Streaming Pattern Based Feature Extraction for Training Neural Network Classifier to Predict Quality of VOD services

Suman Pandey
POSTECH,
Pohang,South Korea
suman17july@gmail.com

Mi Jung Choi
Kangwon National Univeristy
Chuncheon, South Korea
mjchoi@kangwon.ac.kr

Jae-Hyung Yoo
POSTECH,
Pohang,South *Korea*
jhyoo78@postech.ac.kr

James Won-Ki Hong
POSTECH,
Pohang,South Korea
jwkhong@postech.ac.kr

***Abstract*— Netflix, Amazon Prime, and YouTube are the most popular and fastest-growing streaming services globally. We first studied the characteristics of the streaming patterns of these three services and then utilized these characteristics to extract several features for a quality assessment of the stream. Any streaming traffic has three main characteristics including 1) Adaptive Bit Rate (ABR) streaming 2) On-Off Cycle, 3) Buffering, and Steady-state phases. We observed that streaming providers vary in their ABR strategies and rate-controlling mechanisms. The amount of data they download in the buffering state and steady-state varies. Their On-Off cycle length and data block size vary as well. The quality of their services will depend on their streaming characteristics. Therefore we extracted 12 unique features from the streams based on these streaming patterns. We then trained a perceptron based neural network model for video quality assessment. The model was tested with 50 streams of each service, captured at varying access network bandwidth ranging from 75kbps to 30Mbps. The model could successfully identify a good and a bad stream with an accuracy of 0.929 for YouTube, 0.857 for Amazon Prime, and 0.933 for Netflix. At last, we analyzed the importance of each feature for these three services. Our approach can be used to compare and contrast the streaming services strategies and fine-tune their ABR and flow control mechanisms.**

Keywords— QoS, QoE, streaming, Netflix, Amazon, YouTube

I. Introduction

Streaming media is flocking the IP network with video, audio, game, live TV, and educational streaming services. With the present pandemic brought about by Covid-19, social distancing could be a norm, and streaming media consumption could further increase. We often need to perform a video quality test for these streaming services. Performance testing of these applications has become a necessity considering the market competition. Users expect high quality, breakage free, flawless streams even in challenging network conditions. Despite a legacy of research done in streaming technology, this is still an active research topic.

There are two different approaches to quality assessments. First is a subjective approach also called Quality of Experience (QoE). This is a user-perceived quality of the video and, mostly rated using Mean Opinion Score (MOS) [1] in the range of 1 to 5. The second approach is the Objective pproach also called Quality of Service (QoS) [2, 3], measured using the application, network, and transport parametric. Several standards have been developed to translate a QoS model to MOS based QoE models using mathematical equations [4, 5].

Even though several applications, network, and transport parametric models have been developed to calculate the QoS, gathering QoS metrics is always a challenging task. Measuring QoS depending totally on application layer metrics related to video player is impractical, as it will require different techniques for different service providers. Application layer metrics could be startup delay, rebuffering events, playback buffer, etc. Media Presentation and Description files (MDP) [6] or player API such as YouTube iframe API [7] can be used to collect these data, however, MDP files are mostly encrypted and most of the service providers don't provide player APIs. Hence designing a generic QoS model based on application layer metrics becomes a challenging task.

TCP throughput, video packet delay, and jitter, etc. can be used for network parametric models, however, in ABR technology these parameters are ineffective as the video encoded at different resolutions and bit rates are stored in server, and the bitrate changes with the network condition resulting in almost no jitter and packet loss. Also, the playout buffer at the application layer manages the delay to a significant level.

As the streaming technology changes and adopts the ABR and rate controlled streaming, our parametric models need to evolve. Also, the measurement technique to collect QoS metrics should be generic to all the applications. Hence in this paper, we depend totally on streaming patterns to extract QoS metrics. The goal of this paper is to provide feature extraction techniques based on patterns and characteristics of streaming solely.

To achieve this goal we first surveyed three major streaming service providers YouTube, Amazon Prime, and Netflix, and their streaming patterns. For our purpose, we identified three characteristics of streaming. Firstly all the streaming follows ABR technology, which means the bitrates of the traffic depend highly on network conditions, device size, and capacity. Secondly, the streaming traffic is divided into buffering and steady-state phases, and finally, they have an On-Off cycle pattern in steady-state as shown in Figure 1. This way the traffic is rate controlled by the service provider and doesn't totally depend on the network conditions. Every

978-3-903176-32-4 © 2021 IFIP

service providers have their own rate controlling mechanism and ABR algorithm. We surveyed how Netflix, Amazon, and YouTube are different in their rate controlling mechanism and ABR technologies.

Based on these surveys and characteristics we extracted 12 different features from the streams to train a neural network-based classifier. These classifiers can give us the quality of the stream based on their streaming pattern. For labeling the data we relied on the three "ground truth" features including resolution, bitrates, and rebuffering duration. Labeling based on subjective, objective, and estimated MOS score as explained in ITU-T P.1203.1 [1] and ITU-T P.800.1 [4] is another interesting and well-studied area. For keeping our ground truth model simple we considered only bitrate, rebuffering, and resolution-related parameters for labeling purposes. Apart from these network parameters, we gathered 9 other features directly from the flow shown in Figure 1. These features include block size, #blocks, standard deviations of block size, Off duration, standard deviation of Off duration, buffering phase bit rate, steady-state phase bit rate, buffering phase durations, and progressive download ratio. These features are based on the characteristics specific to the rate controlled and ABR based VOD streaming traffic. We then used these feature sets to train a perceptron based neural network classifiers.

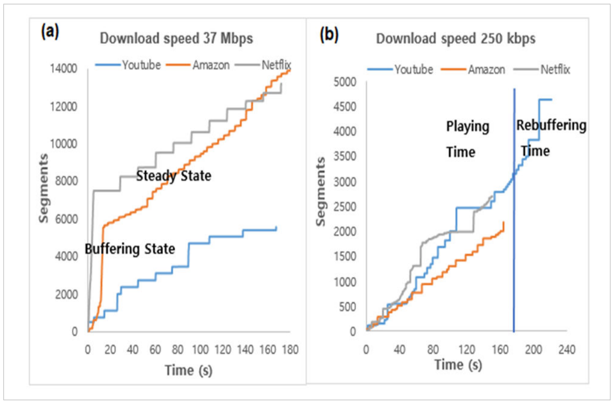

Fig 1. The Number of packets downloaded per time stamp for Peppa Pig, Episode 1: (a) download speed of 37 Mbps, and (b) download speed of 250 kbps.

This work is unique in a way that it provides a detailed survey of streaming characteristics of YouTube, Amazon, and Netflix. Then deduces features that depend totally on the flow data of the first 2000 packets of the stream. Our neural network model could classify the traffic as good and bad with an accuracy of 0.929 for YouTube, 1 for Amazon, and 0.867 for Netflix. And our feature comparison approach will help service providers to fine-tune their ABR technologies and rate-controlling mechanisms.

The rest of the paper is organized in the following way. Section II describes related work. Section III elaborates on our data collection mechanism. Section IV provides a survey of streaming pattern characteristics. Section V explains the feature sets. Section VI evaluates a neural network-based classifier on these feature sets. Section VII concludes this research with future work.

II. Related Work

A great deal of QoE models has been already surveyed in past. Several studies have proposed parametric studies and are standardized by ITUT [1~4]. Several researchers have surveyed these QoE modeling approaches [5, 8]. P. Juluri [9] Presents a survey of parametric features and compared previous research based on the parameters used. Parametric models they compared include startup delay, bitrate, resolution, rebuffering events, bitrate switching events, failure rates, jitter, CPU utilization, packet loss, etc. However, these surveys did not mention any streaming pattern-related parameters and did not study the effect of the streaming parameters over QoE. They also failed in realizing the potential of machine learning in video quality assessment. Another set of studies in this field are QoE measurement surveys of different service providers. YouTube traffic has been studied elaborately in the past [15, 16, 17, and 18] as they provide open APIs to collect the QoE metrics. A detailed measurement survey about Netflix has also been published [19]. This survey focuses mainly on their CDN infrastructure improvement strategies to improve the quality. Another measurement study about QoE of Azure, Amazon, and YouTube was conducted [20] which also focused on improving location strategies.

Another line of studies in this direction is related to ABR technology. ABR based HTTP-DASH is a well-accepted protocol for streaming media delivery. A detailed survey on different ABR technology is presented well by A. Bentaleb et al.[11]. Other researchers have also exploited ABR characteristics but have focused on studying the effect of aggregated streaming traffic [12, 13, and 14]. Besides these studies lack in presenting a detailed feature extraction and modeling a machine learning algorithm for quality assessment. Machine Learning has potential in pattern recognition and classification, however, most of the studies are limited to traffic classification [21,22]. Some recent studies focus on continuous QoE prediction of a single running stream based on this continuous pattern [23]. However, training a model based on the features extracted from the flow based on the streaming characteristic have not been studied yet.

In this research, we proposed unique feature sets, extracted directly from the flow data. The feature is chosen carefully after analyzing the streaming characteristics related to ABR and flow control mechanisms. A machine learning approach is used to train the model for the classification task. And feature importance is analyzed for each service provider. The use cases presented in this paper are based on real traffic traces from YouTube, Netflix, and Amazon Prime.

III. Data Collection

In this research, we solely depend on the flow information to collect the major QoS parameters. The network flow is gathered based on five-tuple information (Protocol, IP/port) from the campus network. We captured 50 different types of videos for each service including action movies, cartoons, and documentaries. Each video is of 180 sec duration excluding ads and was captured with bandwidth range of 75Kbps ~ 30 Mbps, except Netflix that doesn’t provide services below 250Kbps. Chrome browser DevOpp tools have bandwidth configuration options, which were used to configure client bandwidth before capturing a video stream. Wireshark is used for stream capturing and storage. The videos were captured between 4 PM~6 PM during autumn 2019.

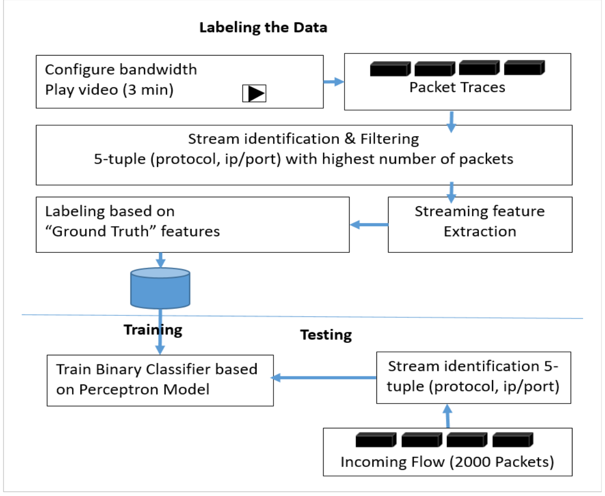

Figure 2. Feature Extraction and Model Traing

Figure 2. provides an overview of the methodology used in this paper. First, the successive IP packets having 5-tuple flow information, including protocol (TCP/QUIC), source address (streaming server IP address), source port (433), destination address (host address), and destination port (port on which the browser is running) is filtered. To classify the right stream, the IP address was matched with the list of domains of the service provider. However, there could be multiple types of flows from the same service providers containing other information such as page structure, logic, or ads. We identified the video flows by filtering flows with the highest number of packets. After the stream is identified, only three features including time (ti), packet (si), and the packet length (sl) of each packet are extracted from the packet trace. These three features are used for extracting all the streaming pattern related features later. We first calculate the rebuffering time and bitrate as a “ground truth” feature for labeling the stream. Then we extract streaming specific features to train our neural network model. Part of the sample data is available on Github [24] for anyone to use.

IV. Streaming Pattern Characteristcs

VOD Streaming traffic is different from other traffic in many ways. VOD streams use ABR technology and are rate controlled.

A) Adaptive bit rate streaming (ABR) - HTTP DASH

Most of the video streaming services including YouTube, Amazon Prime and Netflix, etc, uses HTTP-DASH protocol for streaming. HTTP-DASH is based on ABR technology. Netflix and Amazon Prime use TCP as the transport layer and Youtube can use TCP or UDP-based QUIC protocol at the transport layer depending on the browsers. Whether Youtube uses TCP or QUIC, their underlying traffic is rate controlled and follows ABR technology. In ABR, the internet speed and CPU availability of each viewer are measured to dynamically adjusts the video quality they are being served. We can see this behavior in Figure 2. As the access network bandwidth increases the bitrate increases. This behavior demonstrates ABR technology. In HTTP DASH, the adaptive bit rates will be selected based on the network capacity and end-user devices such as mobile, computer, or wide-screen TV sets. If the network capacity is higher, or the screen size and resolution is higher the bitrate will be higher. We measured the bit rates of two different types of videos available on all these three services by throttling the network bandwidth capacity (Figure 3). We see with increasing bandwidth capacity at end-user the bit rate increases. We see a varied range of bit rates for the same video as the same video is encoded at different resolutions and stored at the cache. Depending on the available bandwidth a video encoded at an appropriate bitrate is chosen. A higher bitrate will accommodate higher image quality in the video output. Even though the ABR technology will adapt the bitrates dynamically there is a maximum bit rate that can be achieved for each video. After a point, the bit rate doesn’t change as shown in Figure 3. Bitrate is almost constant after the 5Mbps capacity for both of these videos, as that is the best quality video available on the server.

The bitrate depends on the codec used and the resolution settings in the codec. For ex, for 352 X 288/240p resolution an average of 1500kb/s is excellent for 352 X 576/480p an average of 3000kb/s is needed, 704 X 576p you start with an average of 4000kb/s etc. Bitrates should be expected to go up whenever the resolution goes up, as more data is being processed. The most popular codec are MPEG4-p10 (AVC/H.264), VP9 or AV1 (royalty-free codec), and HEVC (royalty-bearing codec). The highest bitrate in Figure 3 represents the best available quality of the video in server. We also see that different service providers use different bit rates for the same video. For example, the maximum packets are downloaded by Amazon (16000), followed by Netflix (14000) and YouTube (6000), for the same content (Figure 3). The best bit rate result is shown by Amazon and then Netflix followed by Youtube (Figure 3 [a]). For lower bandwidth, the bitrates were adjusted well for Netflix and Amazon based on adaptive bitrate protocol HTTP-DASH (Figure. 3[b]). For this particular video, Youtube does not have the video encoded at lower bit rates, hence one can see that bitrate is still high for YouTube (Figure 3 [b]). However, in most cases bitrate is directly proportional to the available bandwidth. Even if the available bandwidth is high, some videos are not encoded with high resolutions and hence they still acquire a less amount of bandwidth and eventually a lower bitrate. Although the available bandwidth is high (37 Mbps), some videos show lower bitrate. This is because those videos are encoded at low bit rates and are not available in high resolutions. The bit rate is also decided based on the video Spatio-temporal complexity. In Figure 2, a popular cartoon Pepe Pig is encoded at a much lower level compared to Spider-Man as the Spatio-temporal complexity of spider man is much higher than Pepa Pig.

As we see the bitrate need for each video is different and depends on various factors such as Spatio-temporal complexity of video, codes, network capacity, and end-user devices, hence we can't determine the quality of the video merely based on the bitrates and resolutions.

B) Buffering and Steady-State

Streaming traffic is not only adaptive but also rate controlled. The rate of this traffic is much lower than the end-to-end available bandwidth. The videos are not transferred as a chunk, rather they are transferred in small data blocks. This is done to avoid downloading the entire video at once as the viewer might not watch the entire video. The rate is controlled by downloading the video with a higher bit rate initially in the buffering phase followed by the lower bit rates with the On-Off cycle in the steady-state phase as shown in Figure 3.

In the buffering phase, the download bit rate is higher and the player tries to accumulate enough data before starting the playback. The data accumulated in the buffering phase ensures a smooth video watching experience in the fluctuating network conditions. Starting of the first Off period is the end of the buffering phase. We observed the decrease in the buffering phase as the network bandwidth decrease.

C) On and Off cycle

The other important characteristic of Streaming media is On-Off cycle for streaming packets. This approach reduces the network traffic as well. During each On period, a block of data is transferred. During the Off period, no data is transferred. The size of the block downloaded in the On period could vary from stream to stream. The duration of Off size can also vary from stream to stream.

We observed the average block size of each streaming service is different. A bigger block size means there is enough data accumulated for playout. A bigger Off cycle means the network is free for other traffics. Bigger On-Off cycles ensure a lesser number of blocks for the same amount of data and reduced processing time at the client and servers. A smaller On-Off cycle ensures that the client is never overwhelmed with the transferred data.

Amazon has the minimum block size. They also have a minimum Off cycle in general for all kinds of streams. This short On-Off cycle ensures that the client is not overwhelmed with the transferred data, and also makes sure that there is enough data accumulated for playout. However, this approach causes higher processing at the server as well as the client. Netflix and YouTube have a higher block size, ensuring lower processing at the client and server on the cost of extra accumulated data at the client buffer.

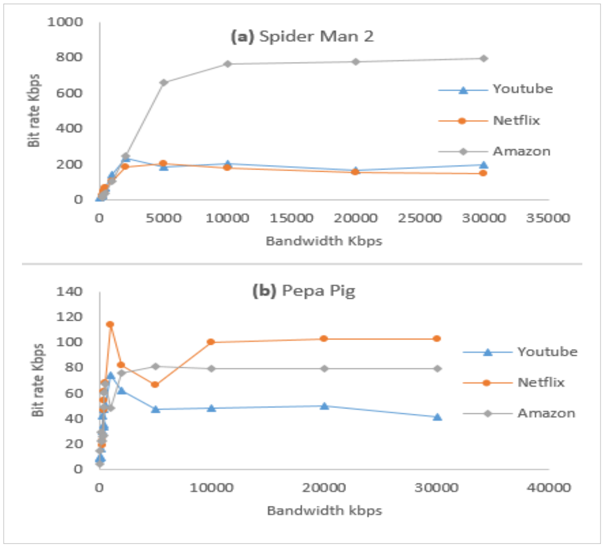

Figure 3. Varying bit rates for Access Network Bandwidth.

Hence we conclude that Over The Top (OTT) service providers typically use different On-Off cycles, buffering, and steady-state phase and bit rates. Also, the bitrates of buffering and steady-state phase is different for different service providers. Thus, even if the same basic ABR technology is used, the performance of ABR streaming can vary depending on these characteristics. Hence in this paper, we extracted several features from these streams based on these characteristics to train our machine learning algorithm for quality assessment.

V. Feature extraction

We process the raw data stream and calculate the statistical features of video flows. After extensive analysis of video flow data, we select a number of QoS related features as explained in Table I. All these features are based on the downstream data only. Later we also analyzed the importance of each feature for each service provider.

A) Ground Truth and Labeling

We need to label each stream as good and bad before passing it to the supervised learning module. Labeling data by calculating the exact MOS in ITU-T P.800.1 [4] score is very challenging. MOS scores are subjective measures and require a lot of user input. It is also subjected to change from user to user, their mood and perception. Calculating a MOS will also require meticulously setting up the environment, using a wide variety of parameters, and making multiple observations. The other approach to gathering a quality matrix is using application-level data. This would involve the MPD files or the application or player specific APIs to collect data from the codec, however, there are no such APIs available for Netflix and Amazon.

Hence we considered bitrate and rebuffering duration as our main criteria to label the stream as good and bad. Bitrate is compared with the screen resolution, and if it is lower than the expected bitrate for this resolution the stream is labeled as a bad stream. Any kind of rebuffering event results in a bad quality stream, so if rebuffering is greater than 0 we consider it a bad stream.

Bitrate: is calculated as the number of bits received and decoded during a play, divided by the total playing time (s / t_n). The actual video length (t) could be lesser than playtime (t_n). All the training dataset are of 180 sec long. However, in reality, the video can be played for longer depending on the interruptions, jitter and rebuffing. The total playtime *(t_n)* is the duration of a video that is being played with jitter, and re-buffering. For a 180-second video, an application can take more than 180 seconds to play completely, especially if network conditions are not good. ITU-T P.1203.1[4] standards define the minimum bitrate requirements for a particular resolution. For video resolution of 240p, 360p, 480p, 720p and 1080p, corresponding recommended bit rate is 75-150 kbit/s, 220-450 kbit/s, 375-750 kbit/s, 1050-2100 kbit/s and 1875-12500 kbit/s. We followed these standards to define a good bitrate for the captured video.

Rebuffering time: This is the time in which a viewer experiences re-buffering issues (i.e., when a video stops playing because of player's buffer underflow and not due to user interventions such as scrubbing or pausing). Rebuffering is the main cause of dissatisfaction among users [15, 23]. Rebuffering is calculated by subtracting actual playtime with a playtime of the video ($t_n - t$). For higher bandwidth, re-buffering time for all these services is null, showcasing a good QoS; however, for lower bandwidth, the re-buffering time for YouTube video is higher compared to Netflix and Amazon. Any kind of rebuffering represents the lower quality of the stream. We labeled all the streams with any rebuffering as bad quality streams. Using bitrate and rebuffering duration we labeled each stream as good and bad. In future, we can improve our ground truth features by incorporating human-rated MOS scores.

Table I. FEATURE SETS BASED ON STREAMING CHARACTERISTICS

Categories	**Feature description**
Ground Truth	Bandwidth ,Resolution, Bitrate , Rebuffering time
On-Off Cycle	Average data block size (ADBS), Number of blocks (NB), Standard Deviation block size (SDBS), Average Off duration (AOD), Standard Deviation Off duration (SDOD)
Buffering and Steady-state	Buffering phase duration (BPD), Buffering phase bit rate (BPBR), Steady-state bit rate (SSBR), Progressive download ratio (PDR)

B) Streaming Feature Extraction

As we have seen, a typical VOD streaming application has several unique characteristics. We extracted several streaming features from the first 2000 packets of the stream. All the stream data is first stored in the CSV file and then a python script is used to extract these features from the raw data. We only require time (ti), packet (si), and the packet length (sl) of each packet to extract these features. The features extracted from the streams include the average size of the data block, no of data blocks, standard deviation of block size, average Off cycle duration, the standard deviation of Off cycle, bit rate at buffering state, bit rate at steady-state, buffering state duration and progressive download ratio. Table I lists all the extracted features with their abbreviations. We combined these features with network parametric features such as bandwidth and resolution to train a perceptron based neural network classifiers.

On-Off cycle features: One of the important characters of streaming traffic is their On-Off Cycle. There are several features we extracted based on this property.

1. ADBS: is a measure of the average height of the On cycle. In one stream, we will have several On cycle. The height of each On cycle represents the number of packets downloaded in the On cycle. We can also call it a data block. The block size of Netflix, Amazon, and Youtube are different for different bandwidth and resolutions.
2. NB: a measure of the number of cycles. This measure is lower if the block sizes are bigger, as the required number of packets are downloaded in less number of cycles. NB is higher for Netflix and YouTube as compare to Amazon Prime.
3. SDBS: each block size in one stream could be different, hence we extract the standard deviation of block size as another feature.
4. AOD: average Off cycle duration is the period of the time when there is no data transferred.
5. SDOD: Off cycle in one stream might be of different durations, so we calculated the standard deviation of Off cycle also as a feature.

Buffering and Steady-state features: The other important characteristics of the video stream are that they are downloaded in two phases, bit rates of buffering phase and steady-state phase differ. We extracted another four features based on this characteristic.

6. BPD: Buffering phase bit rate is much higher than the steady-state bit rate. We divide the steady-state phase with the buffering phase as soon as we encounter an Off cycle in the stream.
7. BPBR: Buffering phase bit rate is usually higher to accumulate enough data blocks for playback.
8. SSBR: The steady-state phase starts when the buffering phase ends. The download bit rate is much lower in the steady-state phase with the On-Off cycle.
9. PDR: Progressive download ratio is the ratio at steady-state bitrate and average bitrate ($(s_s * sl / t_s) / (s * sl / t_n)$). If enough packets have been buffered then, PDR will result in lower values. PDR of 1 indicates there is no buffering done at the buffering state at all.

The slope of the buffering phase as well as steady-state phase depends on the available bandwidth. One can observe that the Netflix buffering phase downloads a higher number of packets in less time compared to Amazon. Netflix has the lowest average PDR of 0.49, which means they buffer the highest amount of data. These features correctly define streaming traffic patterns. Rebuffering, delay, jitter, and distortion of the picture quality depend a lot on these streaming characteristics. Hence we trained a neural network model with these features and labeled the streams based on ground truth features. This model is then used to classify a good stream from a bad stream.

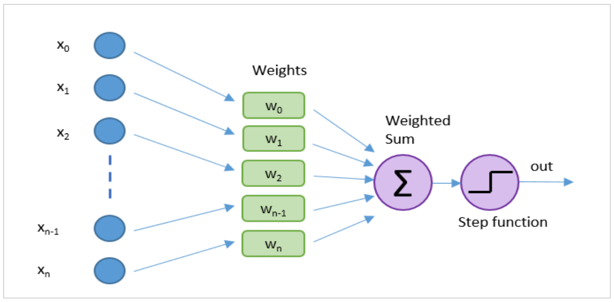

Figure 4. Perceptron model training

VI. Classifier based on Nural Network

The neural network is a proven methodology for pattern recognition and we can clearly see a pattern in the streaming data. This pattern includes the On-Off cycles, block size, number of blocks, Off cycle length, number of Off cycles, bit rates at buffering phase, buffering phase duration, bit rate of steady-state, etc. Machine learning and pattern recognition is an obvious start to solve this problem. The stream pattern recognition technique can distinguish a good stream and a bad stream. We extracted these features and trained a perceptron based neural network binary classifier.

Perceptron is a single-layer neural network. The classification using perceptron begins by taking all the input values and multiplying them by their weights. Then, all of these multiplied values are added together to create the weighted sum. The weighted sum is then applied to the activation function, producing the perceptron's output (Figure 4.). The activation function plays an integral role in ensuring the output is mapped between required values 0 and 1. It is important to note that the weight of input is indicative of the strength of a node. Similarly, an input's bias value gives the ability to shift the activation function curve up or down. Perceptron can only separate the dataset linearly. We used a perceptron based neural network binary classifier to decide whether an input stream of 2000 packets belongs to a specific class.

We used python 3.7 and scikit-learn python library, which provides a user-friendly APIs for machine learning. The training and testing dataset is standardized for each feature dimension before training the perceptron model. The learning rate (eta) is equal to 0.1. Perceptron learning process Our test results could predict the quality of stream of YouTube with accuracy: 0.929, Amazon with accuracy 0.857, and Netflix with accuracy 0.933.

Table II presents the weight assigned to each feature by the neural network model. This shows which features are important for each streaming service. As the patterns for these three services are different their feature importance is also different. SDBS is most important feature for all these three service providers. High standard deviation of the block size is indication of the bad quality of the video. NB is important feature for Youtube and Netflix and SDOD is important feature for Amazon Prime. Number of Blocks is not an indicative feature for Amazon, as their ABR strategies include higher number of blocks with smaller block size, however higher standard deviation of Off duration indicates a poor quality of video. These weights are subjected to change as we expand our training dataset.

Table II. Feature importance

Features	YouTube	Amazon	Netflix
ADBS	0.21107938	0.1042188	-0.00078638
NB	0.39187759	-0.02127561	0.08386438
SDBS	-0.4139249	-0.59889207	-0.10962606
AOD	0.07150587	-0.10519978	-0.08321903
SDOD	-0.01490527	-0.94945443	-0.08138447
BPD	-0.05107396	0.45066718	-0.00795923
BPBR	-0.16957214	-0.2332803	-0.00326056
SSBR	0.2297411	0.2813748	-0.03777679
PDR	-0.20793273	-0.00216228	0.01263221

This paper demonstrates an approach to do a quality analysis based on the flow information. The approach can also be used to compare and contrast the importance of different features of these services. Our analysis can be used by the service providers to analyze their competitors and fine-tune their streaming strategies.

VII. Conclusion and Future Work

This paper can be considered as benchmarking paper. In this article, we compared the most popular VOD service providers, Youtube, Amazon Prime, and Netflix based on their ABR approach and flow control mechanisms. The QoS analysis was conducted based on 12 features extracted from network parametric and streaming characteristics. With these feature sets our neural network model could easily identify a good and bad stream with high accuracy. The proposed approach can help service provider compare their services with their competitors without application-level details and player APIs. Service providers can also do feature analysis and fine-tune these features for better ABR strategy and flow control mechanism.

In the future, we would like to increase our dataset. With the increase of the dataset, we might have different results. We hope to build a dataset of at least 200 streams for each service. We will also explore better learning models like CNN and RNN in our future work.

Acknowledgment

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (2018-0-00749, Development of Virtual Network Management Technology based on Artificial Intelligence).

REFERENCES

[1] "ITU-T P.800.1 Mean opinion score (MOS) terminology," 2006

[2] "Quality of service regulation manual," https://www.itu.int/pub/D-PREF-BB.QOS_REG01-2017, 2017

[3] N. Seitz, "Standard Report: ITU-T QoS Standards for IP-Based Networks," IEEE Communications Magazine, Vol 41, Issue 6, pp. 82 – 89, June 2003

[4] "ITU-T P.1203.1 Models and tools for quality assessment of streaming media," 2019

[5] N. Barman, M. G. Martini, "QoE Modeling for HTTP Adaptive Video Streaming—A Survey and Open Challenges," IEEE Access, volume 7, pp 30831 – 30859 , 2019

[6] "ISO/IEC 23009-1:2014 Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats," https://www.iso.org/standard/65274.html

[7] "Youtube Player API," https://developers.google.com/youtube/iframe_api_reference

[8] D. Pal and T. Triyason, "A Survey of Standardized Approaches towards the Quality of Experience Evaluation for Video Services: An ITU Perspective," International Journal of Digital Multimedia Broadcasting, Vol 2018, pp. 24 – 25, 2018.

[9] P. Juluri, V. Tamarapalli, and D. Medhi, "Measurement of Quality of Experience of Video-on-Demand Services: A Survey," IEEE Communication Surverys & Tutorials, Vol. 18, no. 1, pp 401~418, 2016

[10] "Technical Document Key Network Delivery Metrics,Steaming media Alliance," May 2, 2016

[11] A. Bentaleb, B. Taani A. C. Begen C. Timmerer R. Zimmermann, "A Survey on Bitrate Adaptation Schemes for Streaming Media Over HTTP," IEEE Communications Surveys & Tutorials, Vol 21 , Issue 1 , pp. 562 – 585, 2019

[12] S. Akhshabi , A. C. Begen , and C. Dovrolis, "An Experimental Evaluation of Rate-Adaptation Algorithms in Adaptive Streaming over HTTP," Proceedings of the Second Annual ACM SIGMM Conference on Multimedia Systems, MMSys 2011, Santa Clara, CA, USA, February 23-25, 2011

[13] J. Kua, G. Armitage, and P. Branch, ''A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP,'' IEEE Commun. Surveys Tuts., vol. 19, no. 3, pp. 1842–1866, 3rd Quart., 2017

[14] Ashwin Rao et al "Network characteristics of video streaming traffic," CoNEXT '11 Proceedings of the Seventh Conference on emerging Networking EXperiments and Technologies, Article no 25, Tokyo, Japan, December 06-09, 2011.

[15] H. Nam, K. H. Kim, and H. Schulzrinne, "QoE Matters More Than QoS," IEEE INFOCOM 2016

[16] M. H. Mazhar and Z. Shafiq, "Real-time Video Quality of Experience Monitoring for HTTPS and QUIC," IEEE INFOCOM 2018

[17] M. Ghasemi , P. Kanuparthy , A. Mansy , T. Benson and J. Rexford, "Performance Characterization of a Commercial Video Streaming Service," Proceedings of the 2016 Internet Measurement Conference, November 2016 Pages 499–511

[18] Muhammad Jawad Khokhar Thibaut Ehlinger, Chadi Barakat, "From Network Traffic Measurements to QoE for Internet Video," IFIP Networking Conference (IFIP Networking), Warsaw, Poland, Poland, 2019, Page(s): 1 – 9

[19] T. V. Doan, V. Bajpai and S. Crawford, "A Longitudinal View of Netflix: Content Delivery over IPv6 and Content Cache Deployments," IEEE INFOCOM, 6-9 July 2020, Toronto, Canada.

[20] Chen Wang, Andal Jayaseelan, Hyong Kim, "Comparing Cloud Content Delivery Networks for Adaptive Video Streaming," IEEE 11th International Conference on Cloud Computing (CLOUD), 10 September 2018, San Francisco, CA, USA

[21] Y. Donga, J. Zhaoa , and J. Jinb "Novel feature selection and classification of Internet video traffic based on a hierarchical scheme," Computer Networks, vol 119, pp. 102~111, 4 June 2017

[22] W. Zai-jian,Y. Dong, H. Shi, Y. Lingyun and T. Pingping, "Internet Video Traffic Classification using QoS Features," 2016 International Conference on Computing, Networking and Communications (ICNC), Kauai, HI, USA, 15-18 Feb 2016

[23] T. N. Duc, C. M. Tran , P. X. Tan, and E. Kamioka "Convolutional Neural Networks for Continuous QoE Prediction in Video Streaming Services," IEEE Access, vol. 8, pp. 116268 – 116278, 22 June 2020

[24] https://suman17july.github.io/StreamingData/Streaming%20Data.zip

[25] A. Aloman, A.I. Ispas, P. Ciotirnae R. Sanchez-Iborra and M.D. Cano "Performance evaluation of video streaming using MPEG DASH, RTSP, and RTMP in mobile networks," 8th IFIP, Munich, Germany, 5-7 Oct. 2015.