Estimating Video Quality Using Coarse-Grained Features: Insights and Limitations from Gaussian Mixture Models

Jiamo Liu
UC Santa Barbara
Santa Barbara, CA, USA
jiamoliu@ucsb.edu

Marcin Waz
Viasat
Carlsbad, CA, USA
marcin.waz@viasat.com

Jae Chung
Viasat
Carlsbad, CA, USA
jaewon.chung@viasat.com

Kevin Moslehpour
Viasat
Carlsbad, CA, USA
kevin.moslehpour@viasat.com

David Lerner
Viasat
Carlsbad, CA, USA
david.lerner@viasat.com

Elizabeth Belding
UC Santa Barbara
Santa Barbara, CA, USA
ebelding@ucsb.edu

ABSTRACT

The significant demand for high-quality video streaming, and the wide proliferation of this application in entertainment, education and communication, has created an urgent need for methods that ensure user Quality of Experience (QoE). Traditional approaches for predicting video playback quality have focused on fine-grained network layer features, which can be computationally intensive and require extensive data collection. To reduce the high computation demand and data collection requirements, we propose to use machine learning methods to predict mean video playback bitrate using coarse-grained features derived from real-time packet scheduling priority weights. Although coarse-grained features alone may not always perform optimally due to data ambiguity, they can still enable meaningful predictions in specific contexts. In this work, we investigate the use of coarse-grained features and their ability to predict video streaming QoE. To do so, we introduce a Gaussian Mixture Model (GMM)-based filtering technique to identify regions where the model performs well. Our evaluation shows that while our model initially achieves a 56% macro-average F1 score for the entire dataset, it reaches an 81% macro-average F1 score with the filtered subset of data. This approach highlights the potential of coarse-grained features for accurate QoE estimation by identifying the regions of the dataset where these features possess sufficient discriminative power, thereby enhancing domain knowledge and the trustworthiness of model outputs.

ACM Reference Format:
Jiamo Liu, Marcin Waz, Jae Chung, Kevin Moslehpour, David Lerner, and Elizabeth Belding. 2024. Estimating Video Quality Using Coarse-Grained Features: Insights and Limitations from Gaussian Mixture Models. In *Proceedings of the 20th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '24), December 9–12, 2024, Los Angeles, CA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3680121.3697811

CoNEXT '24, December 9–12, 2024, Los Angeles, CA, USA

ACM ISBN 979-8-4007-1108-4/24/12
https://doi.org/10.1145/3680121.3697811

1 INTRODUCTION

The rapid, worldwide growth in demand for high-quality video streaming, due in part to the benefits of video in entertainment, education and communication, has created an urgent need for methods that ensure user Quality of Experience (QoE) [4, 7]. Most video providers utilize DASH [17] for video playback. DASH enables dynamic, real-time selection of the video bitrate based on network and device conditions to increase the likelihood of a satisfactory viewing experience. While video playback bitrate alone does not fully capture the user experience and factors like rebuffering also play a significant role, bitrate nevertheless serves as a useful proxy for estimating QoE. Adjustment of the video bitrate usually results in modification of video characteristics, such as resolution and frame rate, which in turn can impact QoE. However, video service flows often compete for network resources with other flows, such as web browsing and file transfer. Internet service providers (ISPs) can alter the resources allocated to each service class by assigning different priorities to packets from each class. As a result, ISPs can benefit from solutions that estimate the video playback bitrate in real-time, enabling the detection of quality drops as they occur and allowing for timely countermeasures to maintain high QoE.

Given this need, numerous attempts have been made to estimate video QoE in real-time. Some methods, such as deep packet inspection, have become obsolete due to the adoption of packet payload encryption techniques. Machine learning methods [1, 3, 8–10, 12, 18] have gained popularity in tackling this challenge. In general, the idea is to collect real-time network data to generate features that describe the current network state. These features are then used to train a machine learning model to predict video playback quality. However, these methods typically require capturing additional packet data, which is not part of typical ISP operation. Moreover, processing the volume of data needed for this task requires significant storage, computational cost, and complexity. Prior studies [11, 16] have investigated the feasibility of predicting video QoE using more coarse-grained features. However, the use of coarse-grained features inevitably leads to data ambiguity [13], as these aggregated features often fail to capture the nuanced distinctions necessary for accurate classification. For example, samples with similar features may produce different labels. This lack of distinctiveness can result in a higher rate of misclassification and reduced performance of the model. Therefore, it is in the ISP's interest to not

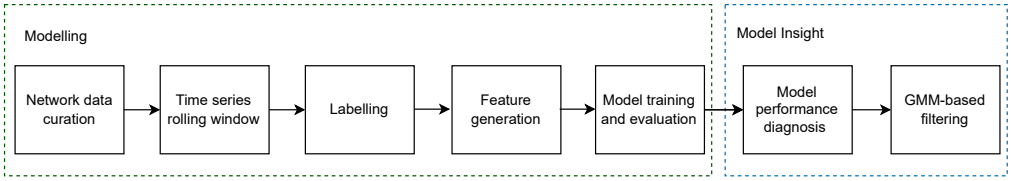

Figure 1: Framework overview.

only determine when the video service experience is unsatisfactory using readily available data, but to also identify parts of the dataset where the coarse-grained model performs well, and understand why it works well, so that it can be used most efficiently.

Based on this tension between the benefits of using coarse-grained features and the potential increase in data ambiguity and misclassifications, we pose the following research questions: *Can we differentiate the regions in which our coarse-grained model does and does not perform well, and in so doing, understand more broadly when coarse-grained features can be utilized?* And to do so, *how can we discover domain knowledge from the filtering criteria to guide future feature collection processes?* To answer these questions, we start by developing a machine learning model with our own coarse-grained features. Then we investigate whether the model has learned semantically meaningful patterns. Lastly, we apply filtering to identify where our model works well and provide explanations on why such filters would improve performance.

Our study benefits from our collaboration with the leading geosynchronous satellite internet provider, Viasat, that provides internet connectivity to 100s of thousands of residential customers as well as in-flight Wi-Fi to thousands of aircraft worldwide. Their wide customer base and extensive production data offer significant diversity in data distribution, enabling us to address the QoE prediction problem with a focus on real-world applicability and deployment. Geosynchronous satellites typically offer a fixed amount of capacity that is shared among many users utilizing different service types. The fair share allocated to each service can be adjusted by altering the packet scheduling weights assigned to each service. In the context of video streaming, the ISP aims to maximize the collective utility of all services while ensuring that video delivery meets a specific bitrate defined by the service level agreement (SLA). To build our machine learning model for predicting mean playback bitrate, we generate aggregated statistics, similar to previous work [3, 8, 10, 18], using video fair-share allocation derived from the production scheduling weights as features. These features are then used as input to a machine learning model to predict whether the video playback experience is satisfactory within a given time window. We note that the generated features are coarse-grained, as we only have six features (median, mean, skewness, variance, standard deviation and kurtosis) for each time window. Other network layer information, such as TCP flags or bytes received at the client, are not available and therefore not included as part of the machine learning model.

Our analysis finds that coarse-grained features do not work well across all scenarios due to their lack of discriminative power. To identify regions of the dataset where the model performs well, we propose a Gaussian Mixture Model (GMM)-based dataset filtering method. By generating filters to remove data ambiguity, we can better understand the limitations of our coarse-grained features and interpret why such features do not perform well in certain scenarios. This analysis offers insight into where our current features lack discriminative power. It also helps us understand when and why we should trust the model outputs and guides future efforts to collect features at a finer granularity by identifying regions of the dataset that require more discriminative power. This analysis forms the basis of our paper. Our contributions are summarized as follows:

(1) We evaluate the feasibility of predicting video QoE using real-world production data with coarse-grained features from a geosynchronous satellite network.
(2) We propose a process to identify the parts of dataset in which coarse-grained features work well.
(3) Based on the filters, we identify why our coarse-grained features lack discriminative power, guiding our future data collection process.

Ethical considerations. All data provided for this research is anonymized, ensuring that individual customers generating the video playback sessions cannot be identified.

2 QOE PREDICTION FRAMEWORK

In this section, we describe the modelling portion of our QoE prediction model framework. This is represented in the left side, denoted "Modelling", of Figure 1, which presents an overview of the full framework. The playback sessions on which we focus exclusively belong to users watching the same major streaming provider over the geosynchronous satellite network link. The content provider offers only long-form video content.

Network data curation. The satellite ISP assigns a real-time packet scheduling priority or weight to each service class (e.g., video, web browsing). This weight governs the distribution of network resources to each service class and enables the calculation of the maximum speed achievable for that service class. This calculation is known as Passive Speed Measurement (PSM) in our network management system; it is performed for each service class every few milliseconds and aggregated into 5-second averages. A low PSM for video indicates congestion in the production network, resulting in each user receiving a low fair-share allocation for video streaming. We collect 642,617 raw PSM measurements for 1,885 unique playback sessions from October 1-12, 2023.

Time series rolling window. The playback sessions durations vary, with a median of 780 seconds, as shown in Figure 2. We apply a seven-minute rolling window to each playback session and generate features and labels for that window. The window rolls forward in

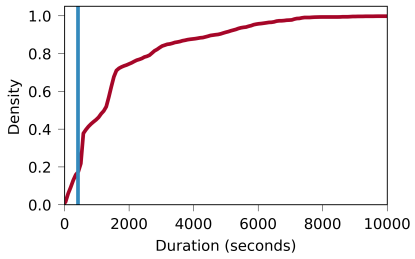

Figure 2: Duration of each playback session; vertical line is 420 seconds.

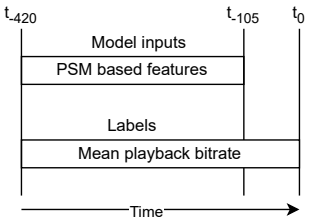

Figure 3: Temporal relationship between features and labels.

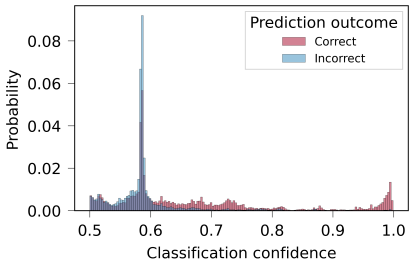

Figure 4: Model prediction confidence vs misclassification.

five second increments, effectively creating many seven minute windows from one playback session. We experimented with time windows smaller than seven minutes, similar to previous work such as [3, 8, 18], and found that the prediction results did not improve.

Our data indicates that 98% of the playback sessions do not overlap in time. Consequently, we can safely disregard sessions with overlapping periods to ensure the model focuses on learning patterns from the non-overlapping playback sessions, which constitute the overwhelming majority. Additionally, we observed that some playback sessions last less than seven minutes. These sessions are excluded from our training and testing datasets, resulting in the removal of approximately 20% of the total playback sessions.

Labelling. The ISP collaborates with a major long-form video streaming provider to provide Content Delivery Network (CDN) service, enabling us to measure the video bitrate of individual playback sessions in real-time. We compute the mean of the bitrate over 5-second windows that align with our PSM data. We generate labels that indicate whether each seven minute time window has a mean playback bitrate less than 1.8Mbps. We use 1.8Mbps because that is the target playback bitrate (approximately 720p) that the satellite ISP aims to support. We label the time window "Satisfactory" if the video bitrate is 1.8Mbps or more; otherwise, we label it as "Unsatisfactory." The dataset is described in Table 1.

Feature generation. Each seven minute time window, treated as a 420-second time series, is used to represent the distribution of the PSM values. We ignore the most recent 105 seconds of each time window during our feature generation computation as shown in Figure 3. Our rationale is that the video content downloaded most recently is not related to the playback bitrate of the current time window due to the streaming provider typically having a substantial playback buffer (in minutes). To determine the number of seconds to ignore, we conducted a grid search ranging from zero to 400 seconds, with increments of 5 seconds. We found that ignoring the first 105 seconds provided the best F1 score for the classifier model before applying any filters. We then generate statistics for each time window, including the median, mean, skewness, variance, standard deviation, and kurtosis of that time window. We experimented with the automatic feature generation package "tsfresh" [5], which generates hundreds of features, but observed that the prediction results were no better than our simple feature engineering. Therefore, we opted to stick with our simple feature strategy, in line with the principle of Occam's Razor. In summary, we created 630,641 time windows, where each time window has six features.

Table 1: Distribution of dataset labels.

Average bitrate (label)	Number of time windows
1.8 Mbps or more	445,451
Less than 1.8 Mbps	185,190
Total	630,641
Unique playback sessions	1,557

3 PRELIMINARY EVALUATION

We begin with an assessment of the performance of machine learning (ML)-based classifiers on our dataset with features generated by the process described in section 2. To do so, we employ several machine learning models for our experiments, utilizing the sklearn library in Python to train Random Forest (RF), XGBOOST and ADABOOST models. To address the class imbalance in our dataset, we set the class weight parameter in these models to "balanced." Additionally, we utilize PyTorch to train Long Short-Term Memory (LSTM) networks, comprising three hidden layers, each with a size of 50 units. The LSTM network was trained for 100 epochs to identify the model that performs best on the test set. We divided all time windows into training and testing sets, allocating 70% for training and 30% for testing. Importantly, to prevent data leakage and ensure a fair evaluation, playback sessions included in the training set were strictly excluded from the testing set. This precaution is crucial as it significantly enhances the models' ability to generalize effectively to unseen playback sessions. Hyperparameters are tuned using grid search to identify the model with the highest macro-averaged F1 score across labels. The results are summarized in Table 2. We found that all models, except for the LSTM, performed similarly well, with the Random Forest model marginally outperforming the other models tested. Consequently, we focus our discussion on the Random Forest model in the remaining sections of the paper.

At first glance, we notice that the precision is low for the "Satisfactory" label while recall is low for the "Unsatisfactory" label. Consequently, the macro-average F1 is only 56% across the two labels. This is not entirely surprising, as there are many factors our PSM measurements cannot capture. PSM measures the speed experience by determining the rate at which the fullest queues send

Table 2: Performance of each ML model.

Model	Label	Precision	Recall	F1 score
RF	Satisfactory	0.4	0.82	0.54
	Unsatisfactory	0.84	0.44	0.58
LSTM	Satisfactory	0.3	0.64	0.41
	Unsatisfactory	0.66	0.31	0.42
XGBOOST	Satisfactory	0.4	0.73	0.52
	Unsatisfactory	0.81	0.41	0.54
ADABOOST	Satisfactory	0.4	0.81	0.54
	Unsatisfactory	0.83	0.4	0.54

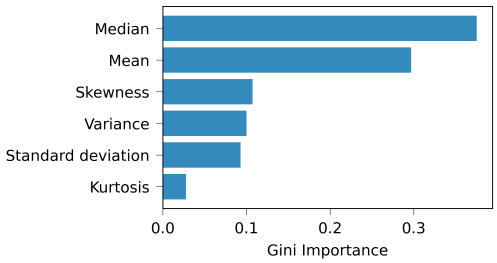

Figure 5: Feature importance.

data; however, this does not necessarily correlate with the amount of resources users actually utilize. For instance, a user could experience low network resource utilization despite high resource allocation due to a bad WiFi signal or a misaligned satellite antenna. Given these limitations, our goal in the next section is to understand the following:

(1) Does the model capture semantically meaningful patterns based on the provided features?
(2) Are samples misclassified, and if so, are there misclassification trends?
(3) Can we identify the subset of the dataset on which the model performs well? If so, how can this help us uncover some domain knowledge?

4 DIGGING DEEPER: MODEL INSIGHTS

In this section, we investigate the situations in which our model does not perform well, and we propose a method to isolate the parts of the dataset where our model does perform well by identifying regions of data ambiguity. The overview is shown as right box, labelled "Model Insight," in Figure 1.

4.1 What the Model Has Learned

Given the relatively poor performance of the model, our first step is to assess whether the model has learned something relevant. Our hypothesis is that if the model has learned a useful subset of patterns, it should make fairly accurate predictions when the classification confidence is high. The probability density function of classification confidence in Figure 4 shows that this is indeed the case. This implies that the model has learned some patterns that can confidently classify a subset of the dataset. We will investigate these patterns in greater detail later in this section.

One way to improve our model accuracy would be to ignore samples with relatively low classification confidence. However, this approach has drawbacks. For instance, determining the appropriate classification confidence threshold can be a challenging task, as it is often highly specific to the dataset in question. Furthermore, this method offers little insight into why certain samples should be excluded from the inference steps at a logical or conceptual level.

As an alternative approach, we rank the importance of each feature in Figure 5 to gain insight into the patterns learned by the model and understand how the generated features guide the decision-making process. We observe that the median of PSM measurements within the time window has the highest feature importance. This feature indicates the level of congestion during that period, as a low median PSM value typically indicates that the amount of network resources allocated to each user for video streaming is relatively low under the current service class priority. Intuitively, we expect a "Satisfactory" label when the median value is high. Figure 6 shows the distribution of the median PSM values within a time window for the two labels. We can visually confirm that time windows labeled "Unsatisfactory" usually have lower PSM values. Specifically, the median and mean PSM values for samples with the "Satisfactory" label are 51 Mbps and 47 Mbps, respectively. In contrast, for samples with the "Unsatisfactory" label, the median and mean PSM values drop to 32 Mbps and 35 Mbps, respectively. However, there is significant overlap between the two labels, potentially reducing the model's ability to distinguish between them. To quantify this overlap, we group the samples of each label into histograms with 100 bins. We then compute the Hellinger distance, a metric used to quantify the similarity between two probability distributions. The computed Hellinger distance is 0.3, where zero indicates perfect similarity and one indicates maximum dissimilarity. The Hellinger distance further supports our visual observation, indicating that the two distributions of our labels have a high degree of overlap.

To confirm that this pattern is indeed captured by the model, we first examine the value of this feature when the model makes a correct prediction, as shown in Figure 7. We observe that the model tends to identify time windows with smaller median PSM values as "Unsatisfactory." Furthermore, we examine the median PSM values when the model makes an incorrect prediction, shown in Figure 8. We note that misclassified samples are primarily those with an "Unsatisfactory" label yet high median PSM value, which contrasts with the correctly predicted samples.

In summary, we argue that the model has learned semantically meaningful patterns to classify the samples. However, the PSM measurements sometimes lack sufficient discriminative power to separate the two classes. There are multiple contributors to this ambiguity. For example, sometimes the content encoding does not require a higher bitrate; sometimes the player cannot use the available bandwidth because the encoding ladder lacks sufficient granularity; or sometimes there are home Wi-Fi or other networking issues that are not captured by the PSM. This latter case can lead to users experiencing low video bitrate even though the ISP observes a high PSM on their end. Our hypothesis is that the model can make relatively accurate predictions when the satellite link is the primary bottleneck. For example, when the median PSM values are low, users cannot achieve speeds higher than the PSM values, and confounding factors become less relevant to the average playback

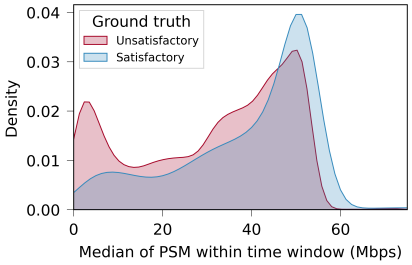

Figure 6: PSM median of the full dataset.

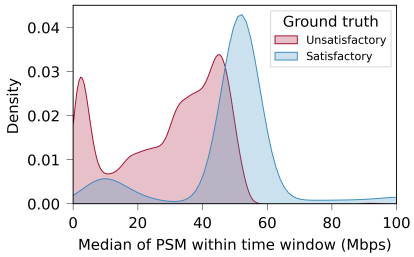

Figure 7: PSM median when model prediction is correct.

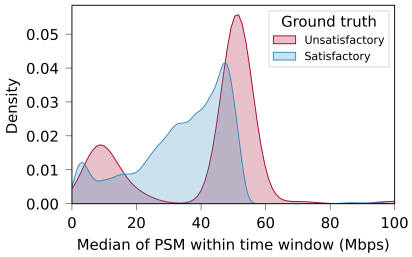

Figure 8: PSM median when model prediction is incorrect.

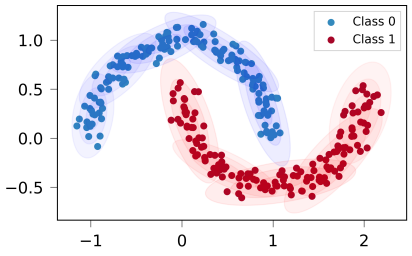

Figure 9: Toy example of how GMM models the label generating process.

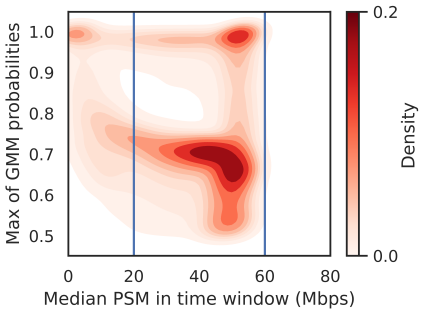

Figure 10: Median PSM vs max of GMM probabilities.

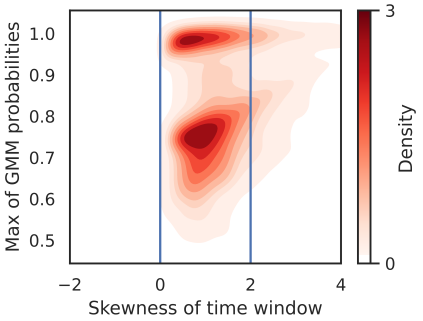

Figure 11: PSM skewness vs max of GMM probabilities.

bitrate of the time window. The lack of discriminative power of features is usually because samples of different labels are close to each other in the feature space [13], making it difficult for classifiers to draw meaningful decision boundaries.

As a next step, we propose using a Gaussian Mixture Model (GMM)-based approach to generate data filters that identify parts of the dataset where our model performs well. A GMM is a probabilistic model that assumes the data is generated from a mixture of several Gaussian distributions with unknown parameters. This allows GMM to model complex distributions of samples corresponding to each label within the dataset. By fitting a GMM to our dataset, we can generate data filters that exclude regions with high data ambiguity. Additionally, this filtering mechanism can guide us in discovering valuable domain knowledge about why the filtered subset of data achieves higher accuracy.

4.2 Dataset filtering

To verify that there is a subset of data where the model trained by our coarse-grained features is accurate, we model the generating process of each label using GMMs of the entire dataset. Each model consists of multiple Gaussian components with parameters estimated using the expectation-maximization technique; a toy example is shown in Figure 9. We can then assign a probability of a sample belonging to either label. A probability close to 0.5 indicates that the sample is likely to be generated from either label generating process; such a data point is therefore ambiguous and would be challenging for the model to classify.

To showcase how the level of data ambiguity correlates with the values of the features, we model each label generating process with three Gaussian components. We then plot the heatmap of each feature versus the maximum probability of the sample belonging to a certain label generating process. Figure 10 shows the correlation between the median of the PSM within the time window and the level of data ambiguity. We observe that most of the data ambiguity occurs when the PSM median is more than 20 Mbps. This suggests that our features are better indicators of the playback bitrate when the median of PSM measurements is less than 20 Mbps. Therefore, confounding factors may be less significant when the PSM measurements are below 20 Mbps. Simply removing samples with PSM measurements greater than 20 Mbps also aligns with the ISP's objective, as it is more important to understand and predict the video playback bitrate during periods of network congestion.

We use the original model trained in Section 3 to make predictions on the subset of the test set where the median PSM measurements are less than 20 Mbps. The prediction accuracy is shown in the second row in Table 3, labeled "Median," where we observe that the macro-average F1 score has increased by roughly 5%.

To refine our filters further, we re-run the GMM model on the reduced dataset. We observe that we can further eliminate data ambiguity by removing samples whose skewness is between zero and two in the reduced dataset, shown in Figure 11. Applying both filters to the test set results in a further increase in the F1 score, achieving a macro-average F1 score of 81%, as shown in the last row

of Table 3 labeled "Median+Skewness." We also observe that using any single filter alone fails to achieve the same level of accuracy.

A positive skewness usually indicates the presence of positive spikes in PSM measurements within the time window. It is, therefore, not surprising that this makes estimating the playback bitrate challenging because the video player has to decide whether to select a higher resolution or maintain the current playback bitrate. Such decisions typically rely on factors that cannot be captured by our PSM measurements, such as the amount of video content buffered in the playback buffer. In summary, GMM-based filtering helps uncover domain knowledge that is not immediately obvious. This method also aids in understanding when we can trust the model outputs based on the given features.

Table 3: Performance metrics on different data subsets.

Filter	Label	Precision	Recall	F1 score
None	Satisfactory	0.40	0.82	0.54
	Unsatisfactory	0.84	0.44	0.58
Median	Satisfactory	0.63	0.53	0.58
	Unsatisfactory	0.59	0.69	0.63
Skewness	Satisfactory	0.86	0.49	0.63
	Unsatisfactory	0.38	0.8	0.52
Median+Skewness	Satisfactory	0.95	0.83	0.89
	Unsatisfactory	0.62	0.87	0.73

5 RELATED WORK

Many prior studies have proposed machine learning methods to map hundreds of fine-grained network layer features to resolution, video bitrate or other forms of QoE [3, 6, 8–10, 12, 14, 18]. For instance, decision trees have been proposed to estimate video QoE using features generated from TCP flags, as well as network-layer information from QUIC connections [12]. Methods to detect video chunks from an encrypted video playback flow were proposed in [8–10]. These solutions generate features based on the packets within each chunk and then use these features to perform QoE prediction. Similarly, [3] used a chunk detection mechanism to enhance model accuracy, exploring the feasibility of using a single composite model across multiple streaming providers such as YouTube, Twitch, Netflix, and Amazon Videos. The work in [18] comprehensively evaluated multiple machine learning models for estimating video QoE with network layer features without use of a chunk detection algorithm.

Use of the reference signal received power (RSRP) to predict video QoE in a mobile network context was proposed in [1], while [15] explored use of LSTM networks to predict video QoE, utilizing the same feature set as described in [18] on an emulated geosynchronous satellite link. There is less prior work that has focused on predicting video QoE using more coarse-grained features. [11] evaluated the prediction of video QoE using aggregated statistics of TCP flow-level features. The authors achieved comparable prediction accuracy to fine-grained approaches with only 38 features and up to 60 times lower computation overhead. However, TCP flow-level and client side features may not always be readily available. Methods to estimate QoE of WebRTC videos using coarse-grained features in a video conferencing context were proposed in [16]. In each of these prior studies, the authors achieved high accuracy utilizing their respective datasets. Hence, these prior works did not have a need to identify which subsets of the data perform well. Finally, [19] and [2] explore concept drift problems in the context of network security and propose techniques such as drifting sample detection to mitigate these challenges. Their approaches are complementary to our work.

6 LIMITATIONS AND FUTURE WORK

While our proposed approach demonstrates promising results, there are several limitations and directions for future research.

More comprehensive QoE Metric: We used bitrate as the metric to estimate video QoE. However, bitrate alone is not a full representation of QoE because it omits other crucial metrics such as rebuffering. Future work should focus on integrating a more comprehensive set of QoE metrics to provide a holistic evaluation of the user experience.

Data ambiguity handling: We address data ambiguity by discarding ambiguous samples, which could result in the loss of potentially valuable information. Future research should explore more sophisticated techniques for handling data ambiguity to mitigate information loss.

Network condition variability: The current methods have not been extensively tested in a variety of network conditions and technologies. Extending the proposed methods to account for different network scenarios, such as other types of satellite and non-satellite networks or non-congested networks, would enhance their generalizability and applicability across diverse real-world contexts.

7 CONCLUSION

In this paper, we presented a method for predicting the mean video playback bitrate using coarse-grained features and machine learning techniques. Through our GMM-based filtering technique, we demonstrated that while such models may not accurately predict across all scenarios, they can still learn a subset of patterns and perform well on regions of the dataset where coarse-grained features are more indicative of the mean video playback bitrate. Our framework provides insight into the efficacy of coarse-grained features and enables discovery of valuable domain knowledge about why such features may fall short. We believe this approach can be utilized with other machine learning models that use coarse-grained features to guide future data collection processes.

REFERENCES

[1] Vivek Adarsh, Michael Nekrasov, Udit Paul, Alex Ermakov, Arpit Gupta, Morgan Vigil-Hayes, Ellen Zegura, and Elizabeth Belding. 2021. Too Late for Playback: Estimation of Video Stream Quality in Rural and Urban Contexts. In *Passive and Active Measurement (PAM) Conference*. 141–157.

[2] Giuseppina Andresini, Feargus Pendlebury, Fabio Pierazzi, Corrado Loglisci, Annalisa Appice, and Lorenzo Cavallaro. 2021. INSOMNIA: Towards Concept-Drift Robustness in Network Intrusion Detection. In *14th ACM Workshop on Artificial Intelligence and Security* (Virtual Event, Republic of Korea) *(AISec '21)*. 111–122.

[3] Francesco Bronzino, Paul Schmitt, Sara Ayoubi, Guilherme Martins, Renata Teixeira, and Nick Feamster. 2019. Inferring Streaming Video Quality from Encrypted

Traffic: Practical Models and Deployment Experience. In *Proc. ACM Meas. Anal. Comput. Syst.*, Vol. 3.

[4] L. Cantor. 2024. The Global Internet Phenomena Report. *Sandvine, Waterloo, ON, Canada, Tech. Rep* (2024).

[5] Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W. Kempa-Liehr. 2018. Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package). In *Neurocomputing*, Vol. 307. 72–77.

[6] Giorgos Dimopoulos, Ilias Leontiadis, Pere Barlet-Ros, and Konstantina Papagiannaki. 2016. Measuring Video QoE from Encrypted Traffic. In *ACM Internet Measurement Conference (IMC)*. 513–526.

[7] Florin Dobrian, Vyas Sekar, Asad Awan, Ion Stoica, Dilip Joseph, Aditya Ganjam, Jibin Zhan, and Hui Zhang. 2011. Understanding the impact of video quality on user engagement. In *SIGCOMM Comput. Commun. Rev. (SIGCOMM '11, Vol. 41)*. Association for Computing Machinery, 362–373.

[8] Craig Gutterman, Katherine Guo, Sarthak Arora, Xiaoyang Wang, Les Wu, Ethan Katz-Bassett, and Gil Zussman. 2019. Requet: Real-time QoE detection for encrypted YouTube traffic. In *ACM Multimedia Systems Conference (MMSys '19)*. 48–59.

[9] Vengatanathan Krishnamoorthi, Niklas Carlsson, Emir Halepovic, and Eric Petajan. 2017. BUFFEST: Predicting Buffer Conditions and Real-time Requirements of HTTP(S) Adaptive Streaming Clients. In *ACM Multimedia Systems Conference (MMSys'17)*. 76–87.

[10] Tarun Mangla, Emir Halepovic, Mostafa Ammar, and Ellen Zegura. 2018. eMIMIC: Estimating HTTP-Based Video QoE Metrics from Encrypted Network Traffic. In *2018 Network Traffic Measurement and Analysis Conference (TMA)*.

[11] Tarun Mangla, Emir Halepovic, Ellen Zegura, and Mostafa Ammar. 2020. Drop the packets: Using coarse-grained data to detect video performance issues. In *ACM Conference on Emerging Networking EXperiments and Technologies (CoNEXT '20)*. 71–77.

[12] M. Hammad Mazhar and Zubair Shafiq. 2018. Real-time Video Quality of Experience Monitoring for HTTPS and QUIC. In *IEEE INFOCOM*. 1331–1339.

[13] Claus Metzner, Achim Schilling, Maximilian Traxdorf, Konstantin Tziridis, Andreas Maier, Holger Schulze, and Patrick Krauss. 2022. Classification at the accuracy limit: facing the problem of data ambiguity. In *Scientific Reports*, Vol. 12. Nature Publishing Group UK London.

[14] Irena Orsolic, Dario Pevec, Mirko Suznjevic, and Lea Skorin-Kapov. 2016. YouTube QoE Estimation Based on the Analysis of Encrypted Network Traffic Using Machine Learning. In *IEEE Globecom Workshops*.

[15] Matthieu Petrou, David Pradas, and Mickaë Royer. 2024. Unveiling YouTube QoE over SATCOM using Deep-Learning. *IEEE Access*.

[16] Taveesh Sharma, Tarun Mangla, Arpit Gupta, Junchen Jiang, and Nick Feamster. 2023. Estimating WebRTC Video QoE Metrics Without Using Application Headers. In *ACM Internet Measurement Conference (IMC '23)*. 485–500.

[17] Iraj Sodagar. 2011. The MPEG-DASH Standard for Multimedia Streaming Over the Internet. 18 (2011), 62–67.

[18] Sarah Wassermann, Michael Seufert, Pedro Casas, Li Gang, and Kuang Li. 2020. ViCrypt to the Rescue: Real-Time, Machine-Learning-Driven Video-QoE Monitoring for Encrypted Streaming Traffic. In *IEEE Transactions on Network and Service Management*, Vol. 17. 2007–2023.

[19] Limin Yang, Wenbo Guo, Qingying Hao, Arridhana Ciptadi, Ali Ahmadzadeh, Xinyu Xing, and Gang Wang. 2021. CADE: Detecting and Explaining Concept Drift Samples for Security Applications. In *30th USENIX Security Symposium*. 2327–2344.