Predicting Video QoE from Encrypted Traffic: Leveraging Video Fingerprinting and Providing System-Level Insights

Somiya Kapoor*, Ethan Witwer*, David Hasselquist*†, Mikael Asplund*, Niklas Carlsson*
*Linköping University, Sweden
†Sectra Communications, Sweden

***Abstract*—Accurate Quality of Experience (QoE) estimation is both important and challenging for network operators: important since it is crucial for improving user satisfaction and challenging due to end-to-end encryption preventing them from accessing critical application-level metrics, such as video quality and buffering, forcing them to rely on indirect network-level data for QoE assessment. In this work, we address the challenge of predicting QoE from encrypted video traffic through two key contributions. First, we adapt state-of-the-art video fingerprinting techniques, originally developed for content identification, to accurately predict QoE in encrypted settings. We demonstrate that our best tested model achieves high prediction accuracy across diverse and fluctuating network conditions, establishing it as a reliable QoE predictor. These findings underscore the potential of deep learning-based classification techniques for predicting QoE from network traffic, offering a practical tool for QoE management across diverse streaming environments. Second, we conduct a systematic analysis of essential system-level factors to provide practical guidance for network operators. Here, we examine the impact of training data composition, varying network conditions, and model generalizability across sites. Our results reveal that robust QoE prediction is possible with our best tested model even with limited training data and varying conditions, making our approach feasible for real-world deployment. By enabling QoE prediction without requiring access to encrypted video content, our approach stands to support network operators in proactively managing QoE and dynamically adjusting network resources, ultimately enhancing user satisfaction.**

I. Introduction

With online video consumption becoming central to daily life, accurate Quality of Experience (QoE) estimation is crucial for network operators aiming to enhance user satisfaction. Today, streaming platforms dominate global Internet traffic [1], with services like YouTube delivering entertainment, news, and information to billions of users [2]. While these services are typically delivered over-the-top using advanced Content Delivery Networks (CDNs), network operators are responsible for maintaining the infrastructure that connects users to content, while ensuring that each user receives sufficient bandwidth and stable connections. With the help of accurate QoE estimation, a network operator can implement informed resource allocation strategies and minimize playback interruptions to improve user satisfaction [3], [4].

However, the rise of encrypted traffic (e.g., HTTPS) prevents direct access to critical application-level metrics like video quality, bitrate switches, and rebuffering – all of which are key to understanding user experience and to diagnosing and addressing issues affecting user experience. This lack of direct access poses challenges for network operators, as they must depend on network-level data to infer QoE and obtain proxy insights. Moreover, network conditions may change over time, leading to fluctuations in latency and available bandwidth. This introduces inconsistencies in user experience: playback may at times be smooth and at other times be plagued by buffering and stalls. As encrypted traffic becomes more complex and widespread, operators need alternative methods to assess and manage QoE without direct access to application-level data.

To tackle these challenges, we investigate two main objectives: (1) leveraging state-of-the-art video fingerprinting techniques to predict QoE from encrypted network traffic accurately, and (2) systematically analyzing key system-level factors that affect practical deployment and performance optimization for network operators.

First, motivated by recent advances in video fingerprinting, we investigate whether state-of-the-art fingerprinting techniques, originally designed to identify video content from encrypted traffic, can be adapted for accurate QoE prediction. These methods, known for their robustness in detecting features of encrypted streams, prove effective for QoE prediction across varying network conditions. Our results confirm this, with a modified version of the Video-Adapted Robust Fingerprinting (vRF) model [5] achieving high accuracy. Unlike most prior QoE prediction studies, our selected models are open source, encouraging further research and development.

Second, we present a systematic analysis of key system-level questions to guide practical deployment and optimize performance. For example: (1) *Does it matter if the viewed videos are part of the training dataset?* Our findings suggest it does not, which simplifies training data selection for network providers. (2) *How sensitive are accuracies to differences or changes in training conditions?* While accuracy does drop when network conditions deviate from those of the training data, we find it remains relatively high even when average bandwidth differs by up to a factor of eight, with best results when the training data reflects higher-bandwidth conditions. (3) *Are there benefits to sharing and/or combining training data across sites with different conditions?* We find that client group-specific models yield the best results, although a global

ISBN 978-3-903176-72-0 © 2025 IFIP

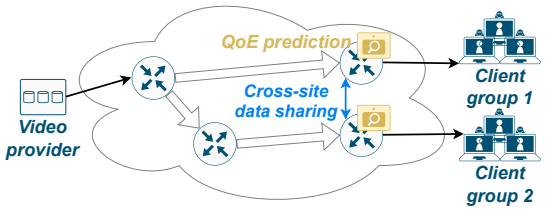


Fig. 1: Example deployment scenario of QoE prediction.

model performs better than a mismatched model. (4) *How much training data and time are needed to achieve high accuracy?* High accuracy is achievable with relatively short training sequences: the first minute of data is often sufficient for reliable QoE prediction throughout the entire session. By examining these critical deployment factors, we contribute insights that both advance the technical accuracy of QoE prediction in encrypted settings and provide practical guidelines for network operators aiming to monitor and optimize user experience under heterogeneous and evolving conditions.

Outline: Section II provides background and an overview of related works. Section III outlines the datasets and QoE metrics used. Section IV presents, analyzes, and compares the QoE prediction performance of our adapted fingerprinting models. Section V presents performance analysis tests to address each of the outlined system questions of interest. Section VI examines the individual components that constitute QoE in detail, discussing how they interact and contribute to the overall QoE assessment. Finally, Section VII presents our conclusions and summarizes key findings.

II. Background and Related Work

A. Adaptive Bitrate Streaming

Dynamic Adaptive Streaming over HTTP (DASH) [6] is the standard protocol for video streaming, often used as-is but also adapted by platforms like YouTube [7] and Netflix [8]. It enables clients to dynamically request video segments at varying quality levels based on network conditions, which allows for optimization of QoE metrics such as overall quality, number of quality switches, and time spent rebuffering. However, due to encryption, network operators lack access to the application-level data required to fully assess or optimize QoE. This is in contrast to QoS metrics (e.g., throughput and delay), which are easily accessible but have less correlation with user experience.

B. Video Quality of Experience

While QoE varies subjectively, prior research [9], [10] has identified critical factors shaping users' viewing experiences, including the overall quality, frequency of quality switches, and rebuffering time, forming the basis for recent advancements in ABR controllers [11]. Many QoE prediction models based on such metrics have been designed to be implemented in the network, as in Figure 1, where prediction is performed by routers for the clients they serve. However, these models are not typically tested in the multi-site case where data from different client groups can be shared between nodes – we return to this situation in Section V-C.

Several studies [12]–[16] use network-layer features with machine learning or heuristics to predict an overall measure of QoE or specific metrics, such as quality, quality switches, and stalls. While some enable near real-time classification, they often lack accuracy and comprehensive evaluation across varied network conditions. Other systems attempt to replicate client state, such as BUFFEST [17], which employs a buffer emulator to replicate the client's buffer conditions and predict stalls in real-time. Two key differences between these studies and our methodology are that existing techniques have not been comprehensively evaluated in different network conditions and generally offer lower accuracies.

More recent studies have begun to employ deep learning. Rec-Live [18] uses a Long-Short Term Memory (LSTM) network to differentiate live streams from video-on-demand and a Random Forest model to predict quality and stalls. Similarly, Oura et al. [19] use an LSTM to predict bitrate, resolution, and stalls, followed by a Random Forest model to compute a mean opinion score. Shen et al. [20] use a Convolutional Neural Network (CNN) to predict quality, rebuffering events, and startup delay. Loh et al. [21] compare deep learning and Random Forest models for predicting events, including quality, quality switches, and stalls, while Seufert et al. [22] investigate various models and transfer techniques to achieve better performance in unseen conditions. Others have proposed real-time quality prediction using a fingerprint database, achieving very high accuracy [23], [24]; but this would require network operators to generate fingerprints for popular videos, rendering these approaches impractical. In contrast to prior work, we focus on practical challenges and systems insights into how to best implement and use state-of-the-art traffic analysis solutions. Furthermore, to address the current lack of available source code, our work is fully open-source and reproducible.

Network-level policies aim to optimize QoE, with server-assisted techniques [25], [26] increasing infrastructure load but not effectively addressing varying network conditions and client interactions. Edge-based solutions [27], [28] mitigate some limitations but depend on tailored clients providing device information, lacking local bitrate adaptation. Krishnamoorthi et al. [3] propose temporary bandwidth caps to enhance buffering and stall recovery. Other works [4], [29] use heuristics or deep learning for QoE and fairness, though with limitations. Caching decisions based on QoE measurements to improve QoE at scale [30]–[34] have also been proposed. These approaches depend on client state awareness and would benefit from our QoE metric predictions.

C. Traffic Analysis

Traffic analysis is a group of techniques used to determine details about encrypted traffic by analyzing network-level patterns, such as packet directions, timing, and sizes [35], [36]. Several works have investigated *video fingerprinting*, which employs heuristic and machine learning-based tools to identify videos streamed over encrypted connections [37].

Recent video fingerprinting models demonstrate a remarkable ability to extract useful information from encrypted video streams. Among them, Beauty and the Burst [38] stands out as the canonical deep learning model, achieving high accuracy with a simple CNN and high-level features. Recently, Carlson et al. [5] adapted two *website fingerprinting* models (designed for identifying websites), Deep Fingerprinting (DF) [39] and Robust Fingerprinting (RF) [40], to the DASH live streaming context, improving upon Beauty and the Burst with high accuracy in challenging scenarios, including variable bandwidth conditions, training for unknown conditions, and variations in live latency between training and testing datasets.

Though Carlson et al.'s two adapted models, Video-Adapted DF (vDF) and vRF (based on DF and RF, respectively), are intended for *classification* of videos, we consider that, due to their great success in identifying key features of encrypted video streams even in challenging network conditions, a similar approach is appropriate for *QoE estimation* of video streams. In this work, we thus investigate and further refine vDF and vRF to predict QoE metrics for encrypted streams. For comparison, we also consider default DF and RF, with more granular input formats.

III. Dataset and QoE Metrics

Designing robust QoE prediction systems presents several challenges. One challenge is managing heterogeneous network conditions. Another is utilizing training data across different network conditions, and how to best utilize training data and system resources in general. We perform a systematic study to provide insights into these key challenges. For fair comparisons of different solutions, we use a sequence of tests. These tests provide quantitatively supported insights into the best ways to address various challenges. We rely on an open, large-scale dataset and well-established performance metrics. In this section, we outline our dataset and metrics.

A. Dataset

To allow direct comparisons of QoE prediction performance across a wide range of scenarios, each with heterogeneous and time-varying bandwidth, we use an extended version of the *LongEnough-variable* dataset [5], [37]. This dataset was specifically crafted to reflect a diverse range of bandwidth conditions encountered during real-time streaming and includes both network traces and QoE metrics for 100 live-streamed videos representing a variety of content.

During data collection, each sample was generated while streaming over a bottleneck link following a session-unique bandwidth pattern, derived from authentic real-world LTE traces [41]. To capture different average bandwidth conditions, scale factors of 1, 2, 4, and 8 were applied to the bandwidth patterns to represent a spectrum of network conditions, from a highly constrained real-world LTE scenario (scale factor 1, average of 3.85 Mbps) to more favorable scenarios (scale factor 8, reaching peaks around 100 Mbps), allowing us to compare model performance over different QoE ranges. For each set of average conditions (Var-X, where X is the scale factor), every video was streamed up to the 10-minute mark (for a maximum duration of 10 minutes), initiated from 10 distinct starting points spaced at 60-second intervals from the beginning of the video. 10 samples were collected at every starting point, resulting in a total of $4 \times 100 \times \sum_{i=1}^{10} i \times 10 = 220,000$ minutes $\approx 3,667$ hours of streamed video traffic.

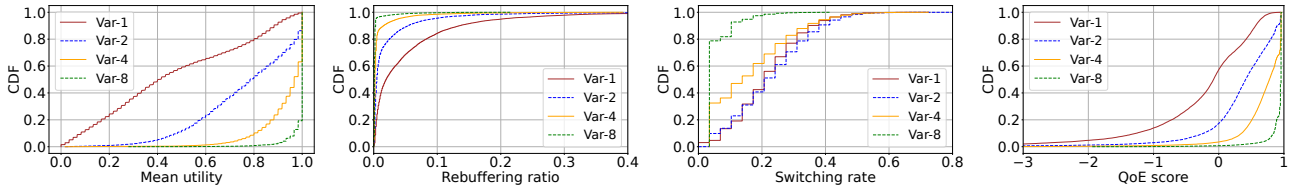
The videos in the dataset were streamed at three distinct quality levels: 1000 kbps (1K), 2000 kbps (2K), and 4000 kbps (4K). This offers a broad view of the streaming quality spectrum. These quality levels, along with variations thanks to variable bitrate (VBR) encoding, allow us to capture the segment size variations of modern streaming services under different network conditions. Using the *LongEnough-variable* dataset, we are also able to capture adaptability characteristics: note that many quality switches and lower average qualities are expected with a scale factor of 1 (average bandwidth less than 4000 kbps), while much fewer switches should occur with a scale factor of 4 or 8. For a comprehensive explanation of the induced bandwidth limitations and dataset collection process, see Appendix B in Hasselquist et al. [37].

While there are other factors that can impact the QoE of modern DASH clients, available bandwidth is likely the factor that impacts their QoE the most. In Section III-C, we show that the above selection of bandwidth scales and bandwidth variations captures a wide range of interesting QoE values.

B. QoE Metrics

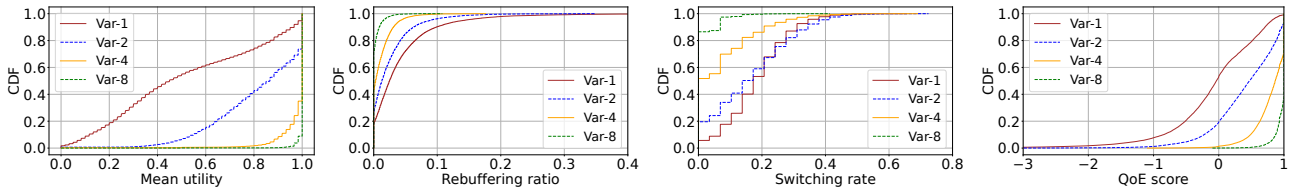
To measure QoE, we extract data from the QoE traces in the dataset and calculate normalized average statistics over two time intervals: the first or last minute of a trace. These two time periods are particularly interesting for this dataset, since the first minute captures the most transient and often most challenging conditions for a client (as the client initially has no data and must both build a buffer and find out what playback rate it can sustain), while clients have typically reached steady-state conditions by the last minute (although bandwidth still changes uniquely with time for all clients).

To ensure consistency with prior works [11], [42] in terms of combined QoE score, we use the same definition of QoE and normalized versions of the three most commonly used QoE metrics – mean utility, rebuffering ratio, and switching rate – as well as an overall QoE metric based on these normalized metrics. These metrics are intuitive and easy to interpret, making them valuable on their own, with adaptive bitrate (ABR) algorithms typically designed to take them into account. With our normalization, each value is scaled between 0 and 1, making it easier to interpret and compare results across different scenarios. The combined QoE metric is defined such that higher scores capture the desirable goals of (1) delivering high video quality, (2) minimizing rebuffering time, and (3) reducing the frequency of bitrate switches during playback. This approach ensures a comprehensive evaluation of the user's viewing experience by addressing both the quality of the stream and the smoothness of playback under varying network conditions. The exact definitions of the components of our selected QoE metric are described next.



(a) Mean utility (b) Rebuffering ratio (c) Switching rate (d) QoE score

Fig. 2: First-minute differences observed across bandwidth conditions (Var-1 to Var-8) with regard to the three key metrics and overall QoE scores.



(a) Mean utility (b) Rebuffering ratio (c) Switching rate (d) QoE score

Fig. 3: Last-minute differences observed across bandwidth conditions (Var-1 to Var-8) with regard to the three key metrics and overall QoE scores.

Mean Utility: We calculate the mean video quality using a logarithmic utility function:

$$\bar{v} = \frac{1}{N} \sum_{i=1}^{N} \frac{\log(r_i/r_{\min})}{\log(r_{\max}/r_{\min})}, \tag{1}$$

where N represents the total number of segments in the video, $r_{\min}$ and $r_{\max}$ correspond to the minimum and maximum bitrates possible for each such segment. This definition provides a normalized score reflecting bitrate closeness to the maximum, capturing the overall viewing experience in terms of resolution, with higher scores indicating better quality.

Rebuffering Ratio: We calculate the rebuffering ratio as:

$$\rho_{\text{rebuf}} = \frac{T_{\text{rebuf}}}{T_{\text{session}}}, \tag{2}$$

where T_{rebuf} is the total rebuffering time and T_{session} is the total duration of the observed playback session. This ratio represents the proportion of the observed session duration spent buffering, with higher values indicating more disruptions and lower values reflecting smoother playback.

Switching Rate: We calculate the switching rate as:

$$p_{\text{switch}} = \frac{N_{\text{switch}}}{N-1}, \tag{3}$$

where N_{switch} is the total number of bitrate switches and N is the total number of video segments. This metric captures the frequency of bitrate changes during playback, with higher values indicating more frequent quality adjustments and potential disruptions. A lower switch ratio reflects more stable, consistent streaming.

Combined QoE Score: The QoE score is calculated as a linear, weighted combination of the above key factors:

$$\text{QoE} = \bar{v} - \beta \cdot \rho_{\text{rebuf}} - \gamma \cdot p_{\text{switch}}, \tag{4}$$

where β and γ are weighting factors. Like Chen et al. [11], we select $\beta = 10$ and $\gamma = 1$, placing greater emphasis on users being frustrated by rebuffering times.

C. QoE Differences between Trace Classes

To provide some intuition for the relative challenges that each bandwidth category in the dataset presents, Figures 2 and 3 show the normalized metrics and QoE for the first and last minute of all trace categories. This selection spans bandwidth conditions from Var-1 (worst) to Var-8 (best), capturing a broad range of normalized values across key metrics and diverse overall QoE scores. This diversity highlights the dataset's suitability for evaluating if training data from one type of conditions may be helpful when predicting the QoE of a different client group's network conditions. For example, we note that the bandwidth conditions were selected such that we see substantial differences across conditions, regardless of metric, with the most challenging conditions (Var-1) spanning almost the full range of mean utilities and the best conditions (Var-8) typically allowing for a mean utility close to one.

IV. Can State-of-the-Art Fingerprinting Attacks become Good QoE Predictors?

One major goal of this study is to determine whether state-of-the-art fingerprinting attack models, designed for content identification, can provide accurate QoE prediction.

A. Adapted Fingerprinting Techniques

Here, we outline the four fingerprinting attack models that we have adapted for QoE prediction purposes.

DF: Deep Fingerprinting (DF) [39] is a website fingerprinting attack that uses a $1\times5{,}000$ matrix to represent the first 5,000 packets in a network trace, where each entry indicates the direction of a packet ($+1$ for outgoing and -1 for incoming, from the client's perspective). The matrix is padded with zeros if there are fewer packets and truncated if there are more. A simple neural network with multiple convolutional layers processes this matrix to extract patterns from the packet sequence. While DF is effective at identifying websites based on raw traffic, it may be sensitive to variations in data structure.

TABLE I: Accuracy (%) when training and evaluating under same average bandwidth conditions, with five QoE classes.

(a) **First minute, exact**

Bandwidth case ↓	Method			
	DF	vDF	RF	vRF
Var-8	74.5	76.5	73.6	**85.5**
Var-4	50.0	53.4	60.7	**74.8**
Var-2	46.7	50.3	60.8	**71.6**
Var-1	63.4	64.9	71.1	**76.8**

(b) **First minute, ±1 class**

Method			
DF	vDF	RF	vRF
97.5	97.8	98.4	**99.2**
90.5	92.0	96.4	**98.6**
89.7	90.7	96.1	**98.7**
95.2	95.8	99.3	**98.8**

(c) **Last minute, exact**

Method			
DF	vDF	RF	vRF
57.6	56.4	68.9	**76.0**
42.8	43.9	58.4	**65.6**
35.6	33.5	48.1	**59.3**
55.9	51.6	63.9	**70.5**

(d) **Last minute, ±1 class**

Method			
DF	vDF	RF	vRF
93.3	91.8	99.1	**99.1**
82.3	82.1	93.5	**96.7**
80.2	77.1	91.5	**96.0**
90.1	88.7	97.8	**98.1**

RF: Robust Fingerprinting (RF) [40] builds on the success of CNNs in website fingerprinting attacks and surpasses previous methods by introducing a new input format called the Traffic Aggregation Matrix (TAM). This matrix separates packets into two rows by direction and groups them into time intervals, called *buckets*, which make up the matrix columns. For the website fingerprinting task, the TAM is specifically adjusted to divide 80 seconds of network traffic into a $2\times1{,}800$ matrix, with each bucket representing 0.044 seconds of traffic.

vDF/vRF: Video-Adapted DF (vDF) [5] is Carlson et al.'s adaptation of DF for identification of video traffic. The primary change from default DF is the introduction of a time series input with buckets containing sums of packet sizes. Video-Adapted RF (vRF) [5] follows a similar approach, extending RF's TAM to also take packet sizes into account. We find that few further modifications are needed to achieve high performance in the QoE prediction setting due to the models' already strong ability to extract features from encrypted video traffic. Based on preliminary testing, the only changes we make are to use a higher learning rate of 0.01 for vRF and a fixed value of 50 epochs for every model, replacing the initial settings of 0.0005 for learning rate and 30 epochs.

B. Training and Labeling

Ground Truth Labeling: Using the QoE logs in the dataset, we label every relevant 60-second interval of the traffic traces (first and last minute) with the normalized QoE values defined above as well as the overall QoE score for that period.

QoE Prediction Granularity: In our evaluations, we use five similarly frequent QoE classes but also report results for varying granularity of the QoE classification. The default choice of five classes matches the scoring scale distinguishable by humans [43] and is often used in various contexts (e.g., Likert tests, student grades, etc.). Furthermore, since in this case being one class off still provides a QoE estimate that may result in reasonable flow prioritization (e.g., based on bad vs. average vs. good QoE flows), we report values for both perfect prediction and being one class off.

To decide the thresholds for the class boundaries, we sort the QoE scores of all 60-second intervals of interest (as observed across the combined set of bandwidth conditions) and then pick thresholds in two ways. For the first minute, we use *equal distribution* such that all classes contain the same number of samples (i.e., equally spaced percentiles). For the last minute, we opt for *perfect score allocation*, in which we assign all samples with perfect score (i.e., QoE = 1) to class A and then split the other samples equally among the remaining $N-1$ classes (e.g., classes B to E). In this scenario, the top class A contains 25% of samples, and the other four classes comprise the remaining 75% of samples. Note that equal distribution is used for the first minute because no samples have a perfect QoE score of 1 and perfect score allocation is thus unsuitable.

C. Performance Comparisons

We start by comparing the performance of the four models across different average bandwidth conditions. For this analysis, we assume that a network provider can collect training data under similar average bandwidth conditions to the clients whose QoE they try to classify, and that the training data exhibits similar high-level bandwidth variability as the evaluation data. This is a reasonable assumption in most cases, as the network provider can either collect data behind the same bottleneck link as the clients they monitor or perform controlled tests under similar conditions as the monitored clients. Later, we consider deviations from this assumption, including how an operator may best use training data from diverse conditions (or client groups) when targeting specific client groups.

Table I shows example results when training and evaluating using each of the tested bandwidth conditions: Var-1 (worst) to Var-8 (best). Here, classification accuracy is presented for both the first minute (initial transient period) and last minute (most stable period). We consider accuracy both when the models exactly predict the QoE class of a trace (Table I(a) and I(c)) and when allowing a margin of one class (Table I(b) and I(d)). In the latter case, the models still provide an informative, actionable estimate of the user's QoE while achieving better performance due to the less stringent requirement. We split the dataset into 80% for training and 20% for testing based on videos and perform five iterations of hold-out cross-validation to enhance robustness by varying the dataset partitions.

All models demonstrate a decreasing accuracy trend as bandwidth scale decreases from Var-8 to Var-2, but accuracy again increases once Var-1 is reached. This is likely due to lower QoE variability at more extreme scales. However, this tendency is most clearly visible when predictions must be exact: vRF, which achieves the best performance in all cases, has up to 16.7% accuracy variation between scales when an exact match is required, but its accuracy is at least 96.0% otherwise. The other models also exhibit less variability and higher accuracy when allowed to be off by one class.

We also see higher accuracy during the first minute of playback, when quality is most likely to fluctuate, than the last minute, when quality is expected to be more stable; this is true for all models. We attribute this to the models' ability to utilize the greater diversity of QoE values observed during the first minute (more balanced classes). Regardless, vRF stands

TABLE II: Accuracy (%) with vRF when using four and six classes. (Training and evaluating under same average bandwidth conditions.)

Bandwidth case ↓	4 classes				6 classes			
	First		Last		First		Last	
	Exact	±1	Exact	±1	Exact	±1	Exact	±1
Var-8	89.4	99.6	78.0	99.1	82.0	97.8	73.3	97.7
Var-4	80.4	99.0	69.9	98.0	71.5	96.8	60.7	94.7
Var-2	74.9	99.3	65.1	98.5	65.4	97.2	53.6	93.3
Var-1	85.1	99.4	79.5	98.9	72.3	98.3	62.4	96.6

out as the most effective model in both scenarios: DF, vDF, and RF achieve only 73.6-76.5% accuracy during the first minute and 56.4-68.9% during the last minute on Var-8 when an exact match is required; in contrast, vRF provides 85.5% and 76.0% accuracy, respectively. This increases further to 99.2% and 99.1% with a one-class margin.

Finally, we note that vRF also performs well when using different numbers of classes. This is illustrated in Table II, where we show example results with 4 and 6 classes. Accuracy improves slightly with 4 classes, and while it decreases with 6 classes (compared to 4 and 5 classes), the differences are relatively small, especially when considering the off-by-one case. This captures that most misclassifications are still off by at most one class with 6 classes. These results also indicate that network operators can tune the tradeoff between model performance and granularity. As vRF is the most effective model by a large margin, we only consider vRF throughout the remainder of the paper.

V. Systems Questions of Interest

We next present a systematic analysis of key system-level questions for practical use and performance optimizations.

A. How Well Do Models Generalize to Videos Outside the Training Dataset?

A key indicator of the viability of a QoE prediction model is whether it effectively generalizes to videos outside of the training set. If this is not the case, the model must be trained on popular videos (i.e., those that users are likely to watch) and may only perform well on those videos, restricting the model's utility and requiring costly continual retraining. In contrast, a model whose performance is not significantly impacted by the choice of video content for training has substantial positive implications for the resources required by network providers: QoE can be proactively managed for all users and videos, and frequent retraining is not necessary.

To evaluate vRF's ability to generalize to videos outside of the training set, we train the model using 80 videos, with 80 samples from each video included in the training dataset. For testing, we prepared two distinct datasets: one for seen videos, consisting of the remaining 20 samples from the 80 training videos, and another for unseen videos, which included all 100 samples from the 20 videos that are not part of the training set. This approach allows us to evaluate vRF's generalizability to both familiar and unfamiliar video content. Figure 4 compares the accuracy of vRF when predicting QoE for seen and unseen videos under all bandwidth conditions.

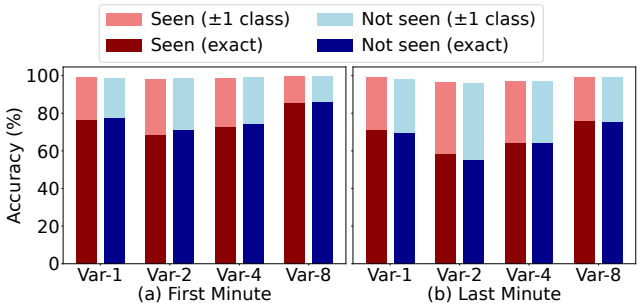


Fig. 4: Accuracy of seen vs. unseen videos.

The results indicate that vRF is equally effective regardless of whether the testing data consists of videos in the training set. In fact, vRF consistently achieves slightly higher accuracy on unseen videos during the first minute of playback, and this is sometimes the case during the last minute. We note that these differences are negligible (no more than 3% between the seen and unseen videos in Figure 4), as is also the case in the few instances where vRF performs slightly better on seen videos. This indicates that frequent retraining is unnecessary and that vRF can provide an accurate indication of users' QoE regardless of which videos they choose to watch.

B. How Sensitive are Accuracies to Differences or Changes in Training Conditions?

In this section, we train and test on different conditions. Two cases are considered: (1) when training and testing are done under varying bandwidth conditions, and (2) a more challenging scenario involving training on the first minute and testing on the last minute, also under varying conditions.

This means that training and evaluation data are collected under different conditions. For example, such a situation may arise when bandwidth conditions change significantly over time and the operator does not have enough time or resources to collect more/new data. Another example arises when a network provider does not have access to instrumented clients for all of their clients' network conditions.

Here, we assume that the network provider collects or has collected training and testing data under different average bandwidth conditions. Table III presents the results for vRF when training and testing across varying bandwidth conditions, with 80 videos for training and 20 videos for testing. Here, accuracy is presented for both the first minute and last minute, and cases in which training and testing data are from the same conditions (diagonals, corresponding to Table I, but without cross-validation here) are marked in bold. We consider accuracy both when the models must exactly predict the QoE class of a trace (Table III(a)) during the first minute and when allowing a margin of one class (Table III(b) and III(c)) during the first and last minute.

Though predicting user experience across bandwidth conditions can be difficult, our experiments indicate that vRF retains high accuracy even in this challenging scenario. We observe the same trend as reported previously when training and testing under the same conditions; i.e., accuracy decreases from Var-8

TABLE III: Accuracy (%) when training and evaluating under different average bandwidth conditions. Here, vRF with five QoE classes is used. (To avoid repetition, better use space, and simplify comparisons one variable at a time, we show results only for first minute/exact, first minute/± 1 class, and last minute/± 1 class.)

(a) **First minute, exact**

Training dataset ↓	Testing dataset			
	Var-8	Var-4	Var-2	Var-1
Var-8	**88.1**	65.7	57.2	71.9
Var-4	85.2	**81.7**	69.9	75.4
Var-2	79.6	73.5	**77.0**	76.9
Var-1	39.1	53.7	66.5	**79.9**

(b) **First minute, ±1 class**

Testing dataset			
Var-8	Var-4	Var-2	Var-1
99.6	97.5	95.7	97.6
99.5	**99.3**	98.2	99.1
99.2	98.9	**99.1**	99.2
98.3	97.9	98.1	**99.4**

(c) **Last minute, ±1 class**

Testing dataset			
Var-8	Var-4	Var-2	Var-1
99.2	93.8	94.4	61.6
99.2	**97.4**	95.9	97.8
99.2	96.6	**97.2**	97.9
69.7	86.9	94.5	**98.6**

TABLE IV: Accuracy (%) with vRF when using first minute for training and last minute for testing. (Training and evaluating under different bandwidth conditions.)

Training on first minute ↓	Testing on last minute							
	Var-8		Var-4		Var-2		Var-1	
	Exact	±1	Exact	±1	Exact	±1	Exact	±1
Var-8	68.4	97.4	52.9	92.8	40.7	91.4	29.4	94.5
Var-4	68.8	99.2	54.9	95.0	60.9	93.6	71.0	96.3
Var-2	71.0	98.0	60.9	95.1	54.9	94.2	68.8	97.0
Var-1	29.4	99.0	40.7	95.2	52.9	94.5	68.4	97.2

to Var-2 and increases again at Var-1. In Table III(a), we see it is better to train on bandwidth conditions similar to the testing set when an exact match is required, as accuracy decreases further from the diagonal. However, if a one-class margin is acceptable, the choice of training and testing conditions does not have a significant impact: accuracy is at least 95.7% for the first minute and, with the exceptions of (1) training on Var-1/testing on Var-4, and (2) training on Var-1/testing on Var-8 and vice versa, 93.8% for the last minute. This indicates that vRF can provide a good indication of QoE, regardless of the conditions of the testing set, when training on intermediate conditions (accuracy is at least 95.9% in these cases).

Temporal differences in data: In the second scenario, a network operator collects data only from the initial part of some videos, e.g., the first minute, and evaluates the model's performance on a later part of the playback session, such as the last minute. This approach eliminates the need to track network traces throughout the entire video and focuses on resource allocation and saving time. Table IV shows the performance of vRF when trained on data from the first minute (of the training set) and tested on the last minute (of the evaluation set) under various bandwidth conditions, including results for both exact predictions and those within a one-class margin. For exact predictions, training on intermediate conditions (such as Var-4 or Var-2) improves generalization across bandwidth scales. Meanwhile, within a one-class margin, accuracy remains high across all scenarios, exceeding 90% and demonstrating vRF's robustness for approximate QoE prediction.

Finally, in comparison with training and evaluating on last-minute data (Table III(c)), the results with a one-class margin are lower by a maximum of 3.0%. This suggests that a network operator must weigh the advantages of potentially slightly higher accuracies from training on the last minute against the added cost of significantly longer data collection periods (to capture steady-state behavior) when creating training data. We also note a few cases where there are significant benefits to using first-minute training data. These cases all correspond to scenarios when the accuracy is generally low due to using training data from a bandwidth scale a factor eight greater or smaller than the monitored clients (e.g., Var-1 vs. Var-8), with benefits likely being related to the first-minute training data being exposed to a wider range of scenarios and conditions.

C. *Are there Benefits to Sharing Training Data Across Sites with Different Conditions?*

Even if bandwidth conditions are known, a network operator with multiple sites may have the option to share data across sites with different bandwidth conditions, as depicted previously in Figure 1 (blue arrow between routers). In this section, we consider whether vRF benefits from such additional training data. We compare its performance when using (1) a *specialized* model, consisting of training data from the same bandwidth conditions as the testing dataset, (2) an *extended specialized* model additionally containing 20% of the data from other bandwidth scales (25% of the 80% we use for training in other experiments), (3) a *combined* ("global") model that uses training data from all client locations, and (4) a *combined scaled-down* model, which also uses data from all sites but limited to the size of specialized model, serving as a second baseline. The results are summarized in Table V.

We see substantial benefits to training a specialized model. vRF consistently achieves better accuracy with the specialized model, and the differences are greater at higher bandwidth scales: for example, during the first minute, vRF has 10.1% better exact-match accuracy on Var-8 and 7.0% better accuracy on Var-1 with the specialized dataset than with the full-scale combined dataset. However, this assumes that the correct training model is selected, necessitating the ability to estimate average bandwidth conditions.

In contrast, there does not appear to be a significant benefit to extending the specialized model. While the extended specialized model achieves 0.1% greater accuracy on Var-2 and Var-8 during the last minute, these improvements are negligible, and accuracy is typically lower than with the specialized model. This indicates that training data from the same bandwidth conditions as the evaluation data contains sufficient features for accurate classification and that addition of data from other bandwidth conditions does not provide the models with further useful information in almost any case.

Both combined models perform similarly to the extended specialized model. Comparing also with results from Table III (training and testing under different conditions), we note that the combined model consistently outperforms the worst

TABLE V: Accuracy (%) when training using training data shared across bandwidth conditions.

(a) **First minute, exact**

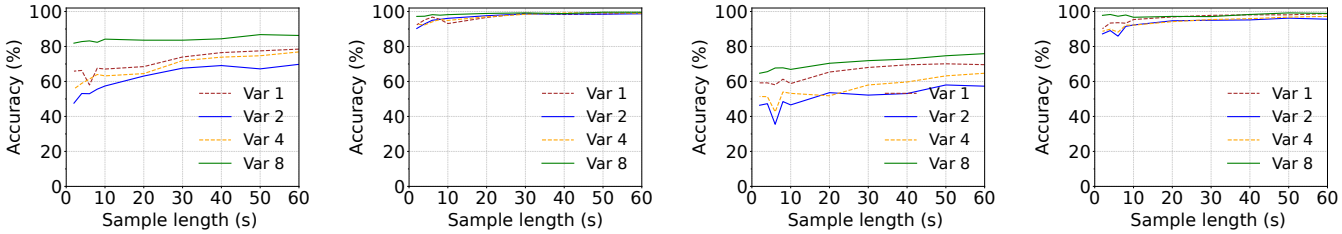
Training dataset ↓	Testing dataset Var-8	Var-4	Var-2	Var-1
Specialized	88.1	81.7	77.0	79.9
Combined, full scale	78.0	65.6	70.5	72.9
Combined, scaled down	78.7	68.5	71.6	74.4
Specialized, extended	75.7	73.7	70.9	77.2

(b) **First minute, ±1 class**

Testing dataset Var-8	Var-4	Var-2	Var-1
99.6	99.3	99.1	99.4
99.1	98.6	98.5	99.5
99.1	98.3	98.9	99.4
99.4	98.9	98.7	98.7

(c) **Last minute, ±1 class**

Testing dataset Var-8	Var-4	Var-2	Var-1
99.2	97.4	97.2	98.6
98.3	96.7	97.1	99.2
98.3	97.1	97.0	99.1
99.3	97.1	97.3	98.2



(a) First minute, exact (b) First minute, ±1 class (c) Last minute, exact (d) Last minute, ±1 class

Fig. 5: Impact of sample duration on accuracy.

matching model but typically does not offer any benefits when off by only a factor of 2. This suggests that a network operator would benefit from training specialized models when they have access to training data from various bandwidth conditions and can classify bandwidth conditions sufficiently well (e.g., within a factor of 2). On the other hand, a global model may be appropriate when it is difficult to identify the approximate bandwidth class. In this case, while exact predictions may be up to 10.1% less accurate, allowing a one-class margin still results in very high accuracy – at least 97.0% regardless of which scale is selected for the global model.

Finally, we emphasize that more data is not always better. We see this in two ways: (1) comparing the accuracies of the two combined models, where the smaller model sometimes perform better; and (2) the reduction in accuracy seen when extending the specialized models (adding training data from other conditions, fourth training set in Table V) compared to just using the specialized models on their own (first training set). These results highlight the value of collecting training data and creating specialized models for conditions similar to those experienced by client groups of interest.

D. How Much Training Data and Time are Needed to Achieve High Accuracy?

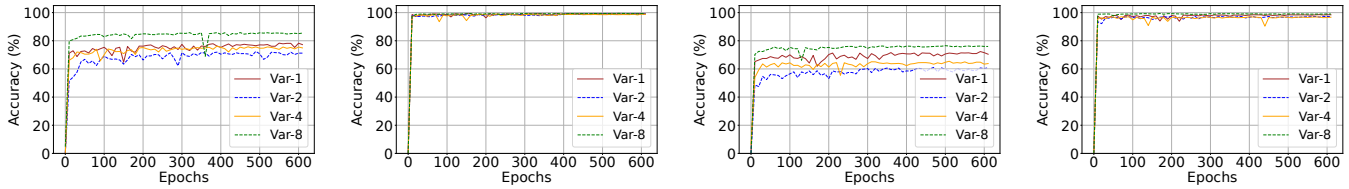
The sooner a network operator obtains an estimate of the user's QoE, the more likely it is that they will be able to apply policies that counteract any ongoing or impending QoE issues. Thus, we find it important to quantify how long a stream needs to be active before vRF can provide an accurate result. Similarly, to bootstrap vRF, initial training is required – it may be favorable for an operator to save resources by limiting training time/computational demands to the minimum required to reach the desired level of performance. In this section, we explore these issues by analyzing the impact of sample duration and epoch count on vRF.

Impact of Sample Duration: The duration of a video stream impacts the amount of data available for QoE prediction, thus affecting models' ability to assess user experience. To evaluate this, we successively trim each sample in the dataset, from 60 to 2 seconds for both the first and last minute, and track the accuracy of vRF at each sample length for all bandwidth scales. Figure 5 shows the results for exact prediction and accuracy within a one-class margin. The improvement in accuracy as sample length increases is (except for Var-8) smaller for the last minute than the first minute when an exact prediction is required, but accuracy does not ever improve by more than 15-20% when increasing the sample length from 2 to 60 seconds. Regardless of sample duration, accuracies are consistently highest for Var-8, with higher accuracies for the first minute compared to the last minute; and the lowest accuracies are observed for Var-2, which also demonstrates the greatest volatility as sample length changes.

These results indicate that near real-time prediction can be carried out with high average bandwidth and that operators can obtain an early initial indication of the user's QoE for all but exceedingly short videos. The results with a one-class margin are even more promising: for the first minute, no notable changes in accuracy are seen with sample lengths above 10 seconds, though accuracy increases by up to 8.5% (depending on bandwidth scale) up to 10 seconds. For the last minute, accuracy is even more stable, and accuracy on Var-8 is *highest* below 10 seconds. As a result, a network operator can obtain an accurate indication of QoE very early on in a session and perform near real-time QoE prediction with vRF.

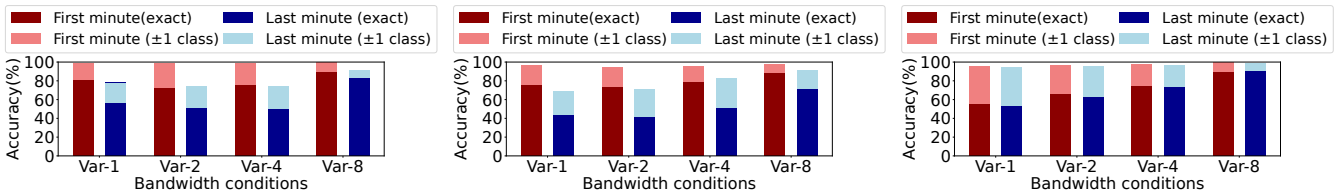
Impact of Number of Epochs: Up to this point, we have not varied the number of epochs for our tested models. Instead, we have consistently set the epochs to 50 across all conditions to keep the epoch count low and based on initial observations that high accuracy is attainable with 50 epochs. However, different settings may be appropriate for different network providers, as epoch count represents a tradeoff between training time/resources and performance. Figure 6 shows vRF's accuracy as a function of epochs for the first and last minute.

We observe in all cases that accuracy is negligible with extremely low epochs (less than 10), but it quickly rises and remains consistent after around 50 epochs, with no further improvements if the epoch count is raised further. This in-



(a) First minute, exact (b) First minute, ±1 class (c) Last minute, exact (d) Last minute, ±1 class

Fig. 6: Impact of number of epochs on accuracy.



(a) Mean Utility (b) Rebuffering Ratio (c) Switching Rate

Fig. 7: Accuracy when considering individual QoE components: (a) mean utility, (b) rebuffering ratio, and (c) switching rate.

dicates that 50 epochs represents a good choice for network operators wishing to obtain maximal accuracy without wasting resources, while it is still possible to achieve comparable results with even less training if resources are limited.

VI. Insights from QoE Components

While the overall QoE score reflects the full user experience, analyzing its individual components can help network providers identify issues, fine-tune services, and enhance performance. This analysis also helps explain variations in prediction accuracy. Therefore, we evaluate vRF's accuracy for each QoE factor: mean utility, rebuffering ratio, and switching rate.

Mean Utility: We use an 80-20 train-test split based on videos, with perfect score allocation, when predicting mean utility. Figure 7(a) shows the results, where we see a similar pattern to the overall QoE score: accuracy is highest for Var-8 and decreases as the bandwidth conditions worsen, before increasing slightly again at Var-1. This reflects that more volatile bandwidth conditions, characterized by more bitrate switches and greater diversity in mean utility values, make accurate prediction more difficult. As with overall QoE, accuracy is higher during the first minute when most quality switches occur (richer training data), and vRF's prediction of mean utility is effectively perfect when a one-class margin is allowed.

Rebuffering Ratio: For this analysis, we follow the same method as described in Section IV-B, utilizing equal distribution for the first minute and perfect score allocation for the last minute. The results are displayed in Figure 7(b). In this scenario, we again observe that accuracy is best for Var-8. However, accuracy decreases with reduced average bandwidth conditions and does not increase again at Var-1. As rebuffering is more common with lower average bandwidth, this indicates that a higher presence of rebuffering and greater variability in rebuffering ratios make it challenging to achieve high exact-match accuracy. However, accuracy is still very high with a one-class margin, and accuracies are similar for first- and last-minute predictions, regardless of bandwidth conditions.

Switching Rate: For the switching rate, we categorize traces in the same way as in the rebuffering ratio experiments and Section IV-B. The results are shown in Figure 7(c). As with mean utility, we see that accuracy decreases with bandwidth scale before increasing again slightly at Var-1; this reflects the greater challenge of predicting switching rate in the presence of more switches and greater inter-trace differences in switching behavior. Conversely, the best accuracies are attained during the first minute, reflecting vRF's ability to take advantage of more varied training data. Accuracy is high in all cases when allowing a one-class margin.

Discussion: The primary aim of the combined QoE score is to improve the overall user experience, encompassing higher video quality, shorter rebuffering times, and fewer bitrate switches. However, rebuffering is the most frustrating for users [44], as reflected in our choice of β for the QoE score. Thus, a network operator may place a higher emphasis on reducing the rebuffering time as much as possible, while focusing less on preventing bitrate switches. This could be done by predicting the rebuffering ratio directly, in which case our analysis indicates that nearly perfect accuracy can be attained for both the first and last minute if a one-class margin is acceptable, or by increasing β relative to γ. In the latter case, the promising results of vRF when predicting the QoE score with $\beta = 10$ and when predicting the rebuffering ratio (compared to the switching rate) suggest that the performance would stay the same or increase slightly.

While we see lower accuracies for mean utility and switching rate during the last minute, this is when quality is expected to be most stable, so it may not be of as much interest for a network operator to perform such predictions in any case. Perfect or nearly perfect accuracy is achievable during the first minute, when these metrics are most sensitive, with a one-class margin. Thus, a network operator can, for example, take measures to optimize and stabilize quality early on in a session and tweak clients' allocated resources throughout the remainder of the stream to reduce rebuffering.

VII. Conclusions

This paper presented (1) a robust method for predicting QoE from encrypted video traffic using adapted video fingerprinting techniques and (2) a systematic analysis of essential system-level factors, offering practical guidance for network operators. The best model we tested (vRF) achieves up to 99.2% accuracy across varied network conditions, demonstrating its adaptability to real-world scenarios. By analyzing system-level factors such as training data composition, variability in bandwidth conditions, and generalizability to unseen videos, we showed that the presented approach is scalable and feasible for practical deployment – it remains effective with minimal training data and generalizes well across sites without any need for frequent retraining – while providing a host of insights for network operators wishing to deploy vRF in their networks. Our proposed method bridges QoE prediction and encrypted traffic analysis, enabling efficient resource allocation and enhanced user satisfaction in modern streaming environments.

Code and Dataset: Our source code and dataset can be found here: https://github.com/trafnex/qoe-live.

Acknowledgement

This work was partially supported by the Swedish Foundation for Strategic Research (SSF), Karlstad Internet Privacy Lab (KIPL), the Swedish National Graduate School in Computer Science (CUGS), and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

[1] Sandvine, "2024 global internet phenomena report," ttps://www.sandvine.com/global-internet-penomena-report-2024, 2024.
[2] N. Newman, R. Fletcher, K. Eddy *et al.*, "Digital news report," Reuters Institute for the Study of Journalism, 2023.
[3] V. Krishnamoorthi, N. Carlsson, and E. Halepovic, "Slow but steady: Cap-based client-network interaction for improved streaming experience," in *Proc. IEEE/ACM IWQoS*, 2018.
[4] M. Hosseinzadeh, K. Shankar, M. Apostolaki *et al.*, "Cane: A cascade control approach for network-assisted video qoe management," *IEEE Trans. on Control Systems Technology*, 2023.
[5] A. Carlson, D. Hasselquist, E. Witwer, N. Johansson, and N. Carlsson, "Understanding and improving video fingerprinting attack accuracy under challenging conditions," in *Proc. ACM WPES*, 2024.
[6] I. Sodagar, "The MPEG-DASH standard for multimedia streaming over the internet," *IEEE MultiMedia*, 2011.
[7] X. Zhang, G. Xiong *et al.*, "Traffic spills the beans: A robust video identification attack against youtube," *Computers & Security*, 2024.
[8] A. Reed and M. Kranch, "Identifying https-protected netflix videos in real-time," in *Proc. ACM CODASPY*, 2017.
[9] D. C. Robinson *et al.*, "Subjective video quality assessment of http adaptive streaming technologies," *Bell Labs Technical Journal*, 2012.
[10] C. G. Bampis *et al.*, "Study of temporal effects on subjective video quality of experience," *IEEE Trans. on Image Processing*, 2017.
[11] T. Chen, *et al.*, "SODA: An adaptive bitrate controller for consistent high-quality video streaming," in *Proc. ACM SIGCOMM*, 2024.
[12] T. Mangla, E. Halepovic *et al.*, "eMIMIC: Estimating HTTP-based video qoe metrics from encrypted network traffic," in *Proc. TMA*, 2018.
[13] M. J. Khokhar, T. Ehlinger, and C. Barakat, "From network traffic measurements to QoE for internet video," in *IFIP Networking*, 2019.
[14] S. Wassermann, M. Seufert, P. Casas *et al.*, "ViCrypt to the rescue: Real-time, machine-learning-driven video-QoE monitoring for encrypted streaming traffic," *Trans. on Network and Service Management*, 2020.
[15] T. Kikuzuki, M. B. Mashhadi, Y. Ma, and R. Tafazolli, "Feature selection for automated qoe prediction," in *Proc. IEEE PIMRC*, 2023.
[16] F. Loh, A. Pimpinella *et al.*, "Uplink-based live session model for stalling prediction in video streaming," in *Proc. IEEE/IFIP NOMS*, 2023.
[17] V. Krishnamoorthi, N. Carlsson, E. Halepovic, and E. Petajan, "BUFFEST: Predicting buffer conditions and real-time requirements of HTTP(S) adaptive streaming clients," in *Proc. ACM MMSys*, 2017.
[18] S. C. Madanapalli, A. Mathai, H. H. Gharakheili, and V. Sivaraman, "ReCLive: Real-time classification and QoE inference of live video streaming services," in *Proc. IEEE/ACM IWQOS*, 2021.
[19] J. Oura *et al.*, "QoE estimation method with time-series features extracted from packet flows for video streaming," in *IEEE CCNC*, 2024.
[20] M. Shen *et al.*, "DeepQoE: Real-time measurement of video qoe from encrypted traffic with deep learning," in *Proc. IEEE/ACM IWQoS*, 2020.
[21] F. Loh, F. Poignée, F. Wamser, F. Leidinger, and T. Hoßfeld, "Uplink vs. downlink: Machine learning-based quality prediction for HTTP adaptive video streaming," *Sensors*, 2021.
[22] M. Seufert and I. Orsolic, "Improving the transfer of machine learning-based video QoE estimation across diverse networks," *IEEE Trans. on Network and Service Management*, 2023.
[23] H. Wu, X. Li, G. Wang *et al.*, "Resolution identification of encrypted video streaming based on HTTP/2 features," *ACM TOMM*, 2023.
[24] Y. Zhao, H. Wu, L. Chen, S. Liu, G. Cheng, and X. Hu, "Identifying video resolution from encrypted QUIC streams in segment-combined transmission scenarios," in *Proc. NOSSDAV*, 2024.
[25] S. Altamimi and S. Shirmohammadi, "QoE-fair DASH video streaming using server-side reinforcement learning," *ACM TOMM*, 2020.
[26] M. Darwich and M. Bayoumi, "Video quality adaptation using CNN and RNN models for cost-effective and scalable video streaming services," *Cluster Computing*, 2024.
[27] A. Zhang, Q. Li, Y. Chen, X. Ma, L. Zou, Y. Jiang *et al.*, "Video super-resolution and caching—An edge-assisted adaptive video streaming solution," *IEEE Trans. on Broadcasting*, 2021.
[28] X. Ma, Q. Li, L. Zou, J. Peng, J. Zhou, J. Chai *et al.*, "QAVA: QoE-aware adaptive video bitrate aggregation for HTTP live streaming based on smart edge computing," *IEEE Trans. on Broadcasting*, 2022.
[29] A. Bentaleb, A. C. Begen, R. Zimmermann, and S. Harous, "SDNHAS: An SDN-enabled architecture to optimize QoE in HTTP adaptive streaming," *IEEE Trans. on Multimedia*, 2017.
[30] F. Sun, B. Liu, F. Hou, H. Zhou, J. Chen, Y. Rui *et al.*, "A QoE centric distributed caching approach for vehicular video streaming in cellular networks," *Wireless Communications and Mobile Computing*, 2016.
[31] C. Li, L. Toni, J. Zou *et al.*, "QoE-driven mobile edge caching placement for adaptive video streaming," *IEEE Trans. on Multimedia*, 2017.
[32] S. K. Mehr, P. Juluri, M. Maddumala, and D. Medhi, "An adaptation aware hybrid client-cache approach for video delivery with dynamic adaptive streaming over HTTP," in *Proc. IEEE/IFIP NOMS*, 2018.
[33] W. Shi, C. Wang, Y. Jiang, Q. Li, G. Shen, and G.-M. Muntean, "CoLEAP: Cooperative learning-based edge scheme with caching and prefetching for DASH video delivery," *Trans. on Multimedia*, 2020.
[34] N. Carlsson and D. Eager, "Cross-user similarities in viewing behavior for 360 video and caching implications," *ACM TOMM*, 2023.
[35] D. Hasselquist, M. Lindblom, and N. Carlsson, "Lightweight fingerprint attack and encrypted traffic analysis on news articles," in *Proc. IFIP Networking*, 2022.
[36] D. Hasselquist, C. Vestlund, N. Johansson, and N. Carlsson, "Twitch chat fingerprinting," in *IFIP Network Traffic Measurement and Analysis Conference (TMA)*, 2022.
[37] D. Hasselquist, E. Witwer, A. Carlson, N. Johansson, and N. Carlsson, "Raising the bar: Improved fingerprinting attacks and defenses for video streaming traffic," in *Proc. PETS/PoPETs*, 2024.
[38] R. Schuster, V. Shmatikov, and E. Tromer, "Beauty and the burst: remote identification of encrypted video streams," in *USENIX Security*, 2017.
[39] P. Sirinam, M. Imani *et al.*, "Deep fingerprinting: Undermining website fingerprinting defenses with deep learning," in *Proc. ACM CCS*, 2018.
[40] M. Shen, K. Ji, Z. Gao, Q. Li, L. Zhu, and K. Xu, "Subverting website fingerprinting defenses with robust traffic representation," in *Proc. USENIX Security*, 2023.
[41] D. Raca, J. J. Quinlan *et al.*, "Beyond throughput: A 4G LTE dataset with channel and context metrics," in *Proc. ACM MMSys*, 2018.
[42] F. Y. Yan, H. Ayers, C. Zhu, *et al.*, "Learning in situ: a randomized experiment in video streaming," in *Proc. USENIX NSDI*, 2020.
[43] R. Likert, "A technique for the measurement of attitudes," *Archives of Psychology*, 1932.
[44] F. Dobrian, V. Sekar, A. Awan *et al.*, "Understanding the impact of video quality on user engagement," *ACM SIGCOMM CCR*, 2011.