DeepCrypt - Deep Learning for QoE Monitoring and Fingerprinting of User Actions in Adaptive Video Streaming

Pedro Casas, Michael Seufert, Sarah Wassermann, Bruno Gardlo, Nikolas Wehner, Raimund Schatz

DeepCrypt - Deep Learning for QoE Monitoring and Fingerprinting of User Actions in Adaptive Video Streaming

Pedro Casas*, Michael Seufert†, Sarah Wassermann*, Bruno Gardlo*, Nikolas Wehner†, Raimund Schatz*
*AIT Austrian Institute of Technology, †University of Würzburg

2022 IEEE 8th International Conference on Network Softwarization (NetSoft) | 978-1-6654-0694-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/NetSoft54395.2022.9844113

***Abstract*—We introduce *DeepCrypt*, a deep-learning based approach to analyze YouTube adaptive video streaming Quality of Experience (QoE) from the Internet Service Provider (ISP) perspective, relying exclusively on the analysis of encrypted network traffic. Using raw features derived on-line from the encrypted stream of bytes, *DeepCrypt* infers six different video QoE indicators capturing the user-perceived performance of the service, including the initial playback delay, the number and frequency of rebuffering events, the video playback quality and encoding bitrate, and the number of quality changes. *DeepCrypt* offers deep visibility into the behavior of the end-user, enabling the fingerprinting and detection of different user actions on the video player, such as video pauses and playback scrubbing (forward, backward, out-of-buffer), offering a complete visibility on the video streaming process from in-network traffic measurements. Evaluations over a large and heterogeneous dataset composed of mobile and fixed-line measurements, using the YouTube HTML5 player, the native YouTube mobile app, as well as a generic HTML5 video player built on top of open source libraries, and considering measurements collected at different ISPs, confirm the out-performance of *DeepCrypt* over previously used shallow-learning models, and its generalization to different video players and network setups.**

I. Introduction

Quality of Experience (QoE) monitoring is a daunting yet critical task for Internet Service Providers (ISPs), who need to shed light on the performance of their networks as perceived by their customers, to avoid churn due to quality dissatisfaction. Among the plethora of applications served through ISP networks, video streaming has attracted most of the attention in recent years. Video streaming is the most popular and most resource-demanding application of the Internet, due to the high number of users and video requests, high bit rates of the video content, and strict real-time requirements of the video playback. While ISPs have traditionally relied on the usage of Deep Packet Inspection (DPI) techniques to understand the performance of video applications from the network side, the wide adoption of end-to-end traffic encryption has drastically reduced their visibility, offering only little information about the traffic contents. This has motivated a surge in the research and conception of machine-learning based approaches to infer application-level QoE-metrics from the streams of encrypted bytes [1]–[5]. In these previous work, standard shallow-learning models have been used in the task.

In this paper we take a step further into this problem, by conceiving approaches based on novel, deep-learning architectures, which provide further visibility into the video streaming process. We conceive *DeepCrypt*, a deep-learning based approach to infer multiple video QoE indicators capturing the user-perceived performance of adaptive video streaming applications, relying exclusively on the analysis of raw features, derived from the encrypted network traffic. *DeepCrypt* provides deep visibility into the behavior of the end-user, enabling the detection of different user playback-related interactions, such as video pauses and playback scrubbing (forward, backward, out-of-buffer). Recent work in this direction [6] has shown that video QoE estimation becomes significantly more challenging when user player interactions are present in the video playback. To the best of our knowledge, this is the first paper explicitly addressing the combined inference of both video QoE metrics and user player interactions. Through extensive evaluation, we verify the out-performance of *DeepCrypt* as compared to previously employed shallow-learning models. In particular, we evaluate *DeepCrypt* using controlled measurements on: (i) multiple YouTube players – including the standard HTML5 YouTube player, the native mobile App, and a generic HTML5 player built on top of open libraries (video.js and dash.js) – (ii) multiple network setups in terms of downlink bandwidth, access technology – WiFi/LTE, and transport protocols (TCP and QUIC) – and (iii) different emulated player interactions, including pauses, forward, backward, and out-of-buffer scrubbing. We also evaluate *DeepCrypt* on measurements collected at operational ISP networks, and verify the generalization of the obtained results.

The remainder of the paper is structured as follows. Section II describes related work on video QoE monitoring from encrypted traffic measurements. Section III presents the principles, raw features, and deep architecture behind *DeepCrypt*, and describes the different datasets used for model training and evaluation purposes. Extensive evaluation results are reported in Section IV, including the estimation of video QoE metrics, the generalization of results to ISP measurements, and the detection of user actions. Section V concludes this work.

II. Related Work

The wide adoption of TLS/HTTPS has motivated a vast literature in the problem of adaptive video streaming QoE monitoring from network encrypted traffic, mainly relying on machine learning models. First approaches [2] considered shallow machine learning-based architectures to estimate YouTube QoE for full video sessions (i.e., once the video session has completed) using features derived from packet sizes, inter-arrival times, and throughput measurements. A similar approach was presented in [1], where authors rely on shallow learning models and measurements in cellular net-

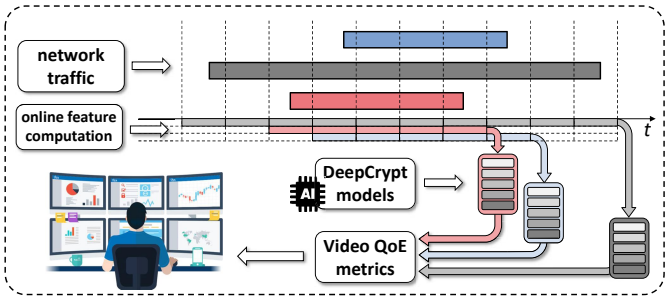


Fig. 1: *DeepCrypt* video QoE and user actions monitoring workflow. Raw traffic features are computed on-line for each detected video session.

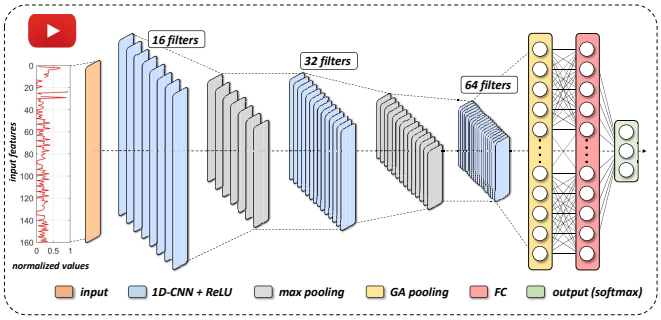


Fig. 2: *DeepCrypt* deep learning architecture.

works to estimate standard QoE indicators for adaptive video streaming services (e.g., video resolutions, rebuffering events), based on more elaborated features such as round-trip times, packet losses, and video chunk sizes – note that this approach requires explicit chunk-detection mechanisms, which are error prone and add complexity to the measurement process. Real-time analysis of YouTube QoE through shallow learning was first introduced in [3], where rebuffering detection and binary low/high video resolution classification are computed for consecutive time windows of a few seconds duration. Other papers have tackled the problem by modeling the video player, in particular by inferring the buffered playtime [7], [8]. Recent papers also relying on shallow learning for real-time video QoE inference include Requet [4] and our previous system, ViCrypt [5]. While both systems provide estimates for multiple video QoE metrics every few seconds, both require complex feature extraction from the stream of packets, and in particular Requet requires chunk-detection. In addition, both systems neglect the identification of user player interactions, which might have a direct impact on the estimation and generalization performance of the learning models, as recently shown [6].

Different from previous work, *DeepCrypt* avoids complex feature computations by relying on deep learning models to automatically construct feature representations from raw input metrics, derived from packets sizes and timestamps. *DeepCrypt* raw features are computed on-line during an on-going YouTube video session, using constant memory space to enable traffic monitoring. At the end of the video session, *DeepCrypt* outputs multiple QoE metrics describing the performance of the video playback. In addition, *DeepCrypt* detects the occurrence of different user player interactions, offering as such an unprecedented visibility on the complete video playback process.

III. *DeepCrypt* Model and Datasets

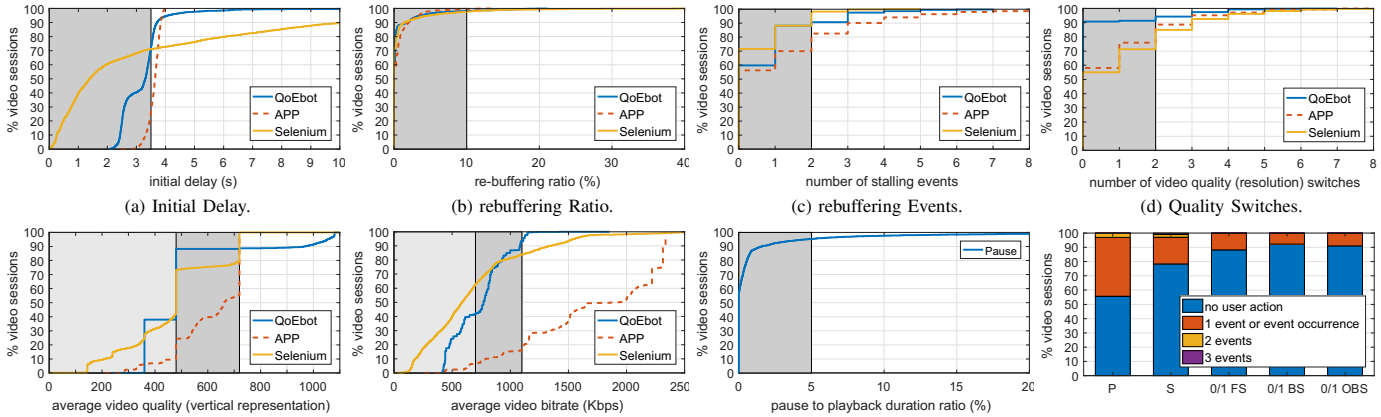
A. Features and Deep Model

Figure 1 depicts the complete *DeepCrypt* video QoE monitoring workflow. Raw traffic features are computed on-line for each independent YouTube video session. In a nutshell, we identify video flows through DNS-based IP addresses to domain names dynamic mappings, following our own work on Web QoE monitoring [10]. While the specific video session detection over encrypted traffic is in itself a very complex problem [10], we deem it out of the scope of this paper. When a video session completes, multiple independent *DeepCrypt* models, each one trained for a specific classification task, are applied on the resulting input features, inferring both video QoE metrics and user actions occurred during the video playback. A total of 160 raw features are extracted from the ongoing encrypted stream of bytes, in an on-line manner. These include simple metrics such as the duration of the session, number of packets, number of bytes, average throughput, and packet inter-arrival times, separately computed for downlink and uplink. Features are computed for the complete session, as well as for consecutive time-windows of 1, 5, and 10 seconds – the rationale here is to capture different phenomena visible at different time scales. These time-window metrics are summarized through the observed minimum, average, maximum, as well as their variance and standard deviation, across the complete video session. All computations are done on-line and recursively, in fixed-size memory space, relying on simple algorithms for one-pass efficient computation [11].

In terms of learning model, Figure 2 describes the underlying deep architecture used by *DeepCrypt*. A series of three consecutive 1D-CNN convolutional layers with 16, 32, and 64 filters respectively – using standard ReLU activation – and intermediate max pooling layers – to compress the size of the representations – form the core of the representation learning stage. The last convolutional layer is connected to a Global Average (GA) pooling layer, which adds additional robustness against over-fitting. The classification stage is composed of a series of standard fully connected layers, using softmax as activation function on the last fully connected layer. The architecture additionally considers batch normalization and dropout layers, to regularize the model.

B. YouTube Adaptive Streaming Datasets' Collection

To study the performance of *DeepCrypt*, we rely on an assorted list of five different datasets we collected back in 2018/2019 for the YouTube HTTP adaptive video streaming service. Table I summarizes these datasets in terms of number of video sessions, specific video player, type of emulated user actions, and measurement environment. While the data is rather outdated, the total number of videos and the heterogeneity of vantage points offers an unparalleled catalog for the study, and in particular to obtain a highly generalizable model.



(a) Initial Delay. (b) rebuffering Ratio. (c) rebuffering Events. (d) Quality Switches.

(e) Avg. Video Resolution. (f) Avg. Video Bitrate. (g) Pause time ratio. (h) Player Scrubbing.

Fig. 3: Characterization of the different datasets used for model training and validation. The empirical distributions of the proposed video QoE metrics and user player actions, along with the underlying classes, are depicted. For some specific metrics, classes are highly imbalanced.

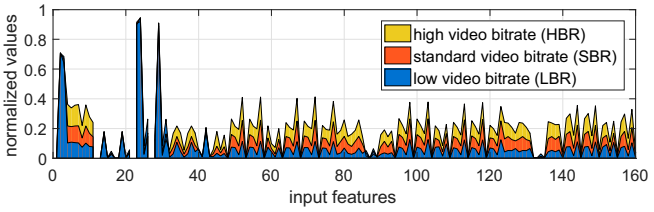


Fig. 4: Average feature vector values for video bitrate metric.

dataset	# samples	player	user actions	environment
QoEbot	4,000	HTML5	P	lab
APP	2,000	APP	no	lab
Selenium	8,000	YT-HTML5	P/S	field
ISP-A	1,000	YT-HTML5	no	field
ISP-B	800	YT-HTML5	no	operational

TABLE I: Heterogeneous datasets for model training and evaluation.

In addition, even if *DeepCrypt*'s underlying model might need re-training for deployment in operational conditions today, the main properties, approaches, and findings of this study remain valid, mainly because the core principles behind HTTP adaptive streaming are still in place.

The total data includes traffic captures for more than 15,000 individual YouTube video sessions. Two of these datasets (*QoEbot* and *Selenium*) correspond to measurements collected on a standard laptop with two different instrumented video players – using Chrome as browser: the first one is a custom-built, fully controllable, and generic HTML5 video player built on top of open source libraries, using in particular video.js and dash.js libraries; the second one corresponds to the standard HTML5 YouTube video player, instrumented through the application of Selenium browser-automation libraries. We streamed and measured more than 12,000 randomly selected YouTube video sessions between June 2018 and February 2019, taking heterogeneous network setups to improve generalization. Video sessions were collected at fixed-line networks (~20% of the videos), WiFi networks (~60%), and LTE mobile networks (~20%). Both QUIC (~45%) and TCP (~55%) were considered as transport protocols. To induce QoE degradation, bandwidth limitations were imposed on some of the sessions. The *QoEbot* dataset was collected in a fully controlled environment (lab), and the *Selenium* dataset was collected in the field, using our own instrumented laptops at different geographical locations. The *APP* dataset corresponds to an open dataset we collected in 2018 [9], including measurements from the native, mobile Android YouTube app. The last two datasets correspond to measurements collected at operational ISPs in 2019 using their own in-network traffic monitoring technology, relying on instrumented end-points for field trials (*ISP-A*) and real user monitoring (*ISP-B*).

Figure 3 depicts the empirical distributions of the target video QoE metrics, including (a) initial playback delay (ID), (b) rebuffering frequency (RF), (c) rebuffering events (R), (d) number of video quality (resolution) changes (QS), (e) average video quality or resolution (D), and (f) average video (encoding) bitrate (BR). The distribution of user actions introduced in some of the sessions are also reported, including (g) the Pause (P) time ratio – total pause time over video duration, and (h) multiple player Scrubbing actions (S): Forward-scrubbing (FS), Backward-scrubbing (BS), and out-of-buffer-scrubbing (OBS), the later referring to a player scrubbing action which results in additional video requests. All values are discretized into classes (labels), treating the analysis as multiple independent classification tasks. Classes include (a) high/low initial delay (HID/LID) – threshold T_h = 3.5 seconds, (b/c) no/low/high rebuffering (NR/LR/HR) – T_h = 10% (for RF) or 2 rebuffering events (for number of R events), (d) no/low/high quality changes (NQS/LQS/HQS) – T_h = 2 changes, (e) low/standard/high definition (LD/SD/HD) – T_h = 480p and 720p, (f) low/standard/high video bitrate (LBR/SBR/HBR) – T_h = 700 and 1100 $kbps$, (g) no/low/high pause ratio (NP/LPR/HPR) – T_h = 5% and (h) number and detection of scrubbing events (NS) – 0 to 3 events.

IV. *DeepCrypt* Evaluation

A. *Video QoE Estimation*

We firstly evaluate the video QoE estimation performance of *DeepCrypt*. We build a combined dataset D_3 = {*QoEbot*

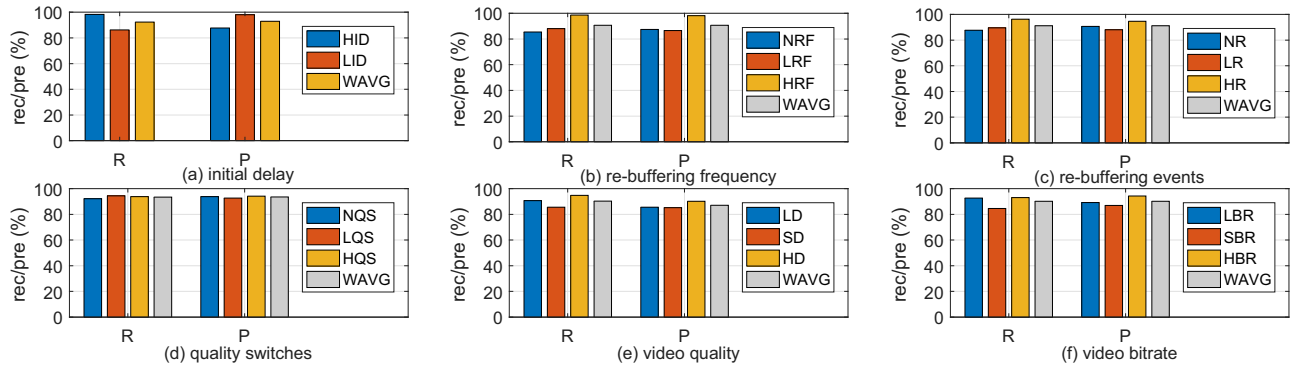


Fig. 5: *DeepCrypt* video QoE analysis. Recall and precision for cross-validation on semi-controlled dataset $D_3 = \{QoEbot \cup APP \cup Selenium\}$.

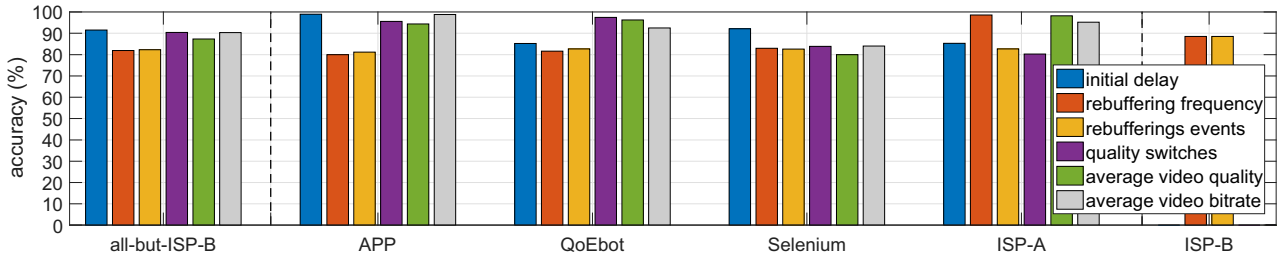


Fig. 6: *DeepCrypt* generalization. Classification accuracy for *all-but-ISP-B* D_4 dataset, and application to individual datasets, including out-of-training *ISP-B*.

$\cup$ *APP* $\cup$ *Selenium*}, using the three controlled-measurement datasets (cf. Figure 3). Unless said otherwise, evaluation results correspond to 5-fold cross validation. Figure 5 reports the obtained results in terms of recall and precision per class, including a weighted-average (WAVG) for each of the metrics as indicative overall performance. Recall and precision are high, above 85% for all classes and all six metrics, and close to 95% for WAVG results in all video QoE metrics. These results suggest that class differences are properly reflected at the input features, and that *DeepCrypt* is able to properly track those differences. To showcase the different fingerprints in the input features, Figure 4 displays the average feature vector values (normalized) for the three video bitrate classes LBR/SBR/HBR, taking all samples in D_3. Similar differences are observed for the other video QoE metrics, partially explaining *DeepCrypt*'s excellent performance.

B. Generalization to ISP Measurements

We now take a look at the generalization of results, exploiting the heterogeneity of the measurements. We consider a combined dataset including all-but-ISP-B measurements, referred to as $D_4 = \{QoEbot \cup APP \cup Selenium \cup ISP\text{-}A\}$. We take 80% of D_4 for training and (cross)validation purposes, and the remaining 20% for testing. Figure 6 reports the obtained results per video QoE metric, taking the overall model accuracy as aggregated performance metric. On the left side (all-but-ISP-B), results correspond to 5-fold cross validation on the training dataset. Results are similar to those obtained in D_3 (cf. Figure 5); however, note how accuracy drops to about 80% for both rebuffering frequency and number of events, suggesting that *ISP-A* measurements add significant complexity to the estimation problem. Moving to the right-side, we take the *DeepCrypt* models trained on the 80% of D_4 and apply them to the remaining 20% of the data, reporting results for each dataset individually (*APP*, *QoEbot*, *Selenium*, and *ISP-A*). Performance generalizes properly across all different datasets, showing in general higher accuracy for video quality related metrics and playback delay, and lower performance for rebuffering related metrics, above 80% for all the different scenarios. Finally, on the right-side end we show the performance of these models on the *ISP-B* dataset, which was not part of the training. We refer to this as an out-of-distribution evaluation. Note that we only have rebuffering metrics as ground truth data for *ISP-B* measurements. Again on this scenario, *DeepCrypt* evidences proper model and performance generalization.

C. Deep vs. Shallow Learning Benchmarking

The next question we investigate is whether the usage of a deep learning architecture results in better performance, as compared to standard shallow learning models. To the best of our knowledge, previous work has relied exclusively on shallow learning models for the estimation tasks. In particular, Random Forest (RF) models are the most popular ones used in the state of the art [1]–[6]. We therefore compare *DeepCrypt* against five shallow-learning models, using the same 160 input features. These models include a standard Naïve Bayes classifier (NB), a 3-layers feed-forward Neural Network (NN), a Support Vector Machines (SVM) classifier, a CART Decision Tree (DT), and a Random Forest model (RF). Figure 7 shows the benchmark results on D_4, for each individual video QoE metric – results correspond to 5-fold cross validation. *DeepCrypt* systematically outperforms the shallow learning models for all six video QoE metrics. As reported in the

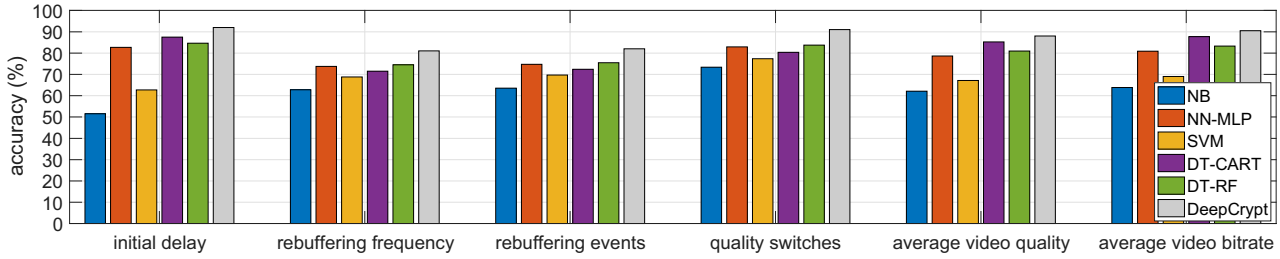


Fig. 7: Overall classification accuracy of *DeepCrypt* vs. shallow learning models, using *all-but-ISP-B* D_4 dataset.

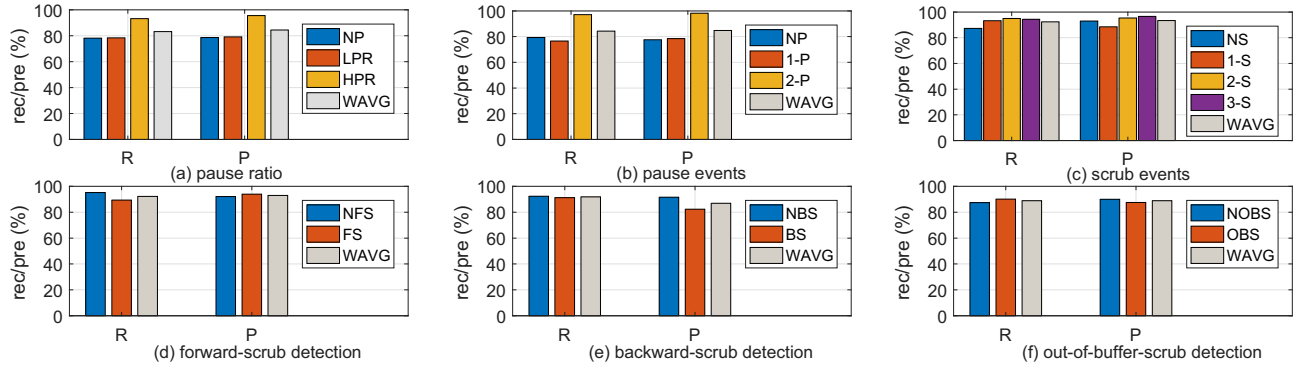


Fig. 8: *DeepCrypt* detection and classification of user player interactions, for a combined $D_2 = \{QoEbot \cup Selenium\}$ dataset.

state of the art, the RF model provides the best performance among the shallow learning models; nevertheless, *DeepCrypt* improves over the RF model with about 10% higher accuracy in all metrics.

D. Identification of User Actions

The last evaluation considers the estimation of user player actions. We take a combined $D_2 = \{QoEbot \cup Selenium\}$ dataset, as these are the only datasets containing user player interactions – cf. Table I. The *QoEbot* dataset contains only pause events (P), located at random positions within the video playback, whereas the *Selenium* dataset contains both pauses and three kinds of player scrubbing events FS/BS/OBS, also occurring at random positions. Figure 8 presents the results obtained by 5-fold cross validation on D_2. Classification performance is outstanding, with values around or above 80% for all classes and player actions. We use the term outstanding due to the complexity of the classification task, as already discussed in previous work [6]; one of the major challenges faced in this classification task is to properly discriminate between user-induced player interactions and video streaming events, such as rebufferings, as both generate similar outcomes in the playback stream. Note that both *QoEbot* and *Selenium* datasets contain user actions and QoE degradation events. Interestingly, performance is higher for detection and classification of player scrubbing actions and lower for pause events, pointing to the aforementioned observations.

V. Concluding Remarks

Using a deep learning architecture, *DeepCrypt* is a novel approach to analyze YouTube adaptive video streaming QoE, from the analysis of encrypted network traffic. Through extensive empirical evaluation on five heterogeneous HTTP video streaming datasets we have shown that: (i) *DeepCrypt* can infer six different video QoE indicators and detect four different user player actions with high precision and recall, providing as such an unprecedented level of visibility on the video streaming process from the analysis of encrypted traffic; (ii) the underlying deep learning model outperforms shallow learning models previously used in the literature for similar classification tasks; and (iii) *DeepCrypt* results properly generalize to multiple different video players, devices, network setups, and operational networks. *DeepCrypt* results offer a promising venue for deep learning models applied to video QoE monitoring and analysis.

References

[1] G. Dimopoulos et al., "Measuring Video QoE from Encrypted Traffic," in *ACM Internet Measurement Conference (IMC)*, 2016.
[2] I. Orsolic et al., "YouTube QoE Estimation Based on the Analysis of Encrypted Network Traffic Using Machine Learning," in *IEEE QoEMC Workshop*, 2016.
[3] M. H. Mazhar et al., "Real-time Video Quality of Experience Monitoring for HTTPS and QUIC," in *IEEE INFOCOM Conference*, 2018.
[4] C. Gutterman et al., "Requet: Real-time QoE Detection for Encrypted YouTube Traffic," in *ACM Multimedia Systems Conference (MMSys)*, 2019.
[5] S. Wassermann et al., "ViCrypt to the Rescue: Real-time, Machine-Learning-driven Video-QoE Monitoring for Encrypted Streaming Traffic," *TNSM*, vol. 17 (4), 2020.
[6] I. Bartolec et al., "Inclusion of End User Playback-Related Interactions in YouTube Video Data Collection and ML-Based Performance Model Training," in *International Conference on Quality of Multimedia Experience (QoMEX)*, 2020.
[7] V. Krishnamoorthi et al., "BUFFEST: Predicting Buffer Conditions and Real-time Requirements of HTTP(S) Adaptive Streaming Clients," in *MMSys*, 2017.
[8] T. Mangla et al., "eMIMIC: Estimating HTTP-based Video QoE Metrics from Encrypted Network Traffic," in *TMA Conference*, 2018.
[9] T. Karagkioules et al., "A Public Dataset for YouTube's Mobile Streaming Client," in *Workshop on Mobile Network Measurement (MNM)*, 2018.
[10] P. Casas et al., "X-Ray Goggles for the ISP: Improving in-Network Web and App QoE Monitoring with Deep Learning," in *TMA Conference*, 2022.
[11] P. Pébay, "Formulas for Robust, One-Pass Parallel Computation of Covariances and Arbitrary-Order Statistical Moments," Sandia National Labs, Tech. Rep., 2008.