer digital video library [best of the web]". *IEEE Signal Processing Magazine*, 30(4):172–174, 2013.

[81] Iso-iec/jtc1/sc29/wg11.iso/iec 14496: "information technology", 2001.

[82] Suramya Tomar. "Converting video formats with FFMPEG". *Linux J.*, 2006(146):10–, June 2006. ISSN 1075-3583.

[83] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. "scikit-learn: Machine learning in python". *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[84] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. "TensorFlow: Large-scale machine learning on heterogeneous systems", 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

[85] François Chollet et al. "Keras". `https://keras.io`, 2015.

[86] Kazuyuki Hara, Daisuke Saito, and Hayaru Shouno. "Analysis of function of rectified linear unit used in deep learning". In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2015.

[87] Arnaud De Myttenaere, Boris Golden, Bénédicte Le Grand, and Fabrice Rossi. "Mean absolute percentage error for regression models". *Neurocomputing*, 192:38–48, 2016.

[88] Vlad Hosu, Franz Hahn, Mohsen Jenadeleh, Hanhe Lin, Hui Men, Tamás Szirányi, Shujun Li, and Dietmar Saupe. "The Konstanz natural video database", 2017. URL `http://database.mmsp-kn.de`.

[89] Tobias Hoßfeld, Raimund Schatz, Thomas Zinner, Michael Seufert, and Phuoc Tran-Gia. "Transport protocol influences on youtube videostreaming QoE". *University of Würzburg, Institute of Computer Science, Tech. Rep*, 2011.

[90] Yi-Min Wen and Bao-Liang Lu. "Incremental learning of support vector machines by classifier combining". In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 904–911. Springer, 2007.

[91] Arjan Gijsberts and Giorgio Metta. "Real-time model learning using incremental sparse spectrum gaussian process regression". *Neural Networks*, 41:59–69, 2013.