

ABR prediction using supervised learning algorithms

Hiba Yousef, Jean Le Feuvre, Alexandre Storelli

► **To cite this version:**

Hiba Yousef, Jean Le Feuvre, Alexandre Storelli. ABR prediction using supervised learning algorithms. IEEE international workshop on multimedia signal processing, Sep 2020, Tampere, Finland. hal-02949449

HAL Id: hal-02949449

<https://hal.archives-ouvertes.fr/hal-02949449>

Submitted on 25 Sep 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ABR prediction using supervised learning algorithms

Hiba Yousef ^{*†}, Jean Le Feuvre ^{*} and Alexandre Storelli [†]
^{*}*LTCI, Telecom Paris, Institut Polytechnique de Paris, Paris, France*
[†]*Streamroot, Paris, France*
{hiba.yousef, alexandre.storelli}@centurylink.com
{hiba.yousef, jean.lefeuvre}@telecom-paris.fr

Abstract—With the massive increase of video traffic over the internet, HTTP adaptive streaming has now become the main technique for infotainment content delivery. In this context, many bandwidth adaptation algorithms have emerged, each aiming to improve the user QoE using different session information e.g. TCP throughput, buffer occupancy, download time... Notwithstanding the difference in their implementation, they mostly use the same inputs to adapt to the varying conditions of the media session. In this paper, we show that it is possible to predict the bitrate decision of any ABR algorithm, thanks to machine learning techniques, and supervised classification in particular. This approach has the benefit of being generic, hence it does not require any knowledge about the player ABR algorithm itself, but assumes that whatever the logic behind, it will use a common set of input features. Then, using machine learning feature selection, it is possible to predict the relevant features and then train the model over real observation. We test our approach using simulations on well-known ABR algorithms, then we verify the results on commercial closed-source players, using different VoD and Live realistic data sets. The results show that both Random Forest and Gradient Boosting achieve a very high prediction accuracy among other ML-classifier.

Index Terms—HTTP Adaptive Streaming, Machine Learning, Classification, P2P

I. INTRODUCTION

Video consumption is dominating the Internet traffic and this has been reported by CISCO to reach 82% of global Internet traffic by 2022 [1]. Achieving a good user Quality of Experience (QoE) has become more important than ever, and although many solutions have been proposed, Adaptive BitRate streaming (ABR) has shown to be the best technique for video delivery over the Internet. In ABR, the media (video, audio) is divided into small parts of roughly constant duration, called segments, and each segment is encoded into different qualities. A client-side adaptive bitrate algorithm is then used to decide on the most convenient quality given the network conditions. In this sense, many ABR algorithms have been proposed, either in the literature or in real open or closed video player deployments. Even if they differ in their deep decision logic, most of the state-of-the-art ABR algorithms rely on heuristic observations as inputs to optimize the bitrate selection of the next segments. These inputs are usually the bandwidth measurements (e.g. TCP throughput and

the download time as seen by the application-layer), the buffer dynamics (e.g. the buffer occupancy and the maximum buffer size mainly), the segment characteristics (e.g. size, duration, and encoding bitrate) and in some cases the device capabilities (e.g., CPU usage, memory, playback speed). Authors in [2] provide an intensive survey on different ABR schemes. Interestingly, they show that the majority of the client-based adaptation schemes rely on bandwidth and/or buffer heuristics. Besides, few algorithms use information about the segment characteristics and device capabilities.

Nowadays Machine Learning (ML) and other control techniques have shown to be promising in many research fields, and video streaming is not an exception. In the context of enhancing the video streaming, ML algorithms have been used in different approaches like CDN caching [3], ML-based ABR solutions [4], video traffic classification [5] and of course the wide application of ML-based video coding [6].

Supervised learning (SL) consists in learning the relation between an input and an output based on previous examples of input-output pairs. SL enables learning from past observations to predict future events, which is the main task of ABR. Therefore, mixing these two approaches is a logical choice. ABR prediction, *Rate Labeling* as referred, was first used in MLASH algorithm [7]. MLASH exploits additional features to improve the prediction accuracy and the QoE of some existing heuristic ABR algorithms. Likewise, authors in [8] used similar Rate Labeling logic, to predict and enhance Scalable Video Coding-based bitrate algorithms. However, both works require some prior knowledge about the ABR algorithm used, and although they perform very good on trace-based evaluation, we are not aware of test reports for real-life VoD and Live scenarios.

Predicting the ABR behavior is important for pre-fetching and caching-based systems. In this direction, there have been some attempts to improve the CDN pre-fetching by informing the CDN about the next segment to be requested, as described in the emerging specification Common-Media-Client-Data (CMCD). However, in the current ABR implementations, the player makes sequential requests only, yet it is not clear how to decide on the ABR decision for the next segments. Additionally, in P2P streaming, a pre-fetcher usually downloads the video segments ahead of the player requests. The current pre-fetching techniques rely on the last requested quality by

the video player to fetch the upcoming videos, thus, in the case of quality switches, this leads to discarding the pre-fetched segments on the previous quality, and requesting new list of segments. Knowing the ABR decisions in advance would help in anticipating the track switches and hence fetching next segments on the predicted quality.

In the current implementations of P2P streaming, the video player is integrated on top of the P2P stack which replaces the HTTP stack as a transport layer. The ABR, in most of the web and native players, is completely unknown to the P2P stack, which makes the P2P-ABR integration more problematic. For this reason, in this paper, we deliberately try to predict any ABR algorithm given the most frequently used input features only, and regardless of the ABR used.

To this end, we choose some of the well-known supervised classifiers to predict the bitrate decision of the ABR algorithm. To justify that our work is ABR agnostic, we first test the models on six well-known state-of-the-art ABR algorithms. Then, using real datasets collected from both VoD and Live sessions, we predict the bitrate decision of three commercial, completely unknown and closed-source algorithms.

The rest of this paper is organized as follows: Section II presents the related work. The bitrate selection is formulated as a classification problem in Section III. The experimental setup is described in Section IV. Section V discusses the performance of the different classifiers. We further detail the application of this work to P2P streaming in Section VI, and finally, the paper is concluded in Section VII.

II. RELATED WORK

A. Adaptive bitrate classifications

Adaptive bitrate algorithms are extensively classified in different prior works [2] [9]. Interestingly, most of these works agree on classifying the ABR algorithms by their required inputs into three main classes: buffer-based, throughput-based, and hybrid-buffer-throughput-based. Recently, ML and control techniques have been used a lot in ABR, leading to a new emerging ML-based and control-based class. In this work, we picked some of the most famous ABR algorithms: BBA [10] and BOLA [11] from buffer-based class, PANDA and CONVENTIONAL [12] and Festive [13] from throughput-based class, and RobustMPC [14] from control-based class. All these algorithms are well-known in the adaptive video streaming domain, and they are frequently used in the literature.

B. Supervised learning for classification problems

Models in SL are trained on labeled datasets with the perspective of using models on yet unknown data. SL involves classification, regression and also structure output predictions. In this work we focus on classification, and depending on the number of the classes, the classification problems are divided into two main types: Binary and Multiclass problems. Some works like [15] divide Supervised classifiers into: (1) Statistical learning techniques, e.g. Naive Bayes. (2) Logic-based techniques, e.g. Decision Tree [16], Random Forest

[17], Adaboost [18] and Gradient Boost [19]. (3) Instance-based techniques, e.g. K-Nearest Neighbors [20] algorithm. (4) Support vector machine (SVM) [21].

C. ML-based classification in adaptive streaming

As previously mentioned, the research in adaptive streaming is shifting towards machine learning and optimization control. ML and ABR intersect at the point of learning from heuristic observations and predicting future decisions. In this context, some prior works tackled the adaptation problem from another point of view, by not proposing a new algorithm from scratch, but improving the existing ones using ML algorithms. In their work [7], authors proposed a model, called MLASH, that uses an additional set of features to train a random-forest classifier to improve the prediction accuracy of some ABR algorithms. However, MLASH uses some information, *True information* as referred, from the ABR algorithms. Besides, we are not aware of any work that implements MLASH on the control-based ABR or any experiments with realistic VoD and Live datasets. A similar work [8] uses Long Short Term Memory (LSTM) to predict a client-side Scalable Video Coding-based bit-rate adaptation using a set of heuristic attributes (e.g., buffer and throughput variables.). Although they compare different ML-algorithms and perform good trace-based evaluation, the proposed model tackles only the SVC-based algorithms, and we are not aware of any work that extends to the classic heuristic and control-based ABR algorithms.

III. BITRATE SELECTION CLASSIFICATION PROBLEM

ABR bitrate prediction is formulated as a multiclass classification problem in this work. The first step of solving this ML problem is to define the features needed to predict the ABR bitrate. As stated earlier in the introduction, most ABR algorithms use a set or a subset of network, buffer, or segment variables. In this study, and to keep the model generic, we will use the following set of features:

- 1) Buffer Level (s): the buffer occupancy when requesting the segment, which is usually available on most video players' public APIs.
- 2) Bandwidth (bps): the TCP throughput as seen by the application layer after downloading the segment, which is simply measured by computing the data downloaded (segment size here) over the download time.
- 3) Previous Bandwidth (bps): the throughput measured when downloading the previous segment; this information is used in different algorithms for smoothing the bandwidth estimation, or when the current estimation is not available as is the case with some buffer-based algorithms.
- 4) Download Time (s): the segment download time.
- 5) Previous Bitrate (bps): the bitrate of the previously selected segment, as used by the majority of ABR algorithms to take the bitrate smoothness into account.

Some other features can be used in the bitrate selection, as RTT, moving average bandwidth over n segments, rewards when ML-based or control-based ABR are used, processing

power, packet loss, some network-assisted information, etc. However, some of these attributes require some information about the ABR algorithm beforehand (e.g., the value of n and the *reward*), which is not compatible with the main goal of this work of being ABR agnostic. Moreover, other metrics are strongly dependent on the device capabilities, and generalizing these metrics over different users is of our interest in future work.

As our features have a wide range of values, we rescale the above-mentioned features, using Min-Max scaler where all features will be transformed into the range $[0,1]$. Then, we build our dataset matrix of pairs of M inputs (features) and output (label) over N instances (video segments in our problem). For training and testing, we perform stratified K-Folds Cross-Validation with $K = 10$, where the data is further split into K different subsets (or folds). The folds are made by preserving the percentage of samples for each class. Then, $k - 1$ folds are used to train the model whereas the subset k is left as test data. The model results are then averaged against each of the folds and tested after against the Test set.

IV. EXPERIMENTAL EVALUATION

In this work, we use a wide set of datasets from simulation and real-life cases.

For simulation, we used six classic ABR algorithms: BBA [10], BOLA [11], CONVENTIONAL [12], PANDA [12], FESTIVE [13] and Robust MPC [14]. We implemented all these algorithms using a MATLAB-based model presented in [22], except for Robust MPC, for which we used an existing Python-based implementation that is available at Pensieve Github¹. The stats were collected from 10 users who were watching the same content and experiencing different bandwidth profiles. The bandwidth profiles were chosen from some publicly available sets of real 3G [23] and 4G [24] bandwidth traces, so that the performance could be evaluated in highly dynamic adaptive scenarios. For the video content, we chose Red Bull Play Streets video from DASH Dataset Sequences², one-hour duration, segmented into 2-second segments and encoded at six different bitrates: 0.3, 0.7, 1.19, 1.99, 2.99 and 4.981 Mbps.

For realistic datasets, and to validate our work with unknown ABR algorithms, we collected the stats from commercial streaming services using STREAMROOT technology providing the P2P backend. The ABR algorithms for these players are enabled to work in P2P using Response-Delay approach presented in [22]. The trials were conducted with various HTML5 video players implementing their own, different ABR algorithms, where some information like the buffer level can be accessed through DOM independently from the ABR implementation. Three commercial service providers, referred as $S1, S2, S3$, were chosen randomly, and for each, the data was collected from an overall range of 5k to 10k concurrent peers. The service provider used different live and

VoD streams, so we built two datasets, one for VoD and one for Live sessions, each with up to 40000 instances. The datasets have shown imbalance bitrate selection for video segments, dominating the highest bitrate, i.e. the bitrate distribution, from the lowest to the highest, is 4, 6, 9, 11, 70 % and 6, 7, 14, 73% for the dataset collected from $S1$ which uses 5 qualities for Live and four for VoD. The second dataset from $S2$ shows using four qualities for Live and three for VoD with distribution of 7, 17, 28, 48% and 12, 14, 74% (Live and VoD respectively). Also, the last dataset related to $S3$ shows a bitrate distribution of 1, 1, 2, 4, 4, 6, 82% and 1, 1, 2, 5, 6, 7, 78 % using seven qualities for Live and VoD respectively.

We performed our experiments over the ML algorithms presented in section II-B: Logistic Regression (LGS), Support Vector Machine (SVM), Random Forest (RF), Decision Tree (DT), Ada Boost (AdBst), Gradient Boost (GrdBst), Naive Bayes (NB), K-Nearest Neighbors (KNN). For their implementation and measurement, we used Scikit-Learn package [25] on a Linux machine (Ubuntu 20.04 LTS), with processor Intel® Core™ i7-8665U, running 8 cores at 1.90GHz with 16 GB RAM.

For evaluation metrics, it is common in classification problems to use some known metrics (e.g. accuracy, precision, recall) [26] to evaluate models. Then for multi-class problems, these metrics are averaged per class. Here, we use the weighted-averaging method due to the imbalance of our datasets and we provide the confusion matrix to show the per-class prediction accuracy.

V. RESULTS AND DISCUSSION

In this section, we present the ABR prediction results, first with simulation datasets from the selected ABR algorithms and then from the realistic datasets from unknown ABR algorithms.

A. Simulation-based datasets

1) *Feature importance*: The feature importance of the ABR algorithms used is presented in Fig. 1. The previous bitrate is noticeably dominating the decision of the selected ABR algorithms, this acknowledges the importance of the smoothness in these algorithms. Additionally, we can infer the nature of the ABR: the buffer-based algorithms (BBA and BOLA) show high importance of the buffer level feature over the network features, while the throughput-based algorithms (CONVENTIONAL, PANDA, and FESTIVE) give higher importance to the bandwidth and the download time features. Finally, we can see that R-MPC, which is a hybrid control-based algorithm, relies on all the selected features (buffer and network features) with different, but close, percentages.

2) *Metrics evaluation*: The prediction accuracy of the different machine learning classifiers is shown in TABLE I. Among the selected ML classifiers, RF and GrdBst achieve the highest prediction accuracy for all the selected ABR algorithms. For BBA and BOLA, this accuracy is more than 99% with GrdBst scoring slightly higher than RF (around 99.9%). With the throughput-based algorithms, we notice a

¹ Available at: <https://github.com/hongzimao/pensieve>

² Available at: <https://dash.itec.aau.at/download/>

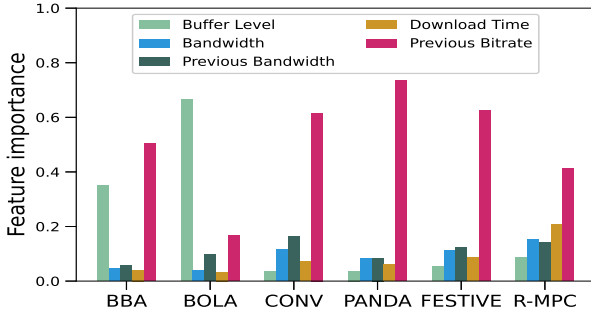


Fig. 1: Feature importance for the selected ABR algorithms

small decline in the overall accuracy for both RF and GrdBst which still show close scores (96-97%). For control-based algorithms, represented by *R-MPC*, it is noticeable that GrdBst scores the best accuracy with slightly higher than 98%.

TABLE I: Classification accuracy (%) of ML classifiers on different ABR datasets

	LGS	SVM	RF	DT	GrdBst	AdBst	NB	KNN
BBA	46.97	87.71	99.63	99.60	99.82	41.36	87.55	85.84
BOLA	53.34	71.23	99.88	99.969	99.97	61.13	63.91	70.01
CONV	44.86	93.29	96.38	94.83	96.23	67.63	93.01	91.79
PANDA	41.05	97.22	98.37	95.79	97.37	65.95	97.13	95.42
FESTIVE	36.61	93.04	96.54	94.34	96.23	63.03	92.46	91.79
R-MPC	39.59	32.35	96.97	94.05	98.01	70.64	80.11	52.47

Moving to TABLE II, which presents the precision and recall metrics achieved with RF and GrdBst only, due to the low accuracy of the other algorithms. The best scores are obtained for buffer-based algorithms, with RF reaching a bit higher than 99.8% (for both BBA and BOLA), while GrdBst reaches the same percentage with BBA and even better with 99.98% for BOLA. Unfortunately, this good behavior drops a bit with throughput-based algorithms: both RF and GrdBst score around 96% for CONVENTIONAL and FESTIVE. For PANDA, RF achieves 1% more than GrdBst (98% to 97%). However, with Robust MPC, the performance is reversed: GrdBst is better than RF with 98.4 to 97.5%.

TABLE II: Precision and Recall(%) of RF and GrdBst classifiers for simulation datasets

	RF		GrdBst	
	Precision	Recall	Precision	Recall
BBA	99.84	99.85	99.84	99.85
BOLA	99.86	98.84	99.98	99.97
CONV	96.83	96.68	96.28	96.14
PANDA	98.18	98.17	97.72	97.71
FESTIVE	96.15	96.12	96.51	96.55
R-MPC	97.57	97.56	98.46	98.45

B. Realistic commercial-based datasets

1) *Feature importance*: The feature importance shows in Fig. 2 that these ABR algorithms are more sensitive to the previous bitrate in the first place, which reveals their smoothness preference. Also, apart from VoD scenario for *S1*, all the ABR algorithms rely on the throughput measurements for the current and the last downloaded segment. Interestingly,

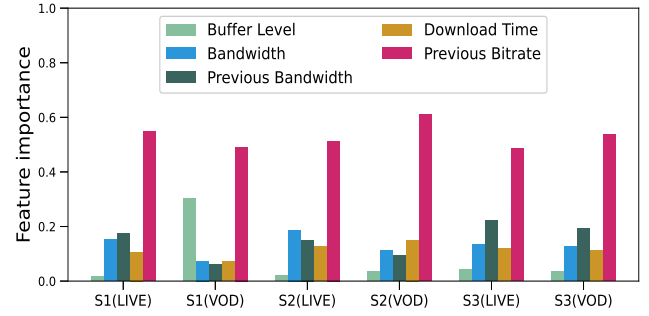


Fig. 2: Feature importance for the unknown ABR gained from three commercial services

the ABR of *S1* seems to give higher importance to the buffer level feature in VoD scenario.

2) *Metrics evaluation*: The classification accuracy of each ML algorithm for the different datasets is presented in TABLE III. This table shows that RF, DT, and GrdBst perform the best among the tested ML algorithms. For *s1* (both Live and VoD), these three algorithms achieve very high accuracy, slightly higher than 99%. For *S2*, RF and GrdBst perform even better than DT with an accuracy around 98% and 99% for Live and VoD respectively. Finally, *S3*, which has the highest number of quality levels (seven qualities), for Live we notice that RF and GrdBst achieve again high accuracy, with GrdBst achieving slightly higher accuracy. Nevertheless, with VoD, RF performs the best with an accuracy of 98.5%, which is nearly 1% better than DT and GrdBst.

TABLE III: Classification accuracy (%) of ML classifiers on LIVE and VoD commercial services datasets

	LGS	SVM	RF	DT	GrdBst	AdBst	NB	KNN
S1(LIVE)	83.37	93.57	99.30	99.06	99.24	61.69	97.86	98.06
S1(VoD)	84.64	96.43	99.29	98.99	99.26	98.31	98.83	98.67
S2(LIVE)	65.07	86.16	97.89	97.19	98.05	92.85	92.48	92.61
S2(VoD)	85.11	96.66	99.25	98.75	99.37	97.01	93.44	96.00
S3(LIVE)	88.90	88.93	98.26	97.75	98.84	87.17	90.84	95.42
S3(VoD)	84.27	84.28	98.57	97.86	97.04	87.79	95.12	96.35

Before moving to precision and recall metrics, we present the confusion matrix of the three best classifiers, in tables IV, V and VI for each commercial dataset. The first column of each confusion matrix shows the True labels of the classes, which is in our problem represents the actual ABR decisions for the quality levels (with class 0 representing the minimum bitrate). The first row presents the predicted ABR qualities.

Looking at these tables, we see that most of the bitrates are correctly predicted when enough segments are downloaded on these bitrates. However, the bitrates, that are rarely selected by the ABR, are predicted with lower accuracy. This behavior is clearly seen with the third dataset *S3*, looking at VI, which has seven qualities. This table shows that the highest bitrate is almost always predicted correctly, but the classifiers suffer from falsely classifying the other bitrates. Even though, it is interesting to see that RF does not fall below 90% for per-class correct prediction, compared to 83% and 80% for DT and

TABLE IV: Confusion matrixes of RF, DT and GrdBst for the first commercial dataset

(a) RF for S1(LIVE)					
True	Predicted Class				
%	0	1	2	3	4
0	96.9	2.2	0.3	0.0	0.6
1	0.3	99.4	0.0	0.1	0.2
2	0.2	0.8	96.8	2.0	0.2
3	0.3	0.2	2.3	94.8	2.3
4	0.0	0.0	0.0	0.1	99.9

(b) RF for S1(VoD)					
True	Predicted Class				
%	0	1	2	3	
0	99.1	0.6	0.0	0.3	
1	1.6	94.6	3.1	0.7	
2	0.0	1.2	95.9	2.9	
3	0.0	0.0	0.2	99.8	

(c) DT for S1(LIVE)					
True	Predicted Class				
%	0	1	2	3	4
0	96.3	2.5	0.3	0.3	0.6
1	0.3	98.3	0.7	0.5	0.2
2	0.0	1.0	94.5	4.5	0.0
3	0.1	0.3	3.1	94.0	2.4
4	0.0	0.0	0.0	0.2	99.8

(d) DT for S1(VoD)					
True	Predicted Class				
%	0	1	2	3	
0	98.7	1.0	0.2	0.1	
1	2.7	92.9	3.4	0.9	
2	0.5	4.5	90.2	4.8	
3	0.0	0.1	0.3	99.7	

(e) GrdBst for S1(LIVE)					
True	Predicted Class				
%	0	1	2	3	4
0	96.3	3.1	0.0	0.3	0.3
1	0.3	98.6	0.8	0.1	0.2
2	0.0	1.2	96.0	2.7	0.2
3	0.2	0.6	2.8	94.4	2.9
4	0.0	0.0	0.0	0.1	99.9

(f) GrdBst for S1(VoD)					
True	Predicted Class				
%	0	1	2	3	
0	99.2	0.6	0.1	0.2	
1	2.6	93.4	3.3	0.7	
2	0.0	3.9	93.2	2.9	
3	0.0	0.0	0.2	99.8	

TABLE V: Confusion matrixes of RF, DT and GrdBst for the second commercial dataset

(a) RF for S2(LIVE)				
True	Predicted Class			
%	0	1	2	3
0	99.7	0.2	0.0	0.1
1	1.3	97.9	0.1	0.6
2	0.5	2.3	93.6	3.5
3	0.2	0.3	0.5	99.0

(b) RF for S2(VoD)				
True	Predicted Class			
%	0	1	2	
0	99.4	0.5	0.1	
1	0.7	97.0	2.2	
2	0.0	0.4	99.6	

(c) DT for S2(LIVE)				
True	Predicted Class			
%	0	1	2	3
0	99.0	0.7	0.1	0.2
1	1.2	97.2	0.9	0.6
2	0.7	0.6	94.3	4.4
3	0.2	0.4	1.0	98.4

(d) DT for S2(VoD)				
True	Predicted Class			
%	0	1	2	
0	98.8	1.1	0.1	
1	1.0	96.4	2.5	
2	0.0	0.6	99.4	

(e) GrdBst for S2(LIVE)				
True	Predicted Class			
%	0	1	2	3
0	99.5	0.4	0.1	0.1
1	1.3	97.5	0.6	0.7
2	0.6	4.3	89.5	5.6
3	0.2	0.4	0.7	98.7

(f) GrdBst for S2(VoD)				
True	Predicted Class			
%	0	1	2	
0	99.5	0.4	0.1	
1	0.6	97.9	1.5	
2	0.0	0.4	99.6	

GrdBst respectively. This performance can be further improved by collecting more data points for these bitrates when we have more qualities involved.

Finally, we present precision and recall metrics in TABLE VII, as many works recommend that using accuracy only is not enough. Depending on the application, one metric may be favored over the other. Taking our bitrate selection problem, if the purpose is to check, for any given bitrate, how often it was truly predicted out of all the total predictions of this class, then precision is the metric that should be taken into account. On the other hand, if the goal is to assure that the actual

TABLE VI: Confusion matrixes of RF, DT and GrdBst for the third commercial dataset

(a) RF for S3(LIVE)									
True	Predicted Class								
%	0	1	2	3	4	5	6		
0	90.2	4.3	0.5	1.0	0.8	0.8	2.5		
1	2.5	93.5	1.2	1.1	0.7	0.3	0.7		
2	0.7	3.6	90.7	2.1	0.9	0.9	1.1		
3	0.1	0.5	2.5	93.4	2.8	0.3	0.5		
4	0.1	0.2	0.1	3.3	93.3	2.2	0.7		
5	0.1	0.0	0.2	0.5	5.6	90.3	3.3		
6	0.0	0.0	0.0	0.0	0.0	0.3	99.6		

(b) RF for S3(VoD)									
True	Predicted Class								
%	0	1	2	3	4	5	6		
0	90.2	4.3	0.5	1.0	0.8	0.8	2.5		
1	2.5	91.5	1.2	1.1	0.7	0.3	0.7		
2	0.7	1.6	90.7	4.1	0.9	0.9	1.1		
3	0.1	0.5	2.5	93.4	2.8	0.3	0.5		
4	0.1	0.2	0.1	3.3	93.3	2.2	0.7		
5	0.1	0.0	0.2	0.5	5.6	90.3	3.3		
6	0.0	0.0	0.0	0.0	0.0	0.3	99.6		

(c) DT for S3(LIVE)									
True	Predicted Class								
%	0	1	2	3	4	5	6		
0	90.0	4.3	1.5	2.3	0.5	0.5	1.0		
1	4.7	87.9	3.4	2.8	0.7	0.4	0.1		
2	2.0	6.1	83.3	5.2	2.2	0.7	0.4		
3	0.6	1.0	3.6	89.6	4.3	0.6	0.4		
4	0.4	0.4	0.6	4.7	89.5	3.8	0.5		
5	0.1	0.0	0.4	0.8	7.7	87.6	3.3		
6	0.1	0.0	0.1	0.0	0.2	0.6	98.9		

(d) DT for S3(VoD)									
True	Predicted Class								
%	0	1	2	3	4	5	6		
0	88.0	6.8	0.8	1.3	0.5	0.5	2.3		
1	3.9	88.2	2.3	3.5	1.0	0.4	0.6		
2	0.7	5.6	87.0	3.3	1.6	0.9	0.9		
3	0.1	0.6	2.5	92.8	3.5	0.1	0.5		
4	0.0	0.2	0.9	5.8	89.5	2.9	0.6		
5	0.2	0.0	0.0	0.4	7.2	88.5	3.7		
6	0.1	0.1	0.1	0.0	0.1	0.4	99.3		

(e) GrdBst for S3(LIVE)									
True	Predicted Class								
%	0	1	2	3	4	5	6		
0	90.4	3.2	0.0	3.8	0.0	2.6	0.0		
1	2.7	90.1	2.3	4.1	0.4	0.4	0.0		
2	0.0	2.3	87.3	5.9	2.7	0.9	0.9		
3	0.0	0.7	2.6	93.5	2.9	0.2	0.2		
4	0.1	0.1	1.0	3.5	91.1	3.8	0.3		
5	0.0	0.1	0.0	0.5	4.3	92.7	2.4		
6	0.1	0.0	0.0	0.0	0.0	0.2	99.8		

(f) GrdBst for S3(VoD)									
True	Predicted Class								
%	0	1	2	3	4	5	6		
0	86.3	2.1	2.9	0.2	2.6	2.6	3.3		
1	3.3	88.8	3.6	4.3	0.0	0.0	0.0		
2	0.5	2.9	80.6	14.9	0.6	0.3	0.3		
3	0.2	0.2	4.4	92.5	2.5	0.1	0.3		
4	0.0	0.0	0.2	3.7	92.5	3.4	0.1		
5	0.0	0.0	0.1	0.3	4.2	92.4	3.0		
6	0.0	0.0	0.0	0.0	0.0	0.3	99.6		

classes are truly predicted then recall is the preferred metric. TABLE VII shows that each RF, DT, and GrdBst achieves high precision and recall, with RF and GrdBst slightly better than DT in some scenarios.

TABLE VII: Precision and Recall(%) using RF, DT and AdBst classifiers on LIVE and VoD commercial services datasets

	RF		DT		GrdBst	
	Precision	Recall	Precision	Recall	Precision	Recall
S1(LIVE)	99.16	99.17	98.87	98.86	99.16	99.17
S1(VoD)	99.21	99.19	99.12	99.11	99.20	99.16
S2(LIVE)	98.98	98.99	98.44	98.43	98.71	98.71
S2(VoD)	99.32	99.33	98.96	98.95	99.33	99.33
S3(LIVE)	97.65	97.83	96.87	96.83	97.89	97.18
S3(VoD)	97.99	97.99	96.08	96.06	96.95	96.94

In addition, processing latency is another important metric, especially for online learning. It is even more important for video streaming when the download decisions rely on bitrate prediction. To guarantee the latency requirements, we present in TABLE VIII the training and prediction times for the different ML algorithms. The training time is computed over the whole dataset, and the prediction time is averaged over 20% of the datasets. This table shows that DT is faster than both RF and GrdBst, for training and prediction phases. For the training phase, GrdBst is computationally the most demanding algorithm, though it performs well, and slightly better than RF, for the prediction phase.

VI. APPLICABILITY TO PRE-FETCHING BASED ABR DEPLOYMENTS

Predicting the ABR decisions serves the pre-fetching based systems in the first place. As mentioned in Section I, both CDN and P2P streaming may need to deploy some pre-fetching techniques to improve the QoE. Hence, it becomes challenging when ABR decisions are needed to avoid any source-

TABLE VIII: Processing time of ML classifiers on Live and VoD commercial services datasets

ML algo	Training		Prediction	
	mean (s)	std(s)	mean (ms)	std (ms)
LGS	1.69	0.99	0.19	0.05
SVM	0.99	0.46	56.69	38.93
RF	1.55	0.38	10.74	1.77
DT	0.04	0.016	0.09	0.02
GrdBst	14.64	6.71	7.51	3.47
AdBst	1.61	0.48	33.80	27.45
NB	0.02	0.003	0.21	0.03
KNN	0.41	0.015	41.68	4.82

wasting downloads. To our knowledge, most of the existing works keep prefetching the last requested quality, ignoring the possible track switching which becomes problematic in the high dynamic adaptive scenarios, especially when the highest bitrates are requested over low bandwidth. Therefore, our goal is rather to predict the switches with the most important impact on overhead and QoE, which is possible according to our results. Moreover, This approach is shown to be representative of the real-world performances, where the highest bitrate is almost predicted correctly which will be translated into less overhead and better QoE. Here, this approach is reasonable for online prediction, for more reasons than just prediction accuracy. On one hand, and for the training phase, many users can participate by sending their measurements of the buffer and the measured bandwidth to a central server, which reduces the time needed for data collection effectively. On the other hand, for the prediction phase, the input features for the upcoming segments are either pre-known (e.g. the segment size, the download time, and the measured bandwidth) or easy to predict (e.g the buffer level giving the download time and the previous buffer level). On another note, the prediction latency, as shown in TABLE VIII, is negligible compared to the segment duration. These reasons make our approach quite efficient to move forward a smarter pre-fetching using ML-algorithms.

VII. CONCLUSION

In this paper, we presented how machine learning-based techniques can be used to predict the behavior of ABR algorithms. We performed our experiments over different ABR algorithms from simulation and real-world closed-source deployments. Our results show that Random Forest and Gradient boost algorithms are able to achieve a very high prediction accuracy, using only the basic information provided as input to the application layer. This work serves, in particular, any pre-fetching based streaming deployment, where the ABR decisions are needed beforehand. We plan to expand this work in this direction and investigate the effects of such prediction on pre-fetching based systems such as P2P networks.

ACKNOWLEDGMENT

We thank our data scientist colleague Igor Mukam for his great work on collecting and processing the realistic datasets.

REFERENCES

[1] Cisco, "Cisco visual networking index: Forecast and methodology, 2017-2022," *White Paper*, 2019.

[2] A. Bentalab, B. Taani, A. C. Begen, C. Timmerer, and R. Zimmermann, "A survey on bitrate adaptation schemes for streaming media over http," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, 2019.

[3] D. S. Berger, "Towards lightweight and robust machine learning for cdn caching," in *Proc of the 17th ACM Workshop on Hot Topics in Networks*. Association for Computing Machinery, 2018.

[4] M. Claeys, S. Latré, J. Famaey, and F. De Turck, "Design and evaluation of a self-learning http adaptive video streaming client," *IEEE Communications Letters*, vol. 18, no. 4, 2014.

[5] T. 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, 2008.

[6] D. Liu, Y. Li, J. Lin, H. Li, and F. Wu, "Deep learning-based video coding: A review and a case study," *ACM Comput. Surv.*, vol. 53, no. 1, 2020.

[7] Y. Chien, K. C. Lin, and M. Chen, "Machine learning based rate adaptation with elastic feature selection for http-based streaming," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2015.

[8] A. Lekharu, S. Kumar, A. Sur, and A. Sarkar, "A qoe aware lstm based bit-rate prediction model for dash video," in *2018 10th International Conference on Communication Systems Networks (COMSNETS)*, 2018.

[9] J. Kua, G. Armitage, and P. Branch, "A survey of rate adaptation techniques for dynamic adaptive streaming over http," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, 2017.

[10] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," in *Proceedings of the 2014 ACM Conference on SIGCOMM*. Association for Computing Machinery, 2014.

[11] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "Bola: Near-optimal bitrate adaptation for online videos," in *INFOCOM*. IEEE, 2016.

[12] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Begen, and D. Oran, "Probe and adapt: Rate adaptation for http video streaming at scale," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, 2014.

[13] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive," in *Proc of Conference on Emerging Networking Experiments and Technologies*. Association for Computing Machinery, 2012.

[14] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over http," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, 2015.

[15] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques." *Informatica*, 2007.

[16] S. K. Murthy, "Automatic construction of decision trees from data: A multi-disciplinary survey," *Data Mining and Knowledge Discovery*, vol. 2, 1997.

[17] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, 2001.

[18] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, 1997.

[19] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, 2001.

[20] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, 1967.

[21] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, 1995.

[22] H. Yousef, J. Le Feuvre, P.-L. Agneau, and A. Storelli, "Enabling adaptive bitrate algorithms in hybrid cdn/p2p networks," in *Proceedings of the 11th ACM Multimedia Systems Conference*, ser. MMSys '20. Association for Computing Machinery, 2020.

[23] H. Riiser, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Commute path bandwidth traces from 3g networks: analysis and applications," in *ACM MMSys*, 2013.

[24] J. van der Hoof, S. Petrangeli, T. Wauters, R. Huysegems, P. R. Alfacc, T. Bostoen, and F. De Turck, "Http/2-based adaptive streaming of hevc video over 4g/lte networks," *IEEE Communications Letters*, vol. 20, no. 11, 2016.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, 2011.

[26] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.