Accelerating Machine Learning Models for Video Streaming Traffic

Bence Ladóczki
Budapest University of Technology and Economics
HUN-REN Information Systems Research Group
ladoczki.bence@vik.bme.hu

***Abstract*—The increasing popularity of video streaming platforms poses new challenges to network operators as a large proportion of the streaming traffic traverses the network in an encrypted format. Network administrators face significant challenges in accurately monitoring and assessing the Quality of Service (QoS) and the end users' Quality of Experience (QoE). Tackling these difficulties, machine learning(ML)-based prediction models are frequently deployed to classify and recognise video streaming patterns and to gauge QoS/QoE. Here we follow the lines of scientific endeavours of other researchers and utilise ML models to infer user-side metrics relying solely on transport layer statistics. The performance of ML classifiers including a neural network is evaluated for YouTube Live, YouTube Gaming, Twitch and CNBC videos with compressed transport data. The ML models are trained on several gigabytes of data and for some parameters, more than 80% of accuracy is attained with a highly compressed dataset. Data compression is achieved by calculating the singular value decomposition of the dataset and keeping only the most significant vectors for prediction. The results are supplemented with comprehensive QoE metrics and the classifiers are evaluated against these metrics as well.**

***Index Terms*—quic, video streaming, QoE, QoS, machine learning, neural networks**

I. Introduction

In the past 6 years, the average smartphone users' data consumption for video streaming has grown from 3.4 GB to 16.3 GB[1]. This clearly indicates that traffic generated by video streaming and sharing platforms takes up a large proportion of the entire Internet traffic. Streaming traffic is mostly generated by large platforms such as YouTube, Facebook, TikTok, Netflix, and Twitch.

ML techniques [1] have proven to be successful in a wide range of network problems from identifying [2] Skype calls to quality of experience (QoE) prediction of video streaming platforms [3] and resource allocation prediction in the cloud for video streaming [4]. Deep learning [5], Bayesian techniques [6] and decision trees [7] are also being deployed to solve network traffic identification and classification problems.

Despite the growing number of scientific publications on machine learning(ML)-based quality of experience/quality of service (QoE/QoS) assurance framework development not much attention has been paid to the applicability and the computational performance of the various ML models. The training time of an ML model is heavily influenced by the size of the learning data. For the case under consideration, learning data is generated from the transport layer using metrics derived by observing transport packets. Although a handful of features can be derived by this approach there is not telling whether all the chosen features are indeed necessary. Important features are usually extracted using PCA (principal component analysis) and less important features can be disregarded. On the other hand, truncation in the training data can lead to decreased prediction accuracy. In this work, a matrix rank reduction technique is applied to the training data to accelerate the training process and the changes in prediction accuracy are also investigated. We scrutinise different ML models and video streaming platforms to emphasise the applicability of rank reduction techniques for this use case.

Drawing inspiration from previous research works of other authors [8]–[10], here we first set the goal of extrapolating user-side metrics solely from transport layer metrics. The methodology proposed here relies on rudimentary data, therefore it can be applied to heavily encrypted data streams. The key principle of our design is to require as little information from the transport layer as possible and to be able to gauge QoE-related metrics of a QUIC-based video streaming service.

To train ML models that can predict QoE/QoS scores up to 80% accuracy it is not uncommon to utilise several thousands of data traces [11] and years of playback time [12]. To illustrate the amount of data collected by some members of the research community let us point out a few figures from scientific publications. In [13] $573,429,697$ packets were analysed, in [14] subjective quality has been performed using 570 encoded videos, 80640 minutes of input data were generated in [15] and the authors in [16] define 228 features and select the 10 most important features with feature-importance–based selection and sequential feature selection. The performance of QoE prediction models has been evaluated over 48000 streaming sessions in [17]. These figures clearly indicate that the ML models that are used to infer user-side QoE/QoS are trained on big data and this necessitates efficient approaches to save training time.

In this work, we show that prediction accuracy for the video streaming QoE/QoS use case can be maintained even if the training data is significantly truncated and that the effective dimension of the training matrix is much less than the ambient dimension would suggest. As is well known, usually a small subset of the transformed features serves as excellent lower-

[1]https://www.ericsson.com/en/reports-and-papers/mobility-report/articles/streaming-video

979-8-3315-2096-0/25/$31.00 ©2025 IEEE

dimensional representation [18], even for video streams [19]. When a large training data is used, such an approach can save a lot of computation time. For the video streaming use-case, the fact that only a small number of features can describe the problem very efficiently has been observed in [20]. We used the best approximation in terms of the Forbenius norm to compress the learning matrix by computing its singular value decomposition (SVD) [21] and test the applicability of the method for different platforms and metrics. We then use compressed training data and derive a comprehensive QoE metric from a weighted sum of 4 different user-side parameters. The weights are varied and prediction performance with different ML classifiers is assessed in terms of accuracy, F1 score, and recall.

Finally, let us point out some similarities with past research works and position our results. While the authors in [22] train ML models on YouTube videos using a random forest classifier and extract the most important metrics for KPI (key performance indicator) prediction, they do not perform training time-related measurements. Data collection is performed in a very similar manner to what we are going to describe in [23] targeting YouTube videos, but there are no results presented in this work that can be used to discuss the performance of the ML models in terms of training time and robustness.

To the best of our knowledge, no scientific work has yet been published on the performance analysis of ML models in terms of training time for the video streaming use case. Filling in this gap this work makes the following contributions:

1) **Low-rank reduction** of the training data.
2) Assessing robustness **over several platforms**
3) **Accuracy vs. runtime** evaluation of the ML models.
4) The evaluation of a **comprehensive QoE** value using reduced datasets

Previously we reported [24] on a tool developed by our research group to collect transport layer data, which can then be used to underpin QoE/QoS-related measurements and enhancement solutions. In this work, we extend this data collector and report on how to accelerate ML models using low-rank approximations. The performance of the methodology under consideration is investigated using four different video streaming platforms (YouTube Live, YouTube Gaming, Twitch and CNBC) and we demonstrate its robustness and applicability to the QoE/QoS prediction problem.

The structure of the paper is as follows: Section II introduces the related works, then in Section III a general overview on a rank reduction technique is introduced. In Section IV we describe our data collection methodology along with the extracted features and the ML models. Then in Section V numerical results are presented. Finally, we conclude our work.

II. Related Work

With the widespread use of encrypted data streams, research on network traffic classification [25]–[31] methods has gained significant momentum in the past few years. For example, a naive Bayesian classifier in [6] has been deployed for this task to provide 65-95% accuracy. Decision trees have also proven to be efficient in solving classification problems [7]. ML classifiers for IP traffic were evaluated in [32] using multiple short sub-flows. Convolutional neural networks and stacked long short-term memory layers can classify network traffic (public QUIC dataset) with over 90% accuracy [33]. Traffic classification methodologies for QUIC have been presented by the authors of [34]. The authors in [35] have demonstrated that the classification problem for network traffic can be solved even with natural language processing by transforming features from packet traces into language-like patterns.

The problem of network traffic identification is also actively researched [36]. A comprehensive survey on ML techniques in the field can be found in [37]. Statistical fingerprinting to detect malicious flows in QUIC is presented [38] and ML was used to detect network intrusions in [39] and in [40]. To increase network security, convolutional neural networks are being tested to detect malware traffic [41].

To solve traffic classification problems not only supervised but also unsupervised learning techniques [42] are deployed [43]. ML-based traffic classification has been shown to work on the IP level [44].

Online gaming [45] is another prominent use case for which ML techniques can be utilised. Similarly, in [46] the authors investigate accurate means to account for user-perceived QoS and find that the effect of lost packets can be devastating to the gaming experience.

QoE models were developed not only for web applications [47] but also for video streaming platforms [48]–[50] to underpin the effective design of bitrate adjustment algorithms. QoE indicators for large-scale video streaming use cases were investigated in [51]. A survey on HTTP adaptive streaming (HAS) services is presented in [52] along with the respective QoE models. In [53] the authors use stalling events as a baseline metric to derive QoE models. A general taxonomy of QoS/QoE models can be found in [54]. The authors in [55] propose a methodology to adaptively choose servers with the best possible QoE using reinforcement learning. QoE models for Netflix were put forth in [56]. Theoretical discussions on user-perceived QoE can be found in [57].

Platform-specific works include research on YouTube videos [58] and the QoS estimation of wireless devices [59], iOS/Android clients [60] and general dynamic adaptive streaming (DASH) clients [61]. WebRTC-based video conferencing applications are also being evaluated [62] using ML. QoE prediction can be performed even in real-life 4G networks as demonstrated in [63].

III. A low-rank approximation

The scalability of classification algorithms to large datasets has received increased attention in the past few years [64]. Here we elucidate a data compression technique to accelerate the training of ML models. The Eckart-Young theorem [65] states that the best rank-r approximation to $\mathbf{M}$ in terms of

TABLE I
PLATFORMS, THE PREDICTED METRICS AND DATA STATISTICS

Platform	Predicted Metric	PCAP Size	Number of Videos	Application Layer	Protocol
YouTube Live	Buffer Health [0, 1, 10, 25, 35, 55, 90], Live Latency [0, 1, 10, 25, 35, 55, 90], Status	52 GB	20	NerdStat	UDP/QUIC
YouTube Gaming	—"—	22 GB	100	NerdStat	UDP/QUIC
Twitch	Playback Bitrate [0, 500, 1000, 2000, 4000, 6000], Latency To Broadcaster [0, 10, 20, 40], Buffer Size [0, 5, 10, 20], FPS [0, 25, 35, 55]	97 GB	100	Video Stats	TCP
CNBC Videos	Buffer Health [1, 5, 10, 15]	8.1 GB	100	About This Video	TCP

the Frobenius norm is given by the truncated SVD of $\mathbf{M}$. Specifically, if the SVD of $\mathbf{M}$ is:

$$\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^T$$

where, $\mathbf{U}$ is an $m \times m$ orthogonal matrix, Σ is an $m \times n$ diagonal matrix with singular values on the diagonal, $\mathbf{V}^T$ is an $n \times n$ orthogonal matrix. Then the rank-r approximation $\tilde{\mathbf{M}}$ is given by:

$$\tilde{\mathbf{M}} = \mathbf{U}_r\Sigma_r\mathbf{V}_r^T$$

where $\mathbf{U}_r$ consists of the first r columns of $\mathbf{U}$, Σ_r is a diagonal matrix containing the largest r singular values from Σ, with the remaining singular values set to zero and $\mathbf{V}_r^T$ consists of the first r rows of $\mathbf{V}^T$.

IV. PLATFORMS AND DATA COLLECTION

In this section, we describe our data collection methodology for both the application and the transport layer.

A. Application Layer

For each platform the predicted metrics are summarised in the second column of Table I and for each metric, the labels are transformed according to the given intervals as indicated in the table. For example for YouTube Live videos, the `Buffer Health` metric has been transformed to take the values [1, 2, 3, 4, 5, 6, 7] depending on the interval in which the actual value falls into (e.g. [0s, 1s, 10s, 25s, 35s, 55s, 90s]). Every metric was transformed according to the second column of Table I. `Status` is an exception because it is an integer variable.. The total size of the recorded video streams and the number of videos are indicated in the 3rd and the 4th columns in gigabytes. Various network conditions have been created by setting bandwidth limitations on the incoming traffic of the network interface of the workstations. These steps are immaterial to the main topic of this work and they are omitted.

The live videos on YouTube and live game streams provide the same user-side metrics via the *Stats for nerds* interface. These streams are transported over a heavily encrypted QUIC protocol, which implies that there is no way to intercept the streams or to peek into the packet payloads. Twitch and CNBC use TCP and it provides real-time statistics using the `About This Video` option in the web application. These statistics include `FPS`, `Buffer Size`, `Latency To Broadcaster`, and `Playback Bitrate`. CNBC[2] broadcasts short videos on its site and it provides a very convenient API by right-clicking on the videos. An automated tool has been developed to capture user-side QoS using `pyautogui`.

B. Transport Layer

Measurements are performed as follows. We choose and play live YouTube Live/YouTube Gaming/Twitch/CNBC videos randomly while packets are being captured on the network interface. We continuously monitor the downlink and the uplink traffic using *tshark* to capture TCP and UDP traffic according to the platform under consideration. After this step, the packet capture files are processed and they are split into 2 large subsets depending on target/source IP addresses to differentiate between uplink and downlink traffic. This approach enabled us to introduce transport-level features in both directions, effectively doubling the size of the training data. The output files are transformed into CSV format and then we use Python to generate data frames, resample the data (*aggregation interval*), and calculate additional metrics such as *mean*, *min*,*max*,*std* and *sum* (refer to Table II for details). The interarrival time of the packets has been calculated in a similar manner. In addition to the standard statistics, we calculate the absolute value of the fast Fourier transform of the interarrival time. The training data is generated by merging the dataset captured from the application layer with the transport layer data. The granularity of the transport layer data is usually much higher than the application layer data and we solve this issue by resampling the datasets to arrive at an identical timescale. Payload-agnostic description of features [66] is a common approach for network traffic classification and identification tasks. Here we do not rely on payload data and this method is applicable to a wide range of video streams.

C. Machine Learning Models

The inner workings of ML models we utilise in this work are well-known and documented in related books [1] and publications. As such we elide these technicalities for clarity of exposition and mention only the input parameters for the models to ensure the reproducibility of our results. Among the numerous ML algorithms we choose 3 (a neural network (MLPClassifier with 10, 20, 10 neurons in the first, second and third layers respectively), AdaBoost [67] and Random Forest) and perform single-label classifications with the `sklearn` toolkit[3]. This choice was made based on our experience with the performance of the tested models. Nevertheless, we are

[2]https://www.cnbc.com/tv/

[3]https://scikit-learn.org/stable/

far from advocating such a rigorous choice and believe that other ML/DL algorithms would show similar results with the proposed methodology. Such a comprehensive analysis is immaterial to the message we wish to convey here and therefore we focus only on a small set of ML algorithms. A similar argument can be found in [1], where we read that there are no context- or problem-independent reasons to favor one learning or classification method over another and that the apparent superiority of one algorithm or set of algorithms is due to the nature of the problems investigated and the distribution of data.

For each class i standard ML metrics are calculated:

$$P_i = \frac{TP_i}{TP_i + FP_i}, \quad R_i = \frac{TP_i}{TP_i + FN_i}, \quad F_i = 2 \cdot \frac{P_i \cdot R_i}{P_i + R_i}$$

, where TP denotes the number of true positives, FN denotes the number of false negatives and FP denotes the number of false positives. In multi-class classification, the metric for each class is computed separately. Here we use weighted averages:

$$\text{Weighted Precision} = \frac{\sum_{i=1}^{n} s_i \cdot P_i}{\sum_{i=1}^{n} s_i}$$

$$\text{Weighted Recall} = \frac{\sum_{i=1}^{n} s_i \cdot R_i}{\sum_{i=1}^{n} s_i}$$

$$\text{Weighted F1 Score} = \frac{\sum_{i=1}^{n} s_i \cdot F_i}{\sum_{i=1}^{n} s_i}$$

, where s_i refers to the number of true instances of class i in the dataset. 20% of the data is set aside for testing, while the remaining 80% is used for training.

TABLE II: The calculated features for TCP and UDP.

Feature	TCP	UDP	Explanation
packet_count	✓	✓	number of packets
bytes_per_second	✓	✓	captured6 bytes/sec.
bytes_mean	✓	✓	mean of —"—
bytes_sum	✓	✓	sum of —"—
bytes_max	✓	✓	max. of —"—
iat	✓	✓	interarrival time (iat)
iat_mean	✓	✓	mean of iat
iat_min	✓	✓	minimum of iat
iat_max	✓	✓	maximum of iat
iat_std	✓	✓	std. dev. iat
iat_fft	✓	✓	abs of FFT of iat
window_size_mean	✓		mean of window size
window_size_min	✓		min. of —"—
window_size_max	✓		max. of —"—
window_size_std	✓		std. dev. of —"—
window_size_sum	✓		sum of —"—

V. Results

The models were trained using an AMD Ryzen 9 3900X 12-core processor and 32GB of memory.

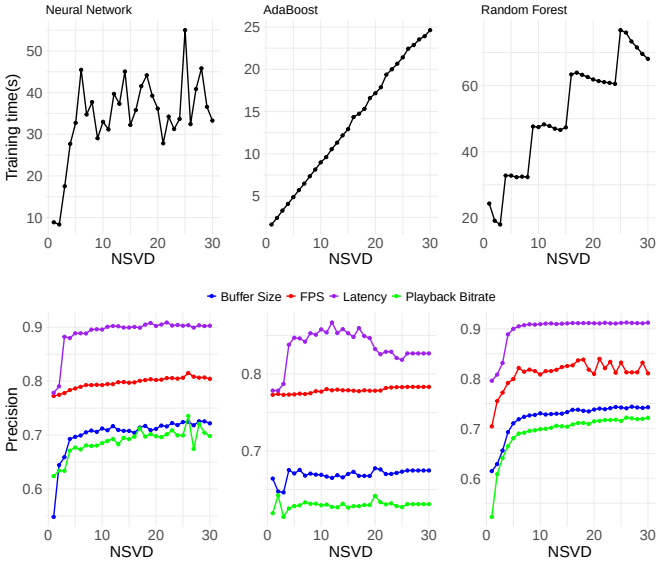

Fig. 1. Training time in seconds and the performance of 3 classifiers in terms of weighted precision for various user side metrics with respect to the number of SVD vectors for the Twitch dataset. The aggregation interval was set to 1s.

A. Training time and accuracy

First, we train a neural network, an AdaBoost classifier, and a random forest model on the Twitch dataset to predict the `Buffer Size`, the `FPS`, the `Latency`, and the `Playback Bitrate` on the client side. Prediction accuracy for this dataset with respect to the number of SVD vectors is presented in Fig. 1. Training times are presented in the upper subplots. The AdaBoost and the random forest classifiers exhibit linear behaviour in terms of learning time with respect to the rank of the training matrix, while the training time of the neural network is fairly erratic. The prediction accuracy of the neural network and the random forest classifier is much better than that of the AdaBoost model. Interestingly the prediction accuracy varies over the predicted parameters. While for example `Latency` can be predicted with over 90% accuracy `Buffer Size` can only be predicted with slightly more than 60% accuracy. As prediction accuracy depends very much on the distribution of the actual metric such comparison should always be taken with a grain of salt. In Fig. 2 the performance of the three classifiers for YouTube Live/YouTube Gaming and CNBC videos are presented. Observe similar trends for these datasets as in Fig. 1, namely that 10-15 SVD vectors are sufficient to reach the maximum possible accuracy and – once precision plateaus – one gains close to nothing by introducing more features. With these results, we demonstrate that the learning data can be efficiently compressed.

B. Comprehensive QoE

Based on the observations we made when comparing runtime and accuracy, we move forward and evaluate the models using rank-reduced training data. When there is more than one predictable feature on the client side, it is hard to tell

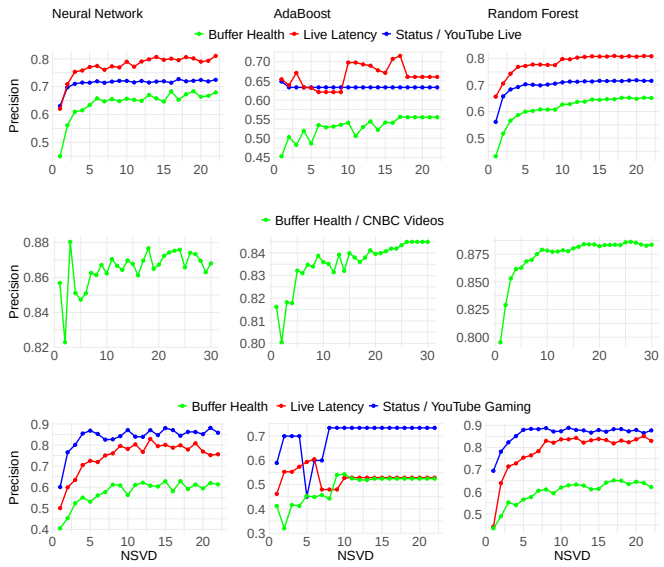

Fig. 2. Performance of 3 classifiers in terms of weighted precision for various user side metrics with respect to the number of SVD vectors for the YouTube Live dataset. The aggregation interval was set to 3s.

which one is more important than the others [68]. In some cases, larger buffer times are more favourable over low latency, while some users might prefer better playback bitrate values. To make our QoE model more robust we introduce a weighted and signed sum of metrics to arrive at a comprehensive QoE value with adjustable weights. The formula for this is according to Eq. 1. Here, we propose a metric M using four predicable client-side metrics: *Playback Bitrate* (B), *Latency To Broadcaster* (L), *Buffer Size* (S), and *FPS* (F). In Eq. 1 different weights are assigned for the client-side metrics (w_B to `Playback Bitrate`, w_L to the `Latency To Broadcaster`, w_S to `Buffer Size` and w_F to `FPS`).

$$M = \frac{1}{w_B + w_L + w_S + w_F} \left[w_B \cdot \frac{B}{B_{max}} - w_L \cdot \frac{L_{min}}{L} + w_S \cdot \frac{S}{S_{max}} + w_F \cdot \frac{F}{F_{max}} \right] \quad (1)$$

The metric is defined such that it is proportional to `Playback Bitrate`, to `Buffer Size` and to `FPS` and it is inversely proportional to `Latency To Broadcaster`. To make it independent of the weights, it is normalised with the sum of w_B, w_L, w_S, and w_F. Note that owing to the fact that the YouTube client-side application and the CNBC site do not expose these values, M is only valid for the Twitch dataset. Eq. 1 provides a flexible method to compute a performance metric by accounting for the key aspects of video streaming quality, with the ability to adjust the relative importance of each factor. The results in terms of (weighted) precision, f1 score, and recall values are summarised in Table III. Observe that the prediction performance in terms of accuracy, f1 score, and recall is the best when $w_B = 3$, $w_L = 5$, $w_S = 4$, and $w_F = 1$ (most likely because latency can be predicted with the greatest precision).

VI. Conclusion and Outlook

This work has presented a new step towards effective QoE/QoS provisioning for both TCP and UDP streams. The data compression technique enables one to quickly train ML models and recalibrate them once accuracy drops due to alterations in the transport layer. It is demonstrated that even a handful of SVD vectors are sufficient to predict user-side QoS accurately and we derived a comprehensive metric that can also be inferred from transport layer data. Our models using compressed data have been evaluated over different video streaming platforms to showcase the robustness of the proposed approach. We show that with approximately 5-10 SVD vectors the prediction accuracy plateaus out and one can gain no more by introducing more features into the model. This observation seems to prove the conjecture mentioned in Section III, namely that the ambient dimension is usually much larger than it would be necessary to efficiently describe an ML prediction problem. Our results can help improve efficiency in time-sensitive applications. Such research directions along with real-time processing will constitute our future endeavours.

VII. Acknowledgement

This work was initially funded by Ericsson Magyarország Kft. and the High-Speed Networks Laboratory (HSNLab). Financial support from the National Research, Development, and Innovation Office (NKFIH, Grant No. K-146347) is acknowledged. The author is grateful for the fruitful discussions with Attila Báder, Alija Pašić, and Gergely Dobreff.

References

[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley, November 2000.

[2] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli, "Revealing skype traffic: when randomness plays with you," in *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 37–48. [Online]. Available: https://doi.org/10.1145/1282380.1282386

[3] P. Casas and S. Wassermann, "Improving qoe prediction in mobile video through machine learning," in *2017 8th International Conference on the Network of the Future (NOF)*, 2017, pp. 1–7.

[4] M. Darwich, "Machine learning technique predicting video streaming views to reduce cost of cloud services," in *2022 IEEE 8th World Forum on Internet of Things (WF-IoT)*, 2022, pp. 1–4.

[5] A. Biernacki, "Improving streaming video with deep learning-based network throughput prediction," *Applied Sciences*, vol. 12, no. 20, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/20/10274

[6] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," *SIGMETRICS Perform. Eval. Rev.*, vol. 33, no. 1, p. 50–60, jun 2005. [Online]. Available: https://doi.org/10.1145/1071690.1064220

[7] J. Park, H.-r. Tyan, and C.-c. J. Kuo, "Ga-based internet traffic classification technique for qos provisioning," in *2006 International Conference on Intelligent Information Hiding and Multimedia*, 2006, pp. 251–254.

[8] H. E. Dinaki, S. Shirmohammadi, E. Janulewicz, and D. Côté, "Forecasting video qoe with deep learning from multivariate time-series," *IEEE Open Journal of Signal Processing*, vol. 2, pp. 512–521, 2021.

TABLE III

THE PERFORMANCE (WEIGHTED ACCURACY, F1 SCORE, AND RECALL VALUES IN %) OF THE ADABOOST (AB), NEURAL NETWORK (NN), AND RANDOM FOREST (RF) CLASSIFIERS PREDICTING COMPREHENSIVE QOE VALUES WITH DIFFERENT WEIGHTS (w_B, w_L, w_S, w_F) TRAINED ON THE TWITCH DATASET. THE CLASSIFIERS WERE TRAINED ON 20 SVD VECTORS AND THE AGGREGATION INTERVAL WAS SET TO 3S.

w_B	w_L	w_S	w_F	AB	RF	NN	AB_F1	RF_F1	NN_F1	AB_Recall	RF_Recall	NN_Recall
1	2	1	1	52	63	63	39	63	62	35	63	63
1	2	1	2	54	66	66	48	66	63	47	67	69
1	2	2	1	73	75	75	73	75	75	73	75	75
1	2	2	4	81	86	86	81	86	86	81	86	86
1	3	3	1	81	84	83	81	83	82	81	84	83
1	3	4	2	74	81	82	63	80	80	62	81	82
1	4	4	1	46	63	66	43	63	63	47	64	65
1	5	1	5	87	86	86	87	86	86	87	86	86
2	1	1	2	77	83	85	74	84	85	71	85	87
2	2	1	3	57	63	61	42	63	63	40	64	66
2	2	5	2	71	83	83	69	83	82	72	84	82
3	5	4	1	90	91	90	90	91	90	90	91	90
4	5	4	3	79	80	80	63	80	81	67	80	81
5	1	1	4	74	81	80	67	81	81	63	83	83
5	1	2	4	55	80	80	57	81	80	63	82	82

[9] I. Bartolec, I. Orsolic, and L. Skorin-Kapov, "In-network youtube performance estimation in light of end user playback-related interactions," in *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, 2019, pp. 1–3.

[10] S. A. Wassermann, "Machine learning for network traffic monitoring and analysis: Application to internet qoe assessment and network security," PhD Thesis, TU Wien, Faculty of Electrical Engineering and Information Technology, 2022.

[11] I. Orsolic, D. Pevec, M. Suznjevic, and L. Skorin-Kapov, "Youtube qoe estimation based on the analysis of encrypted network traffic using machine learning," in *2016 IEEE Globecom Workshops (GC Wkshps)*, 2016, pp. 1–6.

[12] F. Y. Yan, H. Ayers, C. Zhu, S. Fouladi, J. Hong, K. Zhang, P. Levis, and K. Winstein, "Learning in situ: a randomized experiment in video streaming," in *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*. Santa Clara, CA: USENIX Association, Feb. 2020, pp. 495–511. [Online]. Available: https://www.usenix.org/conference/nsdi20/presentation/yan

[13] A. W. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," in *Proceedings of the 6th International Conference on Passive and Active Network Measurement*, ser. PAM'05. Berlin, Heidelberg: Springer-Verlag, 2005, p. 41–54. [Online]. Available: https://doi.org/10.1007/978-3-540-31966-5_4

[14] N. Barman, M. G. Martini, S. Zadtootaghaj, S. Möller, and S. Lee, "A comparative quality assessment study for gaming and non-gaming videos," in *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, 2018, pp. 1–6.

[15] G. Kougioumtzidis, A. Vlahov, V. K. Poulkov, P. I. Lazaridis, and Z. D. Zaharis, "Qoe prediction for gaming video streaming in o-ran using convolutional neural networks," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 1167–1181, 2024.

[16] I. Orsolic and L. Skorin-Kapov, "A framework for in-network qoe monitoring of encrypted video streaming," *IEEE Access*, vol. 8, pp. 74 691–74 706, 2020.

[17] M. Seufert and I. Orsolic, "Improving the transfer of machine learning-based video qoe estimation across diverse networks," *IEEE Transactions on Network and Service Management*, vol. 21, no. 3, pp. 2824–2836, 2024.

[18] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed., ser. Springer Series in Statistics. Springer New York, NY, 2009, eBook Packages Mathematics and Statistics, Mathematics and Statistics (R0). [Online]. Available: https://doi.org/10.1007/978-0-387-84858-7

[19] W. Zai-jian, Y.-n. Dong, H.-x. Shi, Y. Lingyun, and T. Pingping, "Internet video traffic classification using qos features," in *2016 International Conference on Computing, Networking and Communications (ICNC)*, 2016, pp. 1–5.

[20] T. Auld, A. W. Moore, and S. F. Gull, "Bayesian neural networks for internet traffic classification," *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 223–239, 2007.

[21] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. Cambridge: Cambridge University Press, 2012.

[22] I. Bartolec, I. Orsolic, and L. Skorin-Kapov, "Inclusion of end user playback-related interactions in youtube video data collection and ml-based performance model training," in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, 2020, pp. 1–6.

[23] I. Orsolic, M. Suznjevic, and L. Skorin–Kapov, "Youtube qoe estimation from encrypted traffic: Comparison of test methodologies and machine learning based models," in *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, 2018, pp. 1–6.

[24] G. Dobreff, M. Szalay, B. Ladóczki, M. Molnár, L. Varga, A. Báder, and A. Pašić, "Data collection framework for end-to-end radio and transport network quality monitoring," in *2023 15th International Conference on Quality of Multimedia Experience (QoMEX)*, 2023, pp. 127–130.

[25] J. Erman, A. Mahanti, and M. Arlitt, "Byte me: a case for byte accuracy in traffic classification," in *Proceedings of the 3rd Annual ACM Workshop on Mining Network Data*, ser. MineNet '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 35–38. [Online]. Available: https://doi.org/10.1145/1269880.1269890

[26] T. T. T. Nguyen and G. Armitage, "Synthetic sub-flow pairs for timely and stable ip traffic identification," *IEEE/ACM Transactions on Networking*, vol. 20, 01 2007.

[27] S. Zander, T. Nguyen, and G. Armitage, "Automated traffic classification and application identification using machine learning," in *The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05)l*, 2005, pp. 250–257.

[28] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "Blinc: multilevel traffic classification in the dark," in *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '05. New York, NY, USA: Association for Computing Machinery, 2005, p. 229–240. [Online]. Available: https://doi.org/10.1145/1080091.1080119

[29] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli, "Traffic classification through simple statistical fingerprinting," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 1, p. 5–16, jan 2007. [Online]. Available: https://doi.org/10.1145/1198255.1198257

[30] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, "Traffic classification on the fly," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 2, p. 23–26, apr 2006. [Online]. Available: https://doi.org/10.1145/1129582.1129589

[31] S. Huang, K. Chen, C. Liu, A. Liang, and H. Guan, "A statistical-feature-based approach to internet traffic classification using machine learning," in *2009 International Conference on Ultra Modern Telecommunications Workshops*, 2009, pp. 1–6.

[32] T. T. Nguyen and G. Armitage, "Training on multiple sub-flows to optimise the use of machine learning classifiers in real-world ip networks," in *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*, 2006, pp. 369–376.

[33] I. Akbari, M. A. Salahuddin, L. Ven, N. Limam, R. Boutaba, B. Mathieu, S. Moteau, and S. Tuffin, "A look behind the curtain:

Traffic classification in an increasingly encrypted web," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 5, no. 1, feb 2021. [Online]. Available: https://doi.org/10.1145/3447382

[34] P. Zhan, L. Wang, and Y. Tang, "Website fingerprinting on early quic traffic," *ArXiv*, vol. abs/2101.11871, 2021.

[35] Y. Shi, D. Feng, Y. Cheng, and S. Biswas, "A natural language-inspired multilabel video streaming source identification method based on deep neural networks," *Signal, Image and Video Processing*, vol. 15, pp. 1–8, 09 2021.

[36] S. Sen, O. Spatscheck, and D. Wang, "Accurate, scalable in-network identification of p2p traffic using application signatures," in *Proceedings of the 13th International Conference on World Wide Web*, ser. WWW '04. New York, NY, USA: Association for Computing Machinery, 2004, p. 512–521. [Online]. Available: https://doi.org/10.1145/988672.988742

[37] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[38] L. Al-Bakhat and S. Almuhammadi, "Intrusion detection on quic traffic: A machine learning approach," in *2022 7th International Conference on Data Science and Machine Learning Applications (CDMA)*, 2022, pp. 194–199.

[39] V. Paxson, "Bro: A system for detecting network intruders in Real-Time," in *7th USENIX Security Symposium (USENIX Security 98)*. San Antonio, TX: USENIX Association, Jan. 1998. [Online]. Available: https://www.usenix.org/conference/7th-usenix-security-symposium/bro-system-detecting-network-intruders-real-time

[40] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *2010 IEEE Symposium on Security and Privacy*, 2010, pp. 305–316.

[41] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *2017 International Conference on Information Networking (ICOIN)*, 2017, pp. 712–717.

[42] A. McGregor, M. Hall, P. Lorier, and J. Brunskill, "Flow clustering using machine learning techniques," in *Passive and Active Network Measurement*, C. Barakat and I. Pratt, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 205–214.

[43] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," in *Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data*, ser. MineNet '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 281–286. [Online]. Available: https://doi.org/10.1145/1162678.1162679

[44] N. Williams, S. Zander, and G. Armitage, "A preliminary performance comparison of five machine learning algorithms for practical ip traffic flow classification," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 5, p. 5–16, oct 2006. [Online]. Available: https://doi.org/10.1145/1163593.1163596

[45] ——, "Evaluating machine learning methods for online game traffic identification," *Swinburne University of Technology*, 1 2006. [Online]. Available: https://figshare.swinburne.edu.au/articles/report/Evaluating_machine_learning_methods_for_online_game_traffic_identification/26270206

[46] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld, "Gaming in the clouds: Qoe and the users' perspective," *Mathematical and Computer Modelling*, vol. 57, no. 11, pp. 2883–2894, 2013, information System Security and Performance Modeling and Simulation for Future Mobile Networks. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0895717711007771

[47] M. Lycett and O. Radwan, "Developing a quality of experience (qoe) model for web applications," *Information Systems Journal*, vol. 29, no. 1, pp. 175–199, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/isj.12192

[48] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang, "Developing a predictive model of quality of experience for internet video," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, p. 339–350, aug 2013. [Online]. Available: https://doi.org/10.1145/2534169.2486025

[49] S. Chaudhary, P. Sachdeva, A. Mondal, S. Chakraborty, and M. Maity, "Youtube over google's quic vs internet middleboxes: A tug of war between protocol sustainability and application qoe," 03 2022.

[50] D. Saif, C.-H. Lung, and A. Matrawy, "An early benchmark of quality of experience between http/2 and http/3 using lighthouse," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.

[51] T. Guarnieri, I. Drago, A. B. Vieira, I. Cunha, and J. Almeida, "Characterizing qoe in large-scale live streaming," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 2017, pp. 1–7.

[52] N. Barman and M. G. Martini, "Qoe modeling for http adaptive video streaming–a survey and open challenges," *IEEE Access*, vol. 7, pp. 30 831–30 859, 2019.

[53] T. Hoßfeld, M. Seufert, M. Hirth, T. Zinner, P. Tran-Gia, and R. Schatz, "Quantification of youtube qoe via crowdsourcing," in *2011 IEEE International Symposium on Multimedia*, 2011, pp. 494–499.

[54] F. Metzger, S. Geißler, A. Grigorjew, F. Loh, C. Moldovan, M. Seufert, and T. Hoßfeld, "An introduction to online video game qos and qoe influencing factors," *Commun. Surveys Tuts.*, vol. 24, no. 3, p. 1894–1925, jul 2022. [Online]. Available: https://doi.org/10.1109/COMST.2022.3177251

[55] D. K. Tapang, S. Huang, and X. Huang, "Qoe-based server selection for mobile video streaming," in *2020 IEEE/ACM Symposium on Edge Computing (SEC)*, 2020, pp. 435–439.

[56] B. Spang, B. Walsh, T.-Y. Huang, T. Rusnock, J. Lawrence, and N. McKeown, "Buffer sizing and video qoe measurements at netflix," in *Proceedings of the 2019 Workshop on Buffer Sizing*, ser. BS '19. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3375235.3375241

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

[58] F. Wamser, M. Seufert, P. Casas, R. Irmer, P. Tran-Gia, and R. Schatz, "Yomoapp: A tool for analyzing qoe of youtube http adaptive streaming in mobile networks," in *2015 European Conference on Networks and Communications (EuCNC)*, 2015, pp. 239–243.

[59] G. Gómez, L. Hortigüela, Q. Pérez, J. Lorca, R. García, and M. C. Aguayo-Torres, "Youtube qoe evaluation tool for android wireless terminals," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, p. 164, 2014. [Online]. Available: https://doi.org/10.1186/1687-1499-2014-164

[60] I. Orsolic, L. Skorin-Kapov, and T. Hoßfeld, "To share or not to share? how exploitation of context data can improve in-network qoe monitoring of encrypted youtube streams," in *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, 2019, pp. 1–3.

[61] C. Ge and N. Wang, "Real-time qoe estimation of dash-based mobile video applications through edge computing," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2018, pp. 766–771.

[62] T. Sharma, T. Mangla, A. Gupta, J. Jiang, and N. Feamster, "Estimating webrtc video qoe metrics without using application headers," in *Proceedings of the 2023 ACM on Internet Measurement Conference*, ser. IMC '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 485–500. [Online]. Available: https://doi.org/10.1145/3618257.3624828

[63] Y.-T. Lin, E. M. R. Oliveira, S. Ben Jemaa, and S. E. Elayoubi, "Machine learning for predicting qoe of video streaming in mobile networks," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[64] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," in *Machine Learning and Knowledge Discovery in Databases*, W. Buntine, M. Grobelnik, D. Mladenić, and J. Shawe-Taylor, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 254–269.

[65] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, Sep. 1936. [Online]. Available: https://doi.org/10.1007/BF02288367

[66] J. Piet, D. Nwoji, and V. Paxson, "Ggfast: Automating generation of flexible network traffic classifiers," in *Proceedings of the ACM SIGCOMM 2023 Conference*, ser. ACM SIGCOMM '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 850–866. [Online]. Available: https://doi.org/10.1145/3603269.3604840

[67] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine Learning*, vol. 39, no. 2/3, pp. 135–168, 2000.

[68] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, "Understanding the impact of video quality on user engagement," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, p. 362–373, aug 2011. [Online]. Available: https://doi.org/10.1145/2043164.2018478