Estimating QoE from encrypted video conferencing traffic

Michael Sidorov
School of Electrical and
Computer Engineering
Ben Gurion University of the Negev
Be'er Sheba, Israel
sidorov@post.bgu.ac.il

Raz, Birman
School of Electrical and
Computer Engineering
Ben Gurion University of the Negev
Be'er Sheba, Israel
birmanr@post.bgu.ac.il

Ofer Hadar
Senior Member, IEEE
School of Electrical and
Computer Engineering
Ben Gurion University of the Negev
Be'er Sheba, Israel
hadar@bgu.ac.il

Amit Dvir
Center for Cyber Technology
Department of computer Science
Ariel University
Israel
amitdv@g.ariel.ac.il

June 3, 2024

Abstract

Traffic encryption has become commonplace in communication over the internet, due to the threats which lurk there for unprotected "surfers". Though aiding security, traffic encryption renders the analytical applications that rely on clear text, and provide the means to optimize video delivery, useless. One such group of applications is the ones that try to estimate the Quality of Experience (QoE) of the users who consume video content. As the demand for Video Conferencing (VC) solutions surging over the past few years, it's becoming more than ever relevant to provide a reliable QoE assessment tool also for this domain. Many models for QoE prediction have been proposed over the years, but most of them had two main drawbacks, viz. they dealt only with the problem of streaming video and operated mainly in an unencrypted setting, i.e., where the original image is available and can serve as a reference. In this paper, we propose a new model for prediction of the QoE which can assess it in video conferencing applications. We developed a custom Deep Neural Network architecture, which can predict resolution, frames per second (FPS), and Naturalness Image Quality Evaluator (NIQE) criteria with an accuracy of 90.489%, 92.492%, and 97.643% respectively, only from features which may be easily extracted from an encrypted traffic. The code for the network may be found at `https://github.com/mchlsdrv/qoe`.

1 Introduction

As we witness a significant growth of video content consumption on the internet over the past two decades, the ability to assess the QoE of the end user is becoming critical.

As the data communication channel plays an integral role in the final video quality, Internet Service Providers (ISPs) are placed in a unique

position of being able to influence it directly, e.g., by temporarily increasing the bandwidth of a "content-hungry" network segments, or by caching the data on a Content Delivery Networks (CDNs), etc.

As most of the video content is streamed over an end-to-end encrypted channels, this procedure becoming increasingly more challenging for the ISP. While the Video Content Providers (VCP) may derive the quality of the streamed video fairly easily, e.g., by directly comparing the original image to the image which is presented to the end user, this procedure is not as straight forward to ISPs, as all they have access to is the metrics of the channel, which thought has a some indirect connection to the data streamed over it, but it is not clear how the observed metrics such as BW, latency and packet size can be related to the QoE of the end user, or even the image quality of the video.

This situation is further complected as users usually have more than a one application that requires data being transmitted over the internet. If we will add to this, the fact that in most cases data channels governed by the ISPs carry traffic which corresponds to multiple users, this problem complicates even folds more.

After the pandemic of 2019, VC application use surged [2] [7], and due to the wide adaption by many schools and organizations worldwide [3] [6] [15], QoE assessment in this domain remains a "hot" topic today.

Though the term "good QoE" may be perceived by each individual on an intuitive level, as it is plausible to suppose that each person, if asked to rate a video, would be able to do it, it is hard to come up with a clear and objective definition to what criteria made the same person decide on the QoE of some particular video or image, as this decision usually hinges on the subjective judgment of the person asked.

Nevertheless, we still may try to come up with a set of features that we call Key Performance Indicators (KPIs), that may exert influence when one evaluates a video. The most intuitive KPIs to recon on, are the technical ones (e.g., initial startup delay, number and duration of stalling events, and KPIs related to the image quality, viz. bit rate, and resolution.) Unfortunately, there are also quite a few features that have a less straightforward definition. Just to name a few of these are the context of the judgment (i.e., we require a higher quality presented on a higher quality monitor, or we would be less judgmental of an image of a lesser quality that is streamed over a mobile device, etc.), viewers attention to details, or even its emotional state, all may affect its final decision.

Criteria that try to evaluate the QoE of some videos generally may be categorized into two classes, viz. Subjective and Objective. The main difference between the two classes may be summed up as the amount of human participation in the process of scoring. While criteria that correspond to the former category require full human involvement to acquire the rating, criteria that fall into the latter category can do without it.

One of the most widely used subjective criteria of QoE is the Mean Opinion Score (MOS), which was presented by ITU-T Focus Group on IPTV [20], and represents the overall acceptability of an application (or a service), perceived by the end user. It is calculated by averaging the scores assigned by participants (usually on a scale of five points) to video samples, which are designed to simulate a certain quality.

Objective criteria for QoE evaluation may be further divided into three subcategories, viz. full reference, reduced reference, and no reference, where the main difference between the three categories is the use of the original image in the evaluation process. While criteria that may be assigned to the first category use the original image as a reference to evaluate the corrupted image, criteria in the second category try to mitigate this dependence by using only the most important and easily acquired subset of features that may be used to predict the QoE. Criteria that fall into the last category go even further, i.e., they are completely independent of the original image.

2 Related Work

There are three main lines of previous work on predicting QoE from encrypted traffic, viz. methods based on statistical analysis, Machine Learning (ML) algorithms, which mainly use decision trees or their variations, and Deep Learning (DL) algorithms employing Artificial Neural Networks (ANN).

A model for YouTube quality estimation streamed over HTTPS with the Dynamic Adaptive Streaming over HTTP (DASH) protocol was proposed in [5]. They showed that an accurate classification of the image quality is achievable just by looking into the data bursts picked up from the channel during the YouTube session, which they were able to classify into three categories (viz. 360p, 480p, and 720p) with an accuracy of 97%.

In [9] authors extended the work of [5], where they again analyzed YouTube sessions with DASH protocol but tried to predict resolution on a higher granularity, i.e., having six states (144p, 240p, 360p, 480p, 720p and 1080p) instead of just three (480p, 720p and 1080p), with two other metrics, which are specific to DASH sessions (the buffer, and video states.) They trained a decision tree algorithm, and thought scoring 92.0% and 84.2% for the last two metrics respectively, their performance for the resolution was significantly lower than of [5], with just 66% accuracy. Even when the problem was reduced to a binary prediction, with just two categories of small (144p, 240p, 360p) and large (480p, 720p, and 1080p), the accuracy of their model did not exceed 91%.

Mazhar and Shafiq in [12] presented a method for QoE prediction of a video transmitted over an encrypted channel using a decision tree algorithm to predict rebuffering events, video quality, and startup delay, features which all correlate with the QoE of users. Although achieving classification accuracy of 90% for HTTPS and 85% for QUIC traffic, their method has two significant drawbacks, viz. its low granularity, as for each parameter they use only a binary class prediction, i.e., "at least one" rebuffering event in a video session, "high" or "low" video quality, and whether the video started before or after some predefined time threshold, and inability to be used in an online setting, as their model predicts the QoE only retrospectively.

Authors of [16] proposed a YouQ, a framework for live QoE estimation. They used a simple model that combine proportions of the time spent in a high resolution (viz. 1080p or 720p) and the duration of the stalling events during the video playback time. They evaluated their method against the MOS of human participants. In their work, authors have conducted two lines of experiments using a binary label (viz. "high" or "low") in the first and achieved 91% accuracy that dropped to 84% after the addition of just one extra class (viz. "medium"), which highlight the weakness of their approach.

Authors in [1] tried to improve on the results from [12] by extending its predictive granularity. In their experiments, they trained a random forest model on features from layers of the IP stack data. Their dataset included sessions of four popular video streaming platforms, viz. YouTube, Amazon, Netflix, and Twitch. Though the proposed method predicted video resolution with an accuracy of 93%, it could not generalize well to other datasets.

One of the first studies investigating the QoE in VC applications was performed by [2]. The authors studied the influence of the movement in the presented frame on the QoE of the end user. They conducted 700 controlled VC sessions on three popular VC platforms (viz. Zoom, Webex, and Google Meet), in which they presented videos with high and low movement. They used standard metrics for image quality evaluation, viz. PSNR (Peak-to-Noise Ratio), SSIM (Structural Similarity Index), and VIFp (Pixel Visual Information Fidelity) to infer the QoE of the user. The main limitations of this work, as pointed out by the authors themselves, were the lack of generality in their experiments, i.e., they used specific settings that may not always represent the general use case. In addition, they only performed their experiments for small-sized conferencing sessions (less than 11 participants), which may not be projected on medium and large conferencing sessions directly.

Yet another study in [11] considered the VC applications, but this time investigated differences in the minimal requirements to avoid a decrease in the QoE. They performed controlled experiments in which they varied the bandwidth of the down and the uplink separately, and other modalities such as the number of participants and the viewing mode (i.e., single image of the speaker or a gallery), comparing the behavior of three popular VC applications, viz. Zoom, Google Meet, and Teams. Interestingly, the authors showed that viewing mode may reduce uplink utilization through the reduction in the resolution of the displayed image. One possible limitation of this work is its strong association with geographical location, as their entire dataset originated from the same University.

As Zoom uses a proprietary header format that makes feature extraction challenging, a study in [13] provided a method to extract packet-level features such as media bit-rate, frame size, frame rate, latency of the video stream, and jitter. In their work, they outline the problem of retransmission identification and propose a *frame delay* metric, which they define as the time between the first packet of a frame and the end of frame transmission, which should indicate if a high number of retransmissions occur during the session.

3 Data

We evaluated the performance of our model on a custom dataset that imitated Zoom sessions under various channel bandwidths (changed with the NetLimitter application). It included network channel data pickup from 720 controlled Zoom sessions, where the features in table 1 were collected and averaged over one second to decrease the noise.

3.1 Features

Zoom provides several metrics available for the end user through its' Software Developer Kit (SDK), which may be automatically extracted from the session stream in real time. For our dataset we captured all them, as we wanted to explore their predictive ability of the QoE. All the features in our dataset are listed in table 1.

3.2 The QoE Metric

In our experiments we used the Naturallness Image Quality Evaluator (NIQE) metric proposed [14].

3.3 NIQE Metric

Due to the challenges of recording Zoom video sessions with FR and in order to make the dataset more accommodating to future expansions (which do not necessitate complicate setups and saving of a reference for each recorded video clip), we researched the No Reference (NR) QoE metric alternatives. A promising NR metric that has emerged is the Naturalness Image Quality Evaluator (NIQE) [14], which measures the distance of the frame from naturalness, therefore, a smaller score indicates better perceptual quality. NIQE measures the distance between the Natural Scene Statistics (NSS)-based features calculated from an image to the features obtained from an image database used to train the model. The features are modeled as multidimensional Gaussian distributions. We calculated NIQE as a function of bandwidth, resolution and FPS. The corresponding graphs are depicted in Figure 8. As can be seen in Figure 8 (b) and (c), the NIQE is consistent with available bandwidth and video frame resolution respectively, such that the larger the bandwidth or the resolution, the smaller the NIQE, thus indicating better quality. We can observe a range of resolutions for which the NIQE value remains constant, indicating that the image naturalness metric at these resolutions is sufficiently good despite the change. An interesting observation is depicted in Figure 8 (a), indicating a change of NIQE with FPS, whereas the metric is measured per single frame and is therefore spatial by nature. The temporal resolution of the video frames has an impact on the spatial quality. This is due to the compression algorithm that utilizes quantized residuals which are determined by block prediction accuracy, which is in turn impacted by the magni-

Table 1: Features extracted from the encrypted Zoom traffic for the dataset

Feature	Description
Bandwidth	The number of bits per second allowed to be transmitted on the channel
Resolution	The number of pixels in the width and the length of the presented image
FPS	The number of frames per second in the playback
Latency	Initial time after which the video starts
Jitter	Statistical variance of the Real-time Transport Protocol (RTP)[21] data packet in time interval
PPS	Number of transmitted packets in each second
Destination port	The 2^{nd} layer port on the remote host where the transmitted data is directed to
Source port	The 2^{nd} layer port on the local host through which the data is transmitted
Average time between packets	The time difference between the reception of two consecutive packets
Packet length	Number of bytes in each transmitted packet

tude of the Motion Vectors (MV) used for Inter-prediction. The larger the movement between consecutive frames, the larger the MV magnitude and the less accurate is the prediction, thus increasing the value of the temporary predicted block residuals and reducing the quality due to their quantization.

3.4 Exploratory Data Analysis (EDA)

First, we performed a simple Pearson correlation [18] of the extracted features, which is listed in table 2. This analysis highlighted the features which influence the target (i.e., the NIQE), the most. Pearson correlation is a measure of the influence which some feature X has on its' counterpart Y, i.e., how precisely can we anticipate a change (positive or negative) in Y by observing X, and is defined as

$$\rho_{X,Y} = \frac{Cov[X,Y]}{\sigma(X)\sigma(Y)} \tag{1}$$

This measure is located in the range [-1, 1], where 1 points to a perfect positive correlation, i.e., increase in one feature brings increase in the other, while -1 signifies a total reverse correlation, accordingly, which may be seen in Table 2. Features with correlation closer to ± 1 are the most interesting, as it indicates that this feature, or it reverse, has strong effect on the predicted variable. Among the strongest predictive values we see *bandwidth* (-0.78), *resolution* (-0.73), *FPS* (-0.76), *average time between packets* (0.77), and surprisingly, also *packet length* (-0.86).

As for the first three, its is quite intuitive that increase in their value will decrease the NIQE, as higher bandwidth leads directly to better resolution and fps, which in its' turn improves the perceived quality of the video.

The *average time between packets*, which has a strong positive correlation with NIQE, i.e., increase in this feature leads to worse QoE of the observer (as a high NIQE means worse QoE than a low one). This is also intuitively clear, as packages which arive at a lower rate may indicate a lower FPS.

The high negative correlation of the *packet length* though is less intuitive, but yet has an explanation, viz. as we increase the bandwidth, video compressor has the ability to place more data in each

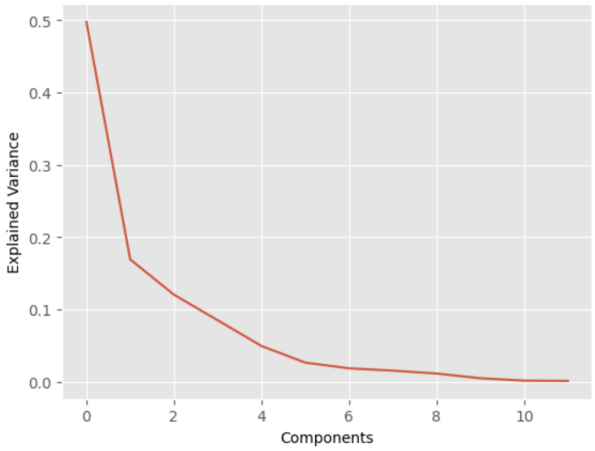

Figure 1: Primary component analysis (PCA) on the 12 features in our dataset

packet, increasing the FPS, and by this increasing also the QoE of the viewer.

In addition, to reduce the dimensionality of our dataset even further, we performed the the Primary Component Analysis (PCA) [19] algorithm, which may be seen as a method to project the data to a lower dimensional subspace, but one that still contain most of the information from the original space. The results of this procedure are presented in fig. 1, and in the corresponding table 3. From this analysis we conclude that the by transforming the features with the corresponding loadings, we can use as few as 5 features instead of the 9 original and still retain 99% of the information present in the data.

4 Zoom Video Conferencing Behaviour

From the captured dataset we have infered the behaviour of the Zoom application, as a function of different channel conditions. We chose as the limittting parameter the bandwidth of the channel, as it is the easy to control, and also the easiest to interpret.

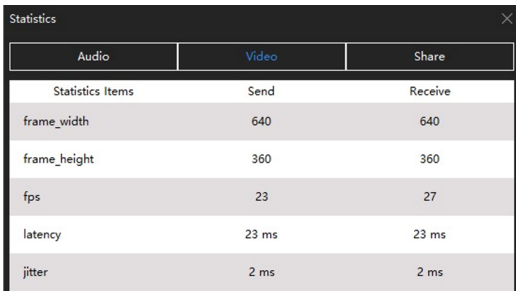

Figure 2: Zoom quality of experience extracted from an application

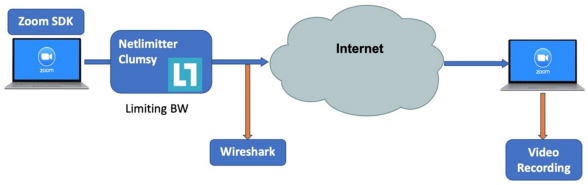

Figure 3: Setup used to capture Zoom traffic features and labels

4.1 Impact of reduced available bandwidth

We shall refer to 'transition' as the period during which the Zoom application adjusts its video transmission parameters when a degradation is introduced to the channel quality, such as a drop of bandwidth for example. We discovered that during transition, Zoom goes through the following cycle: (1) reducing video spatial resolution; (2) reducing frames per second (fps); and (3) restoring spatial resolution to original with lower fps. We observed that such a transition can typically take up to 10 sec, at the end of which the video spatial resolution is restored to the original value while a lower fps is used to compensate for the reduced available bandwidth. This behavior makes sense, given that in typical video conferencing scenario there are only small spatial movements and therefore users' perceived quality will be impacted more by the spatial resolution than by the temporal resolution. The charts of Figure 4 illustrate the steady state variation of fps and latency as a function of available

Table 2: Pearson correlation of different features in the dataset. In bold script are the features with the largest positive or negative correlation with the explained feature, i.e., NIQE

	BW	NIQE	Resolution	FPS	Latency	Jitter	PPS	Destination Port	Source Port	Average time between packets	Packet Length
BW	1.0	**-0.777**	0.586	0.656	-0.274	-0.383	0.946	-0.032	0.056	-0.931	0.873
NIQE	-0.777	1.0	-0.728	-0.758	0.391	0.486	-0.666	0.067	-0.088	0.770	-0.860
Resolution	0.586	**-0.728**	1.0	0.765	-0.417	-0.458	0.493	-0.111	0.132	-0.593	0.719
FPS	0.656	**-0.758**	0.765	1.0	-0.478	-0.495	0.591	-0.071	0.094	-0.679	0.693
Latency	-0.274	0.391	-0.417	-0.478	1.0	-0.798	-0.231	0.042	-0.061	0.284	-0.359
Jitter	-0.383	0.486	-0.458	-0.495	-0.798	1.0	-0.328	0.018	-0.040	0.377	-0.467
PPS	0.946	**-0.666**	0.493	0.591	-0.231	-0.328	1.0	0.022	0.004	-0.933	0.721
Destination Port	-0.032	0.067	-0.111	-0.071	0.042	0.018	0.022	1.0	-0.985	0.012	-0.074
Source Port	0.056	-0.088	0.132	0.094	-0.061	-0.040	0.004	-0.985	1.0	-0.044	0.106
Average time between packets	-0.931	**0.770**	-0.593	-0.679	0.284	0.377	-0.933	0.012	-0.044	1.0	-0.783
Packet length	0.873	**-0.860**	0.719	0.693	-0.359	-0.467	0.721	-0.074	0.106	-0.783	1.0

Table 3: PCA loadings and the corresponding accumulative variance summation for each PC, for features with the correlation $|\rho| > 0.5$, which presented in table 2

	PC0	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Bandwidth	-0.402	0.277	0.212	0.116	0.012	0.126	0.314	0.767
Resolution	-0.346	-0.107	-0.667	0.201	-0.453	-0.411	0.084	0.051
FPS	-0.368	-0.097	-0.432	-0.615	0.495	0.201	0.071	-0.016
Latency	0.236	0.636	-0.240	0.189	0.504	-0.439	0.002	-0.005
Jitter	0.273	0.562	-0.305	-0.239	-0.455	0.502	-0.004	-0.014
PPS	-0.375	0.321	0.350	-0.215	-0.176	-0.215	0.477	-0.533
Avg. time between packets	0.397	-0.263	-0.197	0.193	0.125	0.171	0.807	-0.032
Packets length	-0.394	0.102	-0.113	0.626	0.205	0.504	-0.102	-0.351
Cumulative variance	0.648	0.827	0.907	0.944	0.967	0.990	0.998	1.

bandwidth (measured after waiting for the transition period to pass and the spatial resolution to be restored). They were taken from a single session; however, they represent a consistent behavior that we have observed in multiple similar sessions. A dramatic drop from 25 fps to less than 10 fps is observed when the available bandwidth drops from 120 kbps to 60 kbps.

4.2 Impact of reduced bandwidth on Quantization Parameter (Qp)

Assuming that Qp may be used by Zoom as a dominant parameter to adapt to changing channel bandwidth conditions, and therefore may be used as a good indicator for QoE, we have explored the change of Qp when reducing the available bandwidth. We observed that the Qp remains unchanged for I-frames and is slightly decreased with reduced bandwidth for P-Frames. This indicates that the Zoom application tries to compensate for the reduced bandwidth, which is accommodated by reduced fps, by adjusting to finer quantization of the P-Frames, however, the change is small, and we consider it too hard to estimate from encrypted traffic. Therefore, it is less valuable as QoE predictor. The change of Qp with the reduced bandwidth is illustrated in Figure 5.

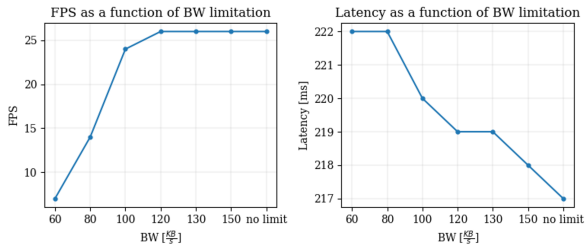

Figure 4: Zoom adaptation patterns to dropping bandwidth

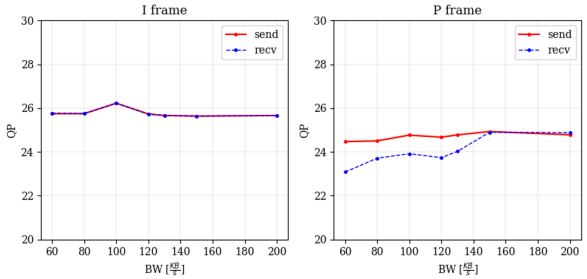

Figure 5: Zoom Qp change as a function of dropping bandwidth

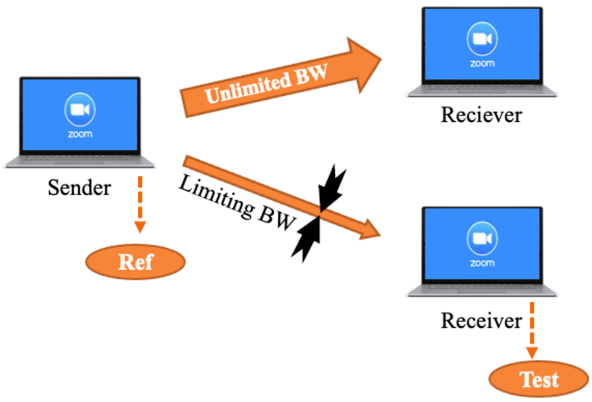

Figure 6: VMAF full reference Zoom testing setup

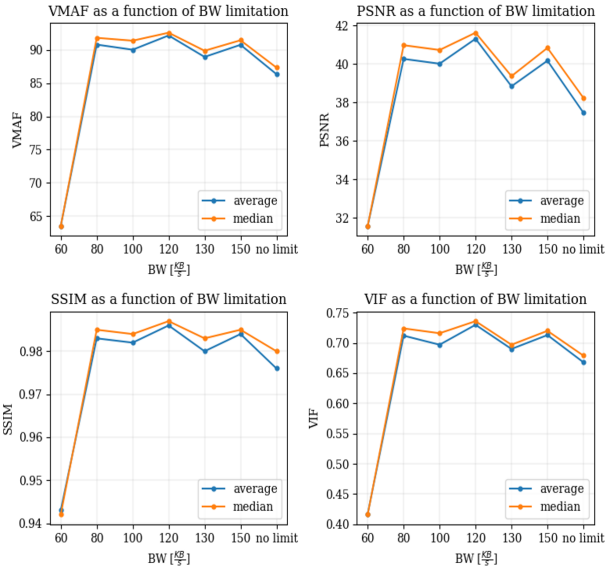

Figure 7: Measured Zoom quality metrics as a function of dropping bandwidth

5 Results

Decision trees emerge as the most accurate ML algorithm for classifying encrypted VoD traffic [16] [4] [17]. We used a Decision Tree model to predict video spatial resolution. The different spatial resolution labels that have been observed are: 1280x720, 1120x630, 960x540, 800x450, 640x360, 480x270, and 320x180. We used the Gini [8] index as a classification criterion. It measures the split quality - let the data at node m be represented by Q_m with N_m samples, let the target be a classification outcome with possible values $0, 1, \ldots, K-1$, then for node m, let the term of eq. (2) be the proportion of class k observations in node m. If m is a terminal node, predicted probability for this region is set to p_{mk}. And the measure of impurity is represented by eq. (3). We used best split as the splitter strategy at each node.

$$p_{mk} = \frac{1}{N_m} \sum_{y \in Q_m} I(y = k) \qquad (2)$$

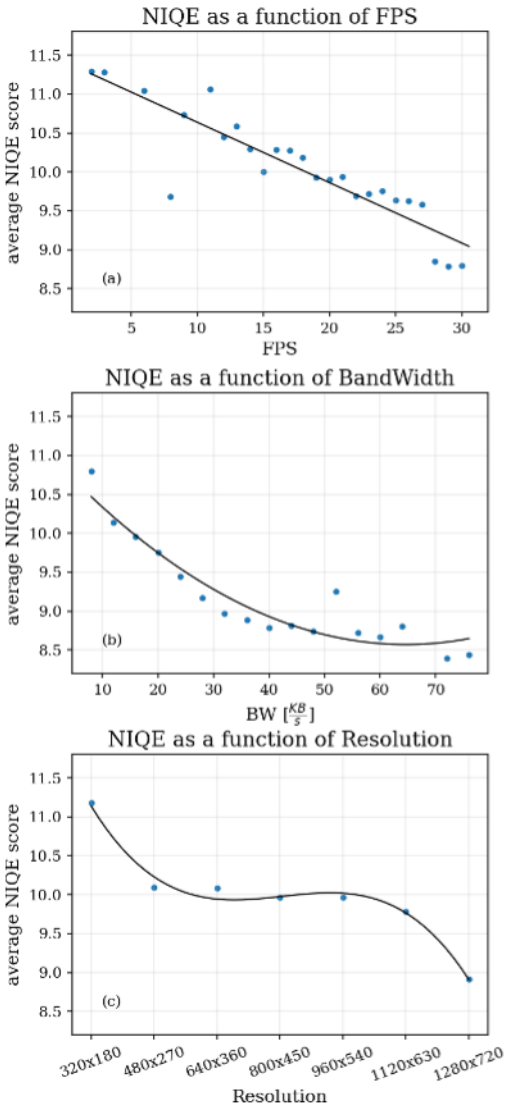

Figure 8: Measured Zoom NIQE quality metric as a function of fps, bandwidth, and resolution

$$H(Q_m) = \sum_k p_{mk}(1 - p_{mk}) \tag{3}$$

We further calculated Permutation Feature Importance, to understand which features are more dominant for the QoE prediction. Permutation Feature Importance provides a score for each feature, whereas the highest the score, the more relevant the feature is for predicting the output label. The permutation feature importance is a model agnostic approach, calculated by noticing the increase or decrease in error when we permute the values of a feature. If permuting the values cause a huge change in the error, it means the feature is important for our model. The permutation feature importance is based on an algorithm that works as follows: (1) Calculate the mean squared error between the model results and the known labels with the original values; (2) Permute the values for the features and make predictions again; (3) Calculate the mean squared error with the shuffled values; (4) Compare the difference between them; (5) Sort the differences in descending order to get features with most to least importance. The results are depicted in Figure 9 and indicate that the bandwidth (Bits Per Second) represents the most dominant feature but Packet Length and Average Time Between Packets fall not far behind. To gain better understanding of the features and their impact on the classification result, we performed a correlation check between the three most dominant features. As can be seen in Figure 10, the correlation between the three dominant features is high, so we can conclude that the most applicable feature is the bandwidth. We used a Decision Tree model for classification of the video resolution. The rest of the QoE labels – fps, NIQE, Latency and Jitter are represented by integer values, which are more suitable for regression. We compared the results of a regression algorithm to those of Artificial Neural Network and found the later to be more accurate. The accuracy of our results is provided in table 4. The network was created using the "Sequential" module of the Keras library. The used fully connected network is depicted in Figure 11. For the regression network we have used the ReLU activation

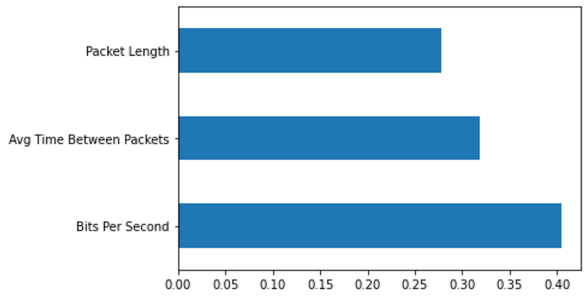

Figure 9: Feature importance calculation

function, batch size 20, and 50 Epochs. We used the 'Adam' optimizer. The accuracy was calculated by applying the trained model to each row of testing dataset (30% of the data samples) as the Absolute Percentage Error. We calculated the average of all the rows to obtain the Mean Absolute Percentage Error (MAPE). We further calculated Permutation Feature Importance, to understand which features are more dominant for the QoE prediction. Permutation Feature Importance provides a score for each feature, whereas the highest the score, the more relevant the feature is for predicting the output label. The permutation feature importance is a model agnostic approach, calculated by noticing the increase or decrease in error when we permute the values of a feature. If permuting the values cause a huge change in the error, it means the feature is important for our model. The permutation feature importance is based on an algorithm that works as follows: (1) Calculate the mean squared error between the model results and the known labels with the original values; (2) Permute the values for the features and make predictions again; (3) Calculate the mean squared error with the shuffled values; (4) Compare the difference between them; (5) Sort the differences in descending order to get features with most to least importance. The results are depicted in Figure 9 and indicate that the bandwidth (Bits Per Second) represents the most dominant feature but Packet Length and Average Time Between Packets fall not far behind.

To gain better understanding of the features and their impact on the classification result, we performed a correlation check between the three most dominant features. As can be seen in Figure 10, the correlation between the three dominant features is high, so we can conclude that the most applicable feature is the bandwidth. We used a Decision Tree model for classification of the video resolution.

Table 4: QoE prediction accuracy results

Predicted Label	Model	Accuracy
Resolution	DT	90.489%
FPS	NN	92.492%
NIQE	NN	97.643%

We trained a deep neural network with 16-64 layers and 128-256 units in each layer, as shown in fig. 11. We split the data into train and test dataset with a proportion of 90:10%, while the train data was further split each train procedure with proportion of 80:20% for the train data and validation respectively to monitor the performance in the course of the training. Parameter selection was done by a 5-fold cross validation where each time the train and test data was chosen anew in a non-overlapping manner (i.e., the tests of all the 5 folds were cohosen as non-overlapping sets). The performance of the NN are summarized in table 5, for each of the predicted labels, viz. NIQE, Resolution and FPS.

6 Ablation Study

We used the parameters $\mathbb{L}$ and $\mathbb{N}$ as the number of layers and units in each layer respectively. To find the best parameters for the NN, we divided the data in 70:20:10 ration for train, validation and test data respectively, where the train and validation datasets were determined randomly each execution, while the test set was randomly chosen, and set aside for an unbiased test evaluation.

We performed the experiments on $\mathbb{L} \in \{2, 4, 8, 16, 32\}$ and $\mathbb{U} \in \{4, 8, 16, 32, 64, 128, 256\}$, and the corresponding results are listed in table 5. For networks with $\mathbb{L} > 8$ we employed skip connections as in [10].

Train Epochs	Number of Layers	Number of Units	Initial Learning Rate	Trainable Parameters	NIQE	Resolution	Frames per Second (FPS)
10	32	128	0.002	537865	**2.746**	11.619	8.692
	64	64	0.001	275081	3.192	**10.257**	10.496
	16	64	0.001	69257	3.015	11.285	**7.541**
50	32	256	0.01	2124297	**2.357**	11.448	8.028
	32	128	0.008	537865	2.876	**9.511**	10.605
	64	256	0.002	4246025	2.738	10.236	**7.508**
100	32	128	0.01	537865	**2.395**	11.268	8.810
	64	128	0.01	1074441	2.831	**9.764**	10.272
	16	256	0.01	1063433	2.778	10.585	**7.523**

Table 5: Parameter selection with 5-fold cross validation, and Adam optimizer set to default parameters, except initial learning rate. In bold face are the best result for each of NIQE, Resolution and FPS

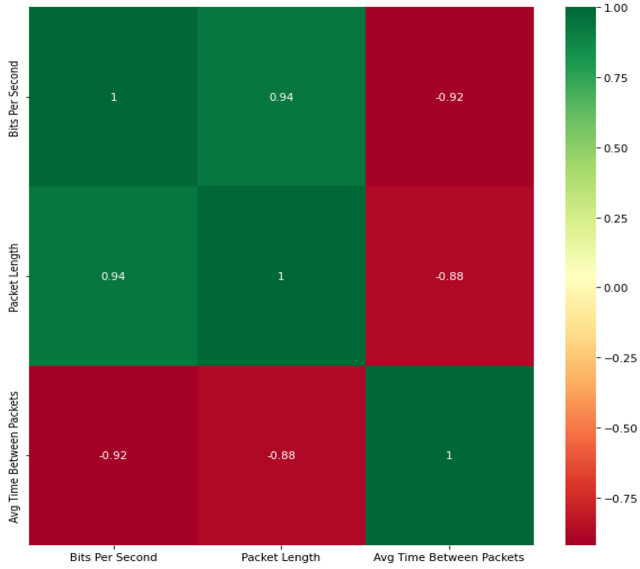

Figure 10: Correlation between features

7 Conclusions and Future Work

We have explored the extraction of QoE metrics of encrypted Zoom video conferencing traffic. By using Decision Tree algorithms, which have emerged as a predominant classification method for VoD traffic, we have obtained similar accuracy results for classifying resolution on our relatively small dataset of 720 samples. We further used a Neural Network model to predict additional QoE metrics that were extracted from the Zoom application and calculated from the video streams themselves. We have demonstrated excellent accuracy in predicting the Naturalness Image Quality Evaluator (NIQE). We are presently working on enhanced automations that will allow us to capture larger datasets in order to drive a more robust and generic prediction model. In addition, we plan to expand the scope of this research to other video conferencing tools, notably Microsoft Teams and Google GoToMeeting. We further plan to create a large dataset of Video Conferencing clips and perform a Mean Opinion Score (MOS) experiment with human observers. We will then compare the results of the metrics to those of the MOS and perform classification for both.

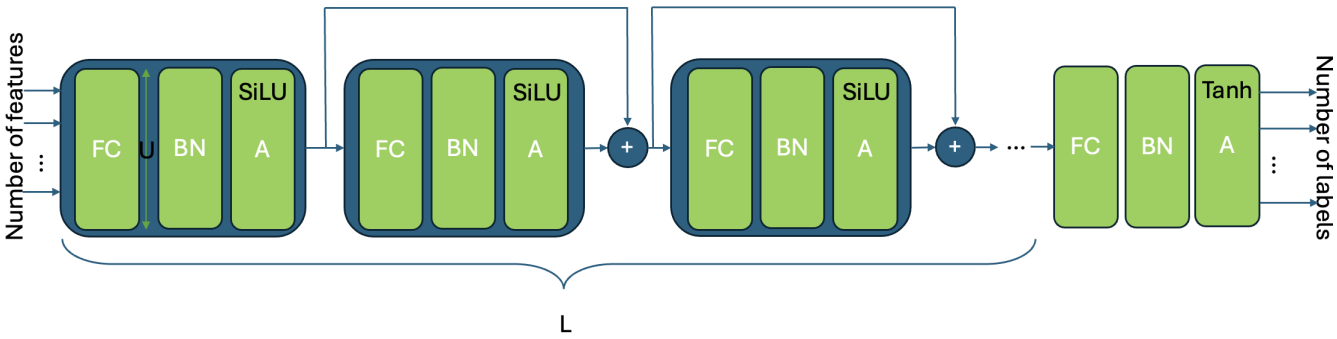

Figure 11: QoE prediction Neural Network architecture

8 Acknowledgments

We thank undergrad students for their diligent contribution to this paper: Sharon Golkarov, Yogev Drori, Nadav Hadad, Tamir Cohen, Max Polnikov, Or Shamir.

References

[1] Francesco Bronzino et al. "Inferring streaming video quality from encrypted traffic: Practical models and deployment experience". In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 3.3 (2019), pp. 1–25.

[2] Hyunseok Chang et al. "Can you see me now? A measurement study of Zoom, Webex, and Meet". In: *Proceedings of the 21st ACM Internet Measurement Conference*. 2021, pp. 216–228.

[3] Albert Choi et al. "Zoom session quality: A network-level view". In: *International Conference on Passive and Active Network Measurement*. Springer. 2022, pp. 555–572.

[4] Giorgos Dimopoulos et al. "Measuring video QoE from encrypted traffic". In: *Proceedings of the 2016 Internet Measurement Conference*. 2016, pp. 513–526.

[5] Ran Dubin et al. "Video quality representation classification of Safari encrypted DASH streams". In: *2016 Digital Media Industry & Academic Forum (DMIAF)*. IEEE. 2016, pp. 213–216.

[6] Augusto Espin and Christian Rojas. "The impact of the COVID-19 pandemic on the use of remote meeting technologies". In: *Available at SSRN 3766889* (2021).

[7] Anja Feldmann et al. "The lockdown effect: Implications of the COVID-19 pandemic on internet traffic". In: *Proceedings of the ACM internet measurement conference*. 2020, pp. 1–18.

[8] Corrado Gini. "On the measure of concentration with special reference to income and statistics, Colorado College Publication". In: *General series* 208.1 (1936).

[9] Craig Gutterman et al. "Requet: Real-time QoE metric detection for encrypted YouTube traffic". In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 16.2s (2020), pp. 1–28.

[10] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[11] Kyle MacMillan et al. "Measuring the performance and network utilization of popular

video conferencing applications". In: *Proceedings of the 21st ACM Internet Measurement Conference*. 2021, pp. 229–244.

[12] M Hammad Mazhar and Zubair Shafiq. "Real-time video quality of experience monitoring for https and quic". In: *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE. 2018, pp. 1331–1339.

[13] Oliver Michel et al. "Enabling passive measurement of zoom performance in production networks". In: *Proceedings of the 22nd ACM Internet Measurement Conference*. 2022, pp. 244–260.

[14] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. "Making a "completely blind" image quality analyzer". In: *IEEE Signal processing letters* 20.3 (2012), pp. 209–212.

[15] Antonio Nistico et al. "A comparative study of RTC applications". In: *2020 IEEE International Symposium on Multimedia (ISM)*. IEEE. 2020, pp. 1–8.

[16] Irena Orsolic, Mirko Suznjevic, and Lea Skorin-Kapov. "Youtube qoe estimation from encrypted traffic: Comparison of test methodologies and machine learning based models". In: *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2018, pp. 1–6.

[17] Irena Orsolic et al. "A machine learning approach to classifying YouTube QoE based on encrypted network traffic". In: *Multimedia tools and applications* 76 (2017), pp. 22267–22301.

[18] Karl Pearson. "VII. Note on regression and inheritance in the case of two parents". In: *proceedings of the royal society of London* 58.347-352 (1895), pp. 240–242.

[19] Karl Pearson. "LIII. On lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin philosophical magazine and journal of science* 2.11 (1901), pp. 559–572.

[20] ITUTG Recommendation. *1080-Quality of experience requirements for IPTV services*. 2008.

[21] Henning Schulzrinne et al. *RFC3550: RTP: A transport protocol for real-time applications*. 2003.