8-27-2015

Cognitive Video Streaming

Devaki Rani Pasupuleti
devaki.pasupuleti@uconn.edu

Cognitive Video Streaming

Devaki Pasupuleti

B.E., Acharya Nagarjuna University,

2007

A Thesis

Submitted in Partial Fullfilment of the

Requirements for the Degree of

Masters of Science

at the

University of Connecticut

2015

APPROVAL PAGE

Masters of Science

Cognitive Video Streaming

Presented by

Devaki Pasupuleti,

Major Advisor

Krishna Pattipati

Associate Advisor

Balakumar Balasingam

Associate Advisor

Peter Willett

University of Connecticut

2015

DEDICATION

to

my lord

my family

and

Cyberlab

ACKNOWLEDGEMENTS

- I would like to sincerely thank my advisor Prof.Krishna R. Pattipati for his continuous guidance and excellent advice. I wish to submit to him my wholehearted gratitude for all the support and help I received from him.

- I owe many thanks to my associate advisor, Dr.Balakumar Balasingam for mentoring and for spending his valuable time in providing excellent guidance and feedback on my research at the University of Connecticut.

- I would also like to thank my associate advisors Dr.Marcus Baum and Prof. Peter Willett for providing feedback on my research.

- I would also like to thank David Sidoti, Diego Fernando Martnez Ayala for their guidance and support during my initial days at Cyberlab.

- Many thanks to Pujitha Mannaru, Manisha Mishra, Gopi Vinod Avvari, Niranjan Raghunathan, Balakishore Nadella, Rajeev Ghimire, Ali Abdollahi, Lingyi for their timely support and friendly attitude.

- Finally, I would like to express my sincere gratitude to my family for their love and encouragement in all my endeavors throughout my academic career.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

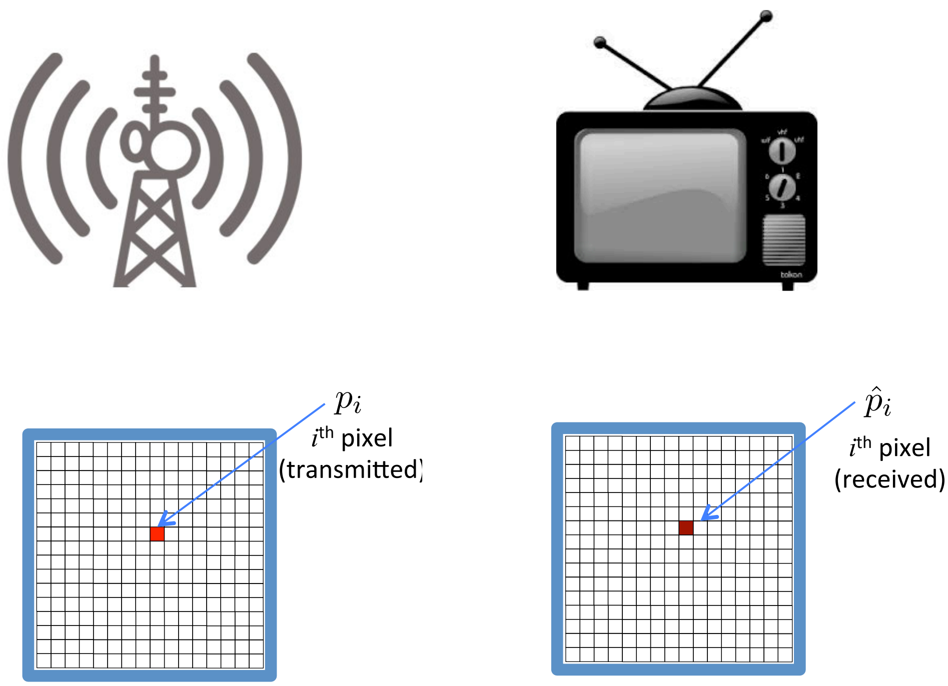
Introduction

1.1 Introduction

Major advances in wireless communication and consumer electronics in the past decade have disrupted the traditional ways in which people used to consume video programs. In a traditional setting, a viewer has to “tune-in” to a TV station via cable, satellite or on-air receivers in order to watch or record his/her favorite program. Today, with internet, smartphone and wireless broadband connectivity, there are several options for a viewer to watch his/her favorite programs at the time of his/her convenience using a device of choice, such as a smart phone, tablet or TV. As a result, the video distribution strategy also has gone through major changes. Figure 1.1 shows the typical elements of a traditional video broadcasting/viewing, and Figure 1.2 shows a block diagram of today’s interactive content delivery system.

A brief description of each of the blocks in Figure 1.2 is given below:

- *Content.* Content can be divided into online streaming, i.e., regular TV programs, and recorded programs that are delivered as video on demand (VoD), the focus



(a) Video transmission

(b) Video reception

Fig. 1.1: Traditional video transmission and reception. Traditional QoS metrics try to quantify viewers' perception using objective metrics computed based on transmitted and received frame sequences.

of this thesis. In VoD, a viewer browses through the lists of available videos and selects one to play. Unlike online streaming, VoD offers the capability to pause and resume videos at any time.

- *Delivery service.* Delivery service providers, such as cable networks, bring the videos to the viewers. Usually, the viewer has to be a subscriber to the delivery service provider in order to get access to most of the videos.

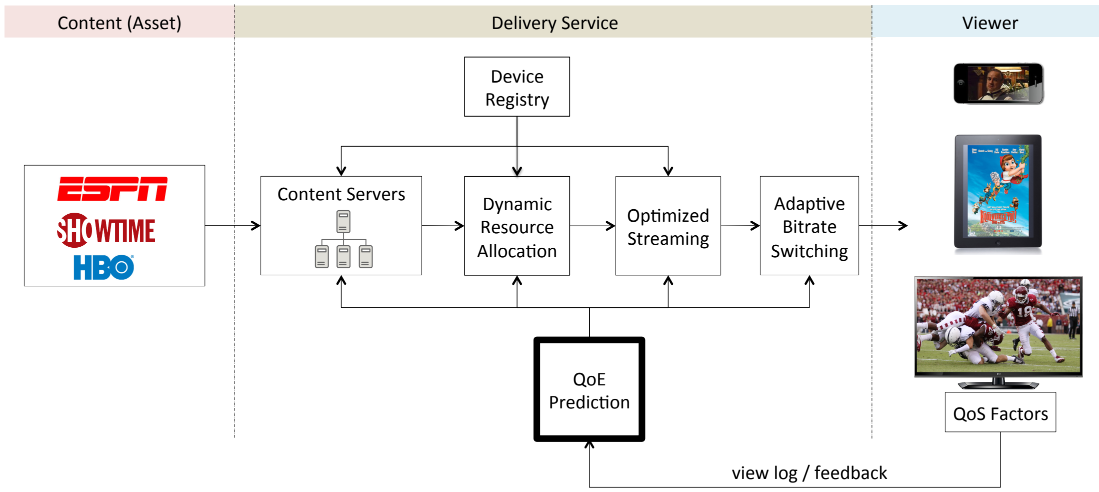


Fig. 1.2: Description of a video on demand (VoD) system. Unlike traditional video transmission systems, the viewers have the option of choosing from a large amount of video content or to select watching online video streaming.

- *Viewer.* The viewer accesses the videos using devices, such as smart phones, tablets, TV and Computer. Each viewing device may have different connectivity and bandwidth. Depending on the access device (portable or desktop), the characteristics of the viewer might be different as well. For example, a viewer might be willing to tolerate intermittent buffering events and longer startup times in a smart phone, while exhibiting lesser tolerance towards similar events in a TV.

- *Content servers.* Content servers respond to the VoD requests and stream videos to the viewers. Based on the popularity of particular videos, content servers adjust priorities in order to provide good QoS to the viewers.

- *Dynamic resource allocation.* Content service providers respond to rapidly increasing/decreasing demands to particular videos, anticipated and unexpected, such as major sports events and unexpected world events, by dynamically adjusting the streaming capacity of videos.

- *Optimized streaming.* Optimized streaming algorithms aim to deliver high quality videos at reduced cost (bandwidth) to the viewer. This is achieved by efficiently compressing subsequent video frames. Some other constraints include the power and memory requirements of the video player at the viewing devices.

- *Device registry.* An important challenge in maintaining superior quality of online video streaming is the increasing number of different types of devices available to viewers in order to play videos. Each of these devices has different hardware and software capabilities. Knowing the exact capabilities of a particular device is important to optimize the video streaming.

- *View logs.* These represent feedback data from the video players to the content delivery service providers. The feedback contains features, such as bit rate, buffering information and media-failed events, that are useful in assessing the quality of experience of the viewer.

- *Adaptive bitrate switching.* In mobile video devices, the available bandwidth can vary based on the location of the receiver. For example, movement of the device (e.g., moving between different parts of a house, traveling in a vehicle, walking

through a mall, etc.), can result in varying download bandwidths at the device. The video streaming algorithms respond to this by adjusting the bit-rate of the content.

The quality of user experience has been a concern in both traditional and the emerging content delivery systems. In traditional video broadcasting scenario, the issue of video quality arises due to video transmission and processing; these manifest in the form of noise, jitter, shape distortion, etc. Traditional QoS assessment schemes focused on quantifying the perception of the viewers on videos with varying types and degrees of video transmission distortions; such distortions are generally defined in terms of QoS metrics, such as peak signal-to-noise ratio (PSNR, [43]), video quality metric (VQM, [37]), moving pictures quality metric (MPQM, [41]), structural similarity index (SSIM, [44]), and noise quality measure (NQM, [15]). The viewers' perceptions as a result of different QoS metrics are obtained through subjective methods and quantified usually as a mean opinion score (MOS, [25]). The MOS is a number, usually between 0 an 5, indicating the perceptual quality of the video; the highest number indicating very pleasant and clear video and the lowest number indicating intolerable video. When poor QoS is detected in some areas, the broadcasters had to find ways to increase the signal to noise ratio to the affected area; this can be achieved by increasing the power of existing transmitters or by installing additional transmitters (or repeaters) to the affected area. The MOS metric in traditional TV broadcasting enjoys wide acceptability (see [22])

In VoD, the QoS factors are different from that in a traditional video setting; some

widely used QoS factors are based on startup time, buffering and transmission bitrate. The startup time is defined as the time between the initial video request (such as clicking on the play button on a web interface) and the time the first video frame is played on the screen. Higher startup time causes the viewer to abandon the video [28]. There are several factors affecting the startup time; connection bandwidth of the viewer, capability of the video distribution server, and network delays are a few of them. In order to reduce the startup time, the player "buffers" a portion of the video before it starts and the rest of the video is buffered while the video is still playing. Buffering is supposed to happen in the background while the video is playing; however, similar to startup time, non-ideal streaming conditions cause the player to pause and wait for data to be buffered. It is reported in [28] that buffering delays negatively impact the likelihood of a viewer's return to the content provider. Adaptive bitrate switching [30] allows one to reduce the startup and buffering delays by adaptively switching the frame quality of the video based on the bandwidth and other hardware capabilities of the video player; The higher the bandwidth and processing capabilities of the player, the higher the bit-rate and quality of the video; the bitrate thus serves as a QoS factor. High average bitrate over a certain period of time indicates that the rendering quality was high and vice versa; frequent bitrate switching with high variation indicates poor quality of experience due to volatile bandwidth. Analysis of viewer responses to the startup time, buffering and bitrate related QoS factors are reported in [16]. The adaptive bitrate streaming technique has been widely adopted by many existing content providers;

in [36], a general overview of the widely adopted HTTP adaptive streaming (HAS) protocol [26] is provided.

Adaptive video streaming itself is a challenging problem and diverse approaches have been published in the literature [40]. Most of the adaptive streaming strategies involve adapting the bitrate uisng metrics derived based on buffering events [19]. Other than adaptive streaming, there are approaches focused on enhancing a specific aspect of QoE; in [4], an approach is suggested to enhance the accessibility in shared video forums; [5] exploits concurrent viewers and use of peer-to-peer P2P in order to offload some of the workload of the content servers; an approach for client side server selection is presented in [31]; in [39] the QoE is modeled based on a packet loss model; in [42], the QoE is modeled in terms of the QoS factors, such as loss, delay and jitter; and [12] talks about providing good quality video, while being aware of the bandwidth quota of the user.

Current adaptive streaming and other approaches developed to enhance QoE are designed to “react” to the QoS factors (that are largely based on startup time, buffer level and average bitrate) from the viewer’s device. This does not guarantee that the quality of experience (QoE) of the viewer will be improved as a result. For example, the decision to downgrade the bitrate (i.e, the quality of the video) as a result of buffering delay may not be appreciated by all viewers; to make things worse, the same viewer may have varying preferences depending on context, such as the time of the day. Further, there is explosive growth in the internet traffic caused by videos delivered

by content delivery networks; this trend is expected to accelerate as more and more viewers are expected to switch from traditional TV to VoD [1]. Expanding the network infrastructure is costly and time consuming; a QoE based adaptive streaming will help ease some of the strain on the network by increasing the bitrate only when it is likely to advance the QoE of the viewer. In other words, a better and futuristic adaptive streaming technique needs to be "proactive" rather than reactive.

The first step in QoE based adaptive video streaming is to come up with accurate methods of estimating the QoE of the viewer. Taking cues from the widely adopted MOS in traditional TV, some initial attempts were made in [34] to estimate the MOS in response to the QoS factors of VoD. However, unlike traditional video, the MOS obtained through a limited experiment is unable to represent the viewers' perception in a wide ranging VoD scenario. It is found that the viewers react differently to the same video content with the same QoS factor; viewers seemed to tolerate QoS deficiencies in live video compared to non-live content [7]; viewers from well connected devices (those with better connection bandwidth) are found to be less tolerant compared to their low-bandwidth counterparts. A VoD viewer has millions and millions of videos to choose from. Instead of traditional TV, there are devices of convenience (with trade offs) for a particular time of day; video in a smart phone might come with too many buffering events and blurry images compared to a TV; however, its portability is appealing to a certain viewer during day-time; the same viewer might prefer to continue the same video using TV during the evening. For content providers, the objective has become

one of attracting and retaining subscribers by providing superior quality of experience. Due to the nature of VoD consumption, it is impossible to capture the QoE in terms of a single metric, such as MOS. Hence, there has been significant effort in the recent past in developing approaches aimed at quantifying the QoE of a certain viewer. Indeed, MOS, which is subjectively estimated using a particular viewing scenario, is not adequate to quantify viewers' QoE [11].

Recently, there have been other attempts to estimate QoE; these approaches are generally termed "passive", "online" or "indirect" approaches for estimating QoE. In [6, 7], it was suggested to create a predictive model of viewer engagement (such as total play time, number of visits and probability of return) based on the observed QoS features (attributes). A machine learning framework to estimate the QoE in mobile applications was proposed in [3]; this approach requires training data form past "good QoE" and "poor QoE" instances. Table 1.1 gives a comparative summary of existing QoS literature corresponding to traditional video transmissions and QoE metrics corresponding to VoD and internet video.

Table 1.1: Summary of QoE Approaches in Traditional TV and VoD

	Traditional Video	**VoD**
QoS factors	PSNR – Peak Signal to Noise Ratio [43], VQM – Video Quality Metric [37], MPQM – Moving Pictures Quality Metric [41], SSIM – Structural Similarity Index [44], NQM – Noise Quality Measure [15]	Startup time [16], Buffering time [38], Buffering count [38], Buffering ratio [16], Rate of buffering events [16], Normalized re-buffer delay [27], Average bit rate [16], Average throughput [36], Frames per second (FPS) [16], Failures [27]
User satisfaction metrics (alternately, Viewer behavior metrics [27])	Mean opinion score (MOS) [25]	MOS [34] Number of views [16], Total play time [16], Session duration ratio [38], Abandonment [27], Engagement [27], Repeat viewers [27]
Related Standards	For cable TV (2004) [22], For standard television (2004) [20], For multimedia applications (2008) [24], Relative to reduced bandwidth reference (2008) [23], Television (2002) [21], Multimedia (2008) [25]	DASH [26] 3GP-DASH [2]

The existing approaches focus heavily on modeling the QoE as related to the QoS factors only. Even though the QoE is significantly influenced by the QoS factors, there could be other factors that wield influence on the QoE of the viewers. For example, considering the vast amount of video contents to choose from, the viewers' QoE can be be influenced by the type of content being accessed. Further, for a given video content, different viewer demography will have different QoE, based on their age, gender, ethnic background and spoken language. In addition to this, the seasonal factors, such as the time of day, day of week and season of year, also affect the QoE of the user towards a particular video content. Finally, there could be many other exogenous factors, such as important local/national/world events, that might contribute to the QoE of a particular viewer. In the next Section, we describe our proposed cognitive video streaming strategy, which considers all the above factors in devising a video streaming strategy.

1.2 Organization of Thesis

The thesis is organized as follows. The proposed cognitive video streaming architecture is described in Chapter 2. In Chapter 3, we develop and demonstrate an online play time prediction tool (PPT), which is a part of the proposed cognitive video streaming architecture. We develop several machine learning algorithms for online playtime prediction and demonstrate their merits using video consumption data provided by Comcast Corporation.

1.3 Publications

[1] D. Pasupuleti, B.Balasingam, M.Baum, K. Pattipati and P. Willett, “Cognitive Video Streaming”, *2nd International Conference on Electrical, Electronics, Engineering Trends, Communication, Optimization and Sciences (E3COS)*,Vijayawada, India, March 2015 (published).

[2] D. Pasupuleti, P. Mannaru, B. Balasingam, M. Baum, K. R. Pattipati, P. Willett, C. Lintz, G. Commeau adn F. Dorigo, and J. Fahrny, “Online Playtime Prediction for Cognitive Video Streaming,” *International Conference on Information Fusion (FUSION 2015),* Washington, D.C, July 2015 (published).

[3] D. Pasupuleti, P. Mannaru, B. Balasingam, M. Baum, K. R. Pattipati, P. Willett, C. Lintz, G. Commeau adn F. Dorigo, and J. Fahrny, “Cognitive Video Streaming,” *Journal of Advances in Information Fusion* (submitted).

Chapter 2

Cognitive Video Streaming

2.1 Cognitive Video Streaming

A block diagram of the proposed cognitive video streaming approach is shown in Figure 2.1. It is comprised of three fundamental modules: an *estimation module*, a *prediction module* and an *adaptation module*. The framework is designed in such a way that each module is able to function with some basic functionalities (sub-modules); as more sub-modules are added, the effectiveness of the module and the integrated system is expected to improve. Next, we describe each module in the proposed solution framework.

2.1.1 Prediction Module

It is observed that most of the opened videos are not watched in their entirety by the viewers. The nature of completion of a particular video changes from viewer to viewer; some videos are abandoned in the process of "browsing"; some videos are terminated by the viewer because of lengthy buffering and other QoS issues; and some videos are

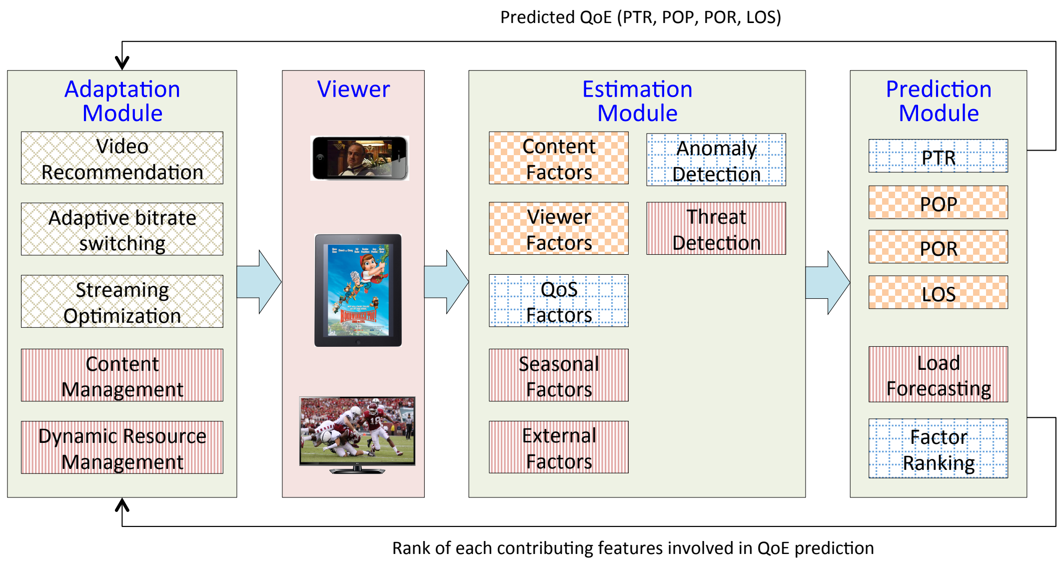


Fig. 2.1: Proposed Cognitive Video Streaming Architecture.

"temporarily" abandoned to be resumed later. Once a viewer starts playing a video, knowing the *remaining play time* of that video is a useful piece of information to the content provider in order to ensure adequate QoE to the viewer. For example, the knowledge of the remaining play time can be used to allocate server bandwidth to the user; it can be used to devise a more appropriate adaptive bitrate switching scheme; and the prior knowledge that a video is possibly terminated by the viewer can be used to recommend more appropriate videos in the first place. At the network level, the predicted play time of each viewing session is useful for predicting and managing network traffic.

In addition to QoS, there are several other factors determining the play time ratio (PTR, with respect to the length) of a particular video. For example, shorter videos

tend to have higher play time ratio (PTR) compared to longer videos [28]; QoS factors, such as buffering, negatively affect the PTR in well connected devices. All the relevant factors must be included in order to accurately predict the play time of a video. We divide the factors affecting PTR into five categories: content related, viewer related, QoS related, seasonal and external factors. Each factor contains several features affecting the play time; in Table 2.1, we have provided some examples.

Table 2.1: Factors Affecting Play-Time Prediction and Sample Features in Each Factor

Factor	**Features**
Content	popularity, age, length, match to viewer's preference
Viewer	age, gender, ethnic background, language
QoS	startup time, buffering, average bitrate, throughput
Seasonal	time of day, day of week, season of year
External	important local/national/world events

Considering all the relevant factors/features helps in accurately predicting the PTR of a particular video session. This also allows us to investigate the features that are significant to PTR prediction. It must be noted that the dominant factor affecting play time will be different from one viewer to the next. Identifying these factors (even after knowing that a particular video has been terminated) will help in devising individualized remedies.

Similar to PTR, there are other user engagement metrics that are indicative of

the QoE of a viewer; the ***probability of return (POR)*** tells if the viewer will return to a previously abandoned video; the ***probability of re-play (POP)*** tells if the viewer will re-play a previously completed video; and the ***average length of scrubbing (LOS)*** tells how long a particular video will be "scrubbed", i.e., skimmed for its contents. Developing the ability to understand and predict all the user engagement metrics will help in developing an adaptive streaming method that is responsive to the QoE of the individual viewer (instead of just the QoS factor of a viewer's device). Another important system variable is load; indeed, ***load forecasting*** algorithms will be useful in dynamic resource allocation. In [8], we experimented with Neural networks [32, 35], Nearest neighbor classifiers [33], and Survival modeling [14] techniques in developing a PTR prediction tool. These will be useful in developing the proposed system and the concomitant user-centered QoE prediction models.

2.1.2 Estimation Module

The objective of the estimation module is to infer and provide the features required by the predictive module. In addition, the estimation module is responsible for ***anomaly detection***; anomaly detection [9] is important for accurate feature extraction, security threat detection and QoE monitoring. Another important functionality of this module is ***threat detection***; detecting threats is more challenging than detecting anomalies. The most effective threat detection combines informative features from both anomaly based and signature based approaches; understanding of normal (and possibly abnormal) sig-

natures is crucial in devising an effective threat detection strategy.

2.1.3 Adaptation Module

The adaptation module consists of the following important sub-modules:

Video recommendation: Video recommendation is an indirect way of improving the QoE of a viewer.

Adaptive bitrate switching: Adaptive bitrate switching strategy helps in achieving uninterrupted play of the video regardless of fluctuating bandwidth (mostly on the user's side).

Streaming optimization: Streaming optimization aims to achieve the most economic usage of bandwidth; for now, we won't be focusing on developing any new algorithm for streaming optimization; rather, existing stream optimization tools will be exploited.

Content management: Content management is required to respond to uneven and unexpected demand of particular video content at particular times.

Dynamic resource management: Dynamic resource allocation [17] helps in optimizing the resources, such as server bandwidth and content, in a way that a guaranteed QoE can be maintained across all (the tens of millions of) subscribers.

2.2 Conclusions

In this chapter, we proposed a cognitive video streaming architecture that is able to keep the entire swath of video on demand (VoD) customers at maximal satisfaction by offer-

ing any video, anytime, anywhere at high quality of experience (QoE) standards; this is achieved by cognitive learning of different features that potentially affect the QoE, such as the quality of service (QoS), suitability/quality of the video content and the viewing behaviors of subscribers. The outcome of such learning is exploited to predict several QoE related factors, such as play time ratio, probability of return, probability of replay and average LOS. The predicted QoE factors are used to adaptively change the content delivery strategy through adaptive bit-rate streaming, streaming optimization, content management, dynamic resource management and video recommendation.

The proposed cognitive video streaming architecture consists of several functional blocks categorized under three modules: the prediction module, the estimation module and the adaptation module. The next chapter of the thesis concentrates on the play time prediction function, which is found under the prediction module of the cognitive video streaming architecture.

Chapter 3

Online Playtime Prediction for Cognitive Video Streaming

3.1 Introduction

In this chapter, we develop a play time prediction scheme which is a part of the prediction module of *cognitive video streaming* (see Figure 3.1) architecture that we introduced in Chapter 2. Prediction of play time ratio (PTR) plays a significant role in understanding the user engagement.

The rest of the chapter is organized as follows: The proposed playtime prediction tool is introduced in 3.2. Section 3.3 describes the datasets used in the experimentations. Three approaches for playtime prediction are summarized in Section 3.4. Numerical results based on collected data are presented in 3.5 and the results of our work is concluded in Section 3.6.

3.2 Playtime Prediction Tool (PPT)

Figure 3.2 shows a typical sequence of events in a viewing session. The session starts when the viewer requests a video. The request may go through an authentication pro-

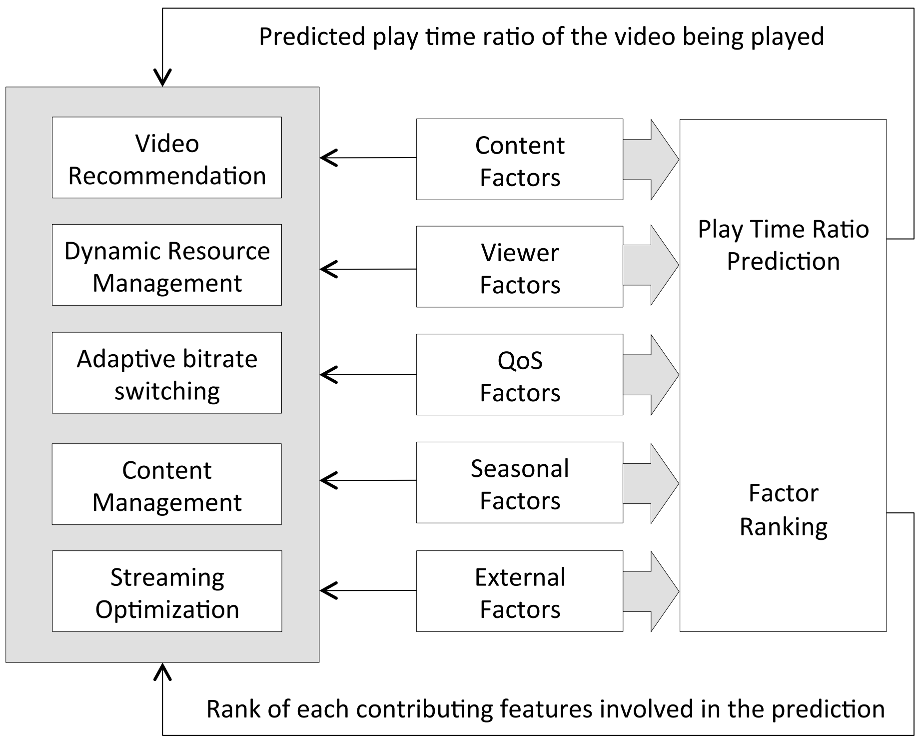


Fig. 3.1: Cognitive Video Streaming Architecture.

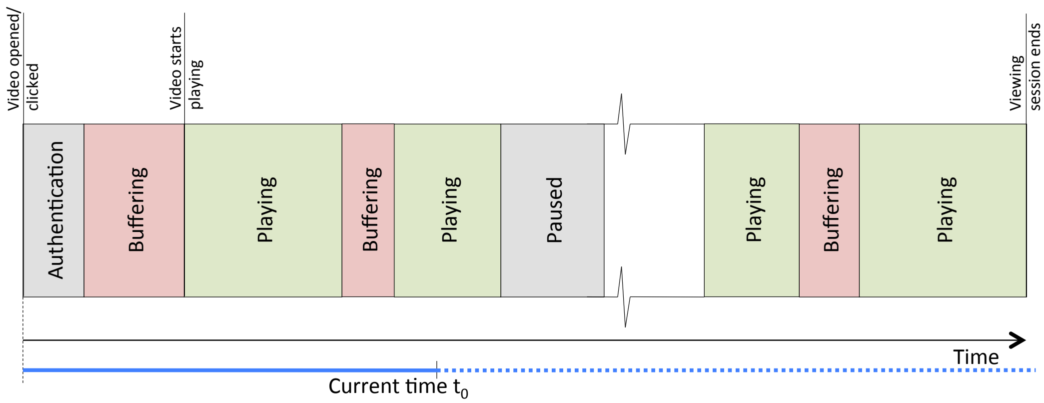


Fig. 3.2: Typical video viewing session. The purpose of the play time prediction tool (PPT) is to estimate the remaining playtime at the current point in time t_0.

cess for non-public videos and then, the video starts buffering into the local player. The amount of video being buffered (before the first video frame starts playing) depends on various factors such as the player or the bandwidth. Once a certain portion of the video buffer is filled, the video starts playing in the local player. If the streaming rate is poor, the video player might be forced to temporarily stop playing the video due to an empty buffer. As soon as the buffer is filled again, playing resumes. Nowadays, most streaming protocols use adaptive bitrate switching – meaning the bitrate is adapted dynamically in order to get the best possible video quality for the current bandwidth. The viewing session ends when the entire video is finished playing or when the viewer actively closes that video.

3.2.1 Functionality of the Playtime Prediction Tool (PPT)

In this work, we aim at developing an online *playtime prediction tool (PPT)* that successively estimates a prediction of the remaining playtime in a viewing session, see Figure 3.2. Technically, the tool may run on both the client side or the server side.

To the best of our knowledge, there is no work yet on an online prediction of the session playtime based on an ongoing session. The most similar work [16] aims at developing methods for predicting the playtime of completed sessions.

The PPT presented in this work is the first step in creating a tool that forecasts the entire set of events of a session.

3.2.2 Used Data

In order to perform playtime prediction, the tool exploits protocol data reported by the video player. Typically this data contains high-level information about the video session, such as in Figure 3.2. Content related features, e.g., the popularity of the video, also play an important role. A detailed description of the features used in this work will be given in Section 3.3.

3.2.3 Methods

Realistically, the playtime can usually not be predicted exactly due to random effects and unknown circumstances (e.g., power outages or user behavior). As a remedy, we aim at creating a playtime predictor that performs well in an "average" sense. For

this purpose, we propose to exploit previously logged protocol data in order to "learn" a predictor using machine learning algorithms. In general, it is possible to learn a playtime predictor for both specific users, VoD assets, or the entire data set.

3.2.4 Benefits

The PPT is of high value to the content provider. First and foremost, it allows the content provider to react before the session is terminated. For example, the content provider can enact counter measures to increase the service quality or recommend alternative contents. Even if the PPT predicts a long playtime, the content provider in general could decrease the quality of service to minimum acceptable level.

Second, the learned playtime prediction model encodes important information about the viewer behavior (of the entire population or even a specific viewer). For example, it is possible to perform a diagnosis that gives the most relevant features that influence the playtime. Also, a playtime prediction model allows for detecting a change in user behavior.

Last but not least, playtime is a very strong indicator of the QoE. Intuitively, if the QoE is bad, the playtime will be low, too. And if the playtime is long, the QoE cannot be that bad. Hence, a model for the playtime will always be a significant part of a QoE model. In this sense, content providers are interested in increasing the playtime, i.e., the user engagement.

All told, the PPT has substantial impact on improving the overall QoE of video

streaming.

3.3 Comcast Data Set: Features

Our work is based on a data set from the video-on-demand streaming service *Xfinity On Demand* from *Comcast*. The available data was logged by the video players and consists of a sequence of events that are attached with time stamps, device ids, and further information. Specifically, we use the following logged events:

- *Opening:*

  Indicates that a new viewing session is opened. An asset id is available for identifying the content.

- *Playing:*

  Video starts playing.

- *Buffering:*

  The player starts buffering; the video is not played anymore.

- *Paused:*

  User triggered pausing of the video.

- *Closing:*

  Video stopped because of a user triggered event or the stream ends.

- *Bitrate switched:*

  Indicates that the bitrate is changed and gives the new bitrate.

We define a viewing session as the events between the opening and closing events of a particular device. Based upon the above described events, we determine the following session features that potentially affect the playtime and the QoE.

QoS related features are session attributes that are dynamically collected during an ongoing viewing session. Hence, these features depend on the time t_0 that elapsed since the opening event.

- *Total play time until t_0:* $\mathrm{PLT}(t_0)$
- *Total pause time until t_0:* $\mathrm{PAT}(t_0)$
- *Total buffering time until t_0:* $\mathrm{BUT}(t_0)$
- *Average bitrate until t_0:* $\mathrm{BR}(t_0)$
- *Number of paused events until t_0:* $\mathrm{NRP}(t_0)$
- *Number of buffering events until t_0:* $\mathrm{NRB}(t_0)$
- *Startup time (time span from opening to first playing event):* STT
- *Average Frame Rate:* $\mathrm{FR}(t_0)$
- *Buffering ratio until t_0*: $\mathrm{BUR}(t_0) = \frac{\mathrm{BUT}(t_0)}{\mathrm{PLT}(t_0)+\mathrm{BUT}(t_0)}$

Of course, there may be strong correlations among these features. For example, number of buffering events and total buffering time are probably correlated. For the buffering ratio and the playtime, there is even a functional relationship. It is part of this work to figure out which features are best at predicting the playtime.

3.4 Methods for Play-Time Prediction

In this section, we introduce several approaches for playtime prediction at a single specific time t_0. Hence, we can omit t_0 in the notations in the remainder of this chapter.

3.4.1 Linear Least Squares Prediction

A simple prediction model of playtime is a linear combination of the observed features:

$$y_i = \sum_{n=1}^{N_x} k_n x_{i,n} \tag{3.1}$$

where $x_{i,n}$ is the n^{th} observed feature corresponding to the i^{th} viewing session, and y_i is the playtime. The parameter vector $\mathbf{k} = [k_1, k_2, \ldots, k_{N_x}]$ can be estimated by collecting the observation pairs $\{y_i, \mathbf{x}_i\}$ where $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,N_x}]^T$ for $i = 1, \ldots, M$, i.e.,

$$\hat{\mathbf{k}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1} \mathbf{X}^T\mathbf{y} \tag{3.2}$$

where $\mathbf{y} = [y_1, y_2, \ldots, y_M]^T$ and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M]^T$.

For a given observed feature $\mathbf{x}_j = [x_{j,1}, x_{j,1}, \ldots, x_{j,N_x}]^T$, the predicted playtime is given as

$$\hat{y}_j = \hat{\mathbf{k}}^T\mathbf{x}_j \tag{3.3}$$

The linear prediction is useful as a benchmark for comparison against other non-linear approaches described later.

3.4.2 K-Nearest Neighbor Method

In the k-nearest neighbor approach, the target and feature pairs $\{\mathbf{y}, \mathbf{X}\}$ are kept as training-data. Given the observed feature $\mathbf{x}_j$, first, the following distance metric is computed

$$d_{i,j} = \mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) \tag{3.4}$$

where $\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j)$ is a distance measure between the arguments $\mathbf{x}_i$ and $\mathbf{x}_j$. Let $\mathbf{y}^k$ correspond to the play time of the first k of the smallest distance measures. Now, $\hat{y}_j$ is obtained in two different ways: (i) mean of $\mathbf{y}^k$, (ii) median of $\mathbf{y}^k$. The median is robust to anomalies and outliers.

3.4.3 Survival Models

Survival modeling has found a wide range of applications in a number of areas, including medicine [14] and equipment failure analysis [29]. Survival modeling was employed to derive a QoE metric in [13]. In this section, we briefly describe how survival models can be used for playtime prediction.

Let ξ be the time of termination of a particular video. The probability density function of ξ can be written as

$$P_\xi(t) \triangleq f(t) \tag{3.5}$$

where $f(t)$ is also known as the *survival density function*.

The cumulative probability distribution function of ξ

$$F(t) = P(\xi \leq t) = \int_0^t f(u)du \tag{3.6}$$

is the fraction of videos terminated by time t. The remaining (still playing) portion of videos is given by

$$R(t) = P(\xi > t) = 1 - F(t) \tag{3.7}$$

where $R(t)$ is also known as the *reliability.*

Given that a video has survived until time t, it is often of interest to know the probability that it will be terminated in the interval [t,t+Δt]. This is given by h(t)Δt, where

$$h(t) = f(t|\xi > t) = \frac{f(t)}{R(t)} \tag{3.8}$$

denotes the instantaneous risk or *hazard* of the system. It must be noted that as t increases, the hazard (or the risk of being terminated) increases. Let us rewrite (3.8) as

$$h(t) = \frac{f(t)}{1 - F(t)} = \frac{F'(t)}{1 - F(t)} = -\frac{R'(t)}{R(t)} \tag{3.9}$$

Integrating both sides of (3.9)

$$-\int_0^t h(u)du = \ln R(t) \tag{3.10}$$

Hence,

$$R(t) = \exp\Big\{ - H(t) \Big\} \tag{3.11}$$

where $H(t) = \int_0^t h(u)du$ is the *cumulative hazard function.*

Using (3.7) and (3.11)

$$\begin{aligned} 1 - F(t) &= \exp\Big\{ - H(t) \Big\} \\ f(t) &= h(t) \exp\Big\{ - H(t) \Big\} \end{aligned} \tag{3.12}$$

So far it was assumed that $f(t)$ (and hence $R(t)$ and $h(t)$) are all functions of time only. However, all of these functions are dependent on features $\mathbf{x} = \{\mathbf{x}_i\}$, or *covariates*. The *proportional hazard function*, proposed by Cox [14], suggests to separate the time-dependent and feature-dependent hazards as follows:

$$h(t, \mathbf{x}) = \lambda(t) \exp\left\{\mathbf{b}^T\mathbf{x}\right\} \tag{3.13}$$

where $\lambda(t)$ is the baseline time-dependent hazard function, x_i is the covariate, and b_i is the coefficient corresponding to the i^{th} covariate, x_i.

Now, (3.11) and (3.12) are rewritten as

$$f(t) = \lambda(t) \exp\Big\{\mathbf{b}^T\mathbf{x} - \Lambda(t)e^{\mathbf{b}^T\mathbf{x}}\Big\} \tag{3.14}$$

$$R(t) = \exp\Big\{ - \Lambda(t)e^{\mathbf{b}^T\mathbf{x}}\Big\} \tag{3.15}$$

where $\Lambda(t) = \int_0^t \lambda(u)du$. Cox suggested that the the model parameters $\mathbf{b}$ can be estimated independent of $\lambda(t)$ by maximizing the partial likelihoods. Once $\mathbf{b}$ is estimated, there are several approaches in the literature to model and estimate (the parameters of) $\lambda(t)$.

Once the parameters are estimated, the remaining play time at time u can be computed as

$$\hat{y}_j(u) = \frac{\int_u^{\infty}(t-u)f_j(t)dt}{R_j(u)} \tag{3.16}$$

where $f_j(t)$ and $R_j(u)$ are obtained by substituting $\mathbf{x}_j$ for $\mathbf{x}$ in (3.14) and (3.15), respectively, and u is the time elapsed.

An advantage of the survival model-based approached described above is that the playtime prediction can be updated as the video progresses. We assume $\lambda(t) = \lambda$.

3.4.4 Neural Networks

The playtime can be modeled as a function of the observed features using artificial neural networks (e.g., multi-layer perceptrons)

$$y_i = f(\mathbf{x}_i, \{w_{l,k}\}_{l=1,k=1}^{N_L,N_h}) \tag{3.17}$$

where $w_{j,k}$ are different weights and N_L is the number of layers and N_h is the number of hidden nodes. Given a set of (past) training data $\mathbf{y}, \mathbf{X}$, there are several approaches to learn the weights [18]. A trained neural network can be used to predict the playtime for a given feature set $\mathbf{x}_j$.

3.5 Simulation Studies

In this section, we evaluate the proposed approaches using data from 8808 viewing sessions lasting up to 8 minutes. We focus on the first 8 minutes as we try to detect

early termination of video due to the low streaming quality. All the viewing sessions occurred on the same day. 50% of this data is randomly selected and denoted as the "training" data set, and the rest is kept for testing. Each feature in the testing data is used for predicting its playtime. This procedure is repeated for 10 Monte-Carlo runs (note that portion of the data for learning is selected randomly in each run).

3.5.1 Data Analysis and Visualization

The following features are used in our current analysis: number of buffering events, number of paused events, inter buffering time, startup time, average bit rate, and buffering ratio. Figure 3.4 shows the histogram of playtime and Figure 3.3 shows histograms of corresponding features.

3.5.2 Performance Metrics

In this section, we use the algorithms introduced in Section 3.4 for playtime prediction and assess their performance. Due to lack of knowledge about any statistical properties of the playtime, it is important to use several, relevant metrics for the assessment of playtime prediction. We use the following four metrics for this purpose.

Normalized Mean-Squared Error (NMSE)

This metric gives insight about the error in playtime prediction and is given by

$$\mathrm{NMSE} = \frac{1}{M}\sum_{i=1}^{M}\left(\frac{y_i - \hat{y}_i}{y_i}\right)^2 \tag{3.18}$$

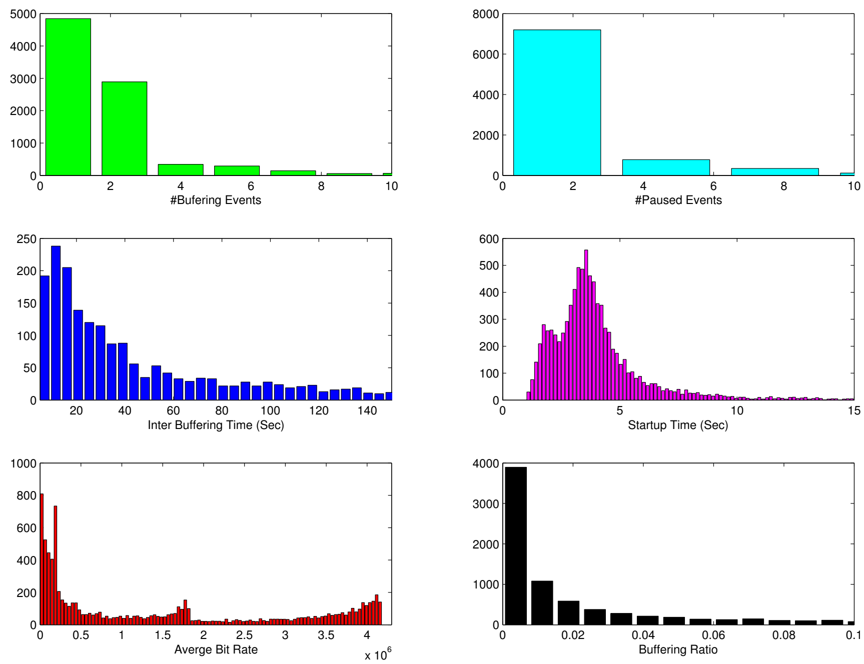


Fig. 3.3: Histogram of features.

R^2 Fit

The coefficient of determination, R^2, gives insight into how well the data points fit the statistical model used for playtime prediction. A value of $R^2 = 1$ indicates perfect fit and smaller the R^2, the poorer the fit is.

$$R^2 = 1 - \frac{\sum_{i=1}^{M} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{M} (y_i - \bar{y})^2} \tag{3.19}$$

where $\bar{y} = \frac{1}{M} \sum_{i=1}^{M} y_i$.

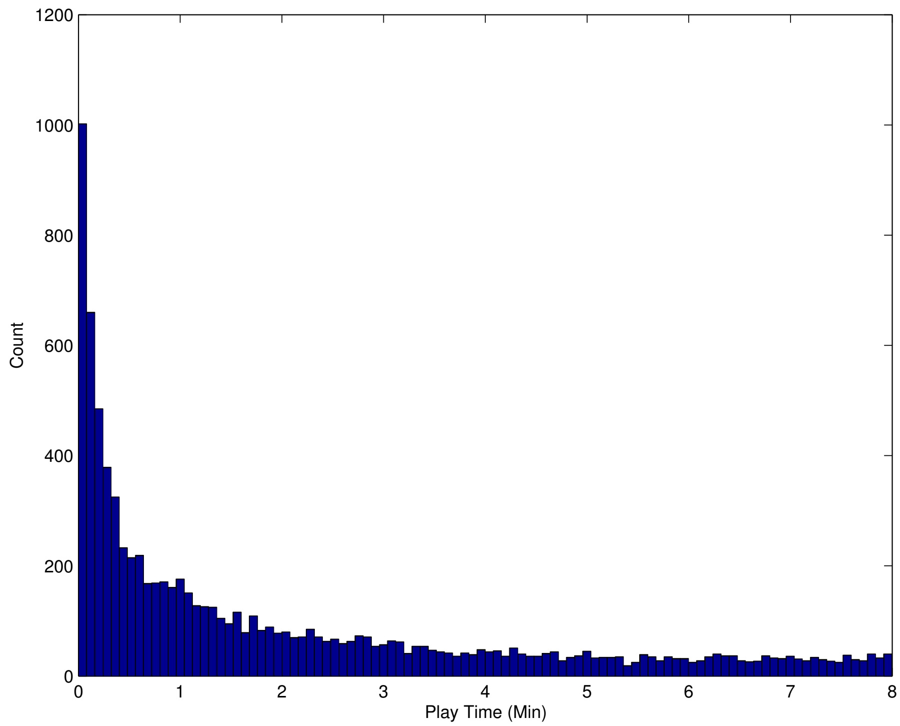


Fig. 3.4: Histogram of playtime.

Ratio of Predicted and True Greater than r

The playtime is a quantity that can generally vary anywhere from less than 1 minute to several minutes. A prediction error of 1 min is significant if the actual play time is 5 min; however, it is not so significant if the actual play time is 120 minutes. The NMSE captures this through normalization; however, the following metric captures this error in a different light.

$$\mathrm{RG}(r) = \frac{\#\left\{\frac{\hat{y}_i}{y_i} > r\right\}}{M} \tag{3.20}$$

where $\#\left\{\cdot\right\}$ denotes the number of times the argument is true.

Ratio of Predicted and True Less than $1/r$

Similar to $\mathrm{RG}(r)$, the following metric captures the ratio of instances when the prediction was significantly smaller than the true value of playtime.

$$\mathrm{RL}(1/r) = \frac{\#\left\{\frac{\hat{y}_i}{y_i} < \frac{1}{r}\right\}}{M} \tag{3.21}$$

3.5.3 Feature Selection

With N features there are $2^N - 1$ possible subset of features. Although it might be thought that more is better, in machine learning one can be subject to the "curse of dimensionality": extra features that are uninformative actually hurt performance by "fitting the noise". In Figures 3.5, 3.6, 3.7 and 3.8, we show the performance(s) plotted against binary representation of feature combination, from 1 to $2^N - 1$. Each time,

half the dataset is randomly selected and used for learning and the playtime prediction is performed on the rest of the data. This procedure is repeated for 10 Monte-Carlo runs (This is called 10×2 cross validation.) and the median of each of the metrics is plotted in Figures 3.5–3.8. There are six subplots in each of Figures 3.5–3.8, showing the results of different playtime prediction approaches: Survival modeling, k-nearest neighbor (mean), k-nearest neighbor (median), LS, neural networks and random. In the "random" approach, we randomly select a playtime from the training data set.

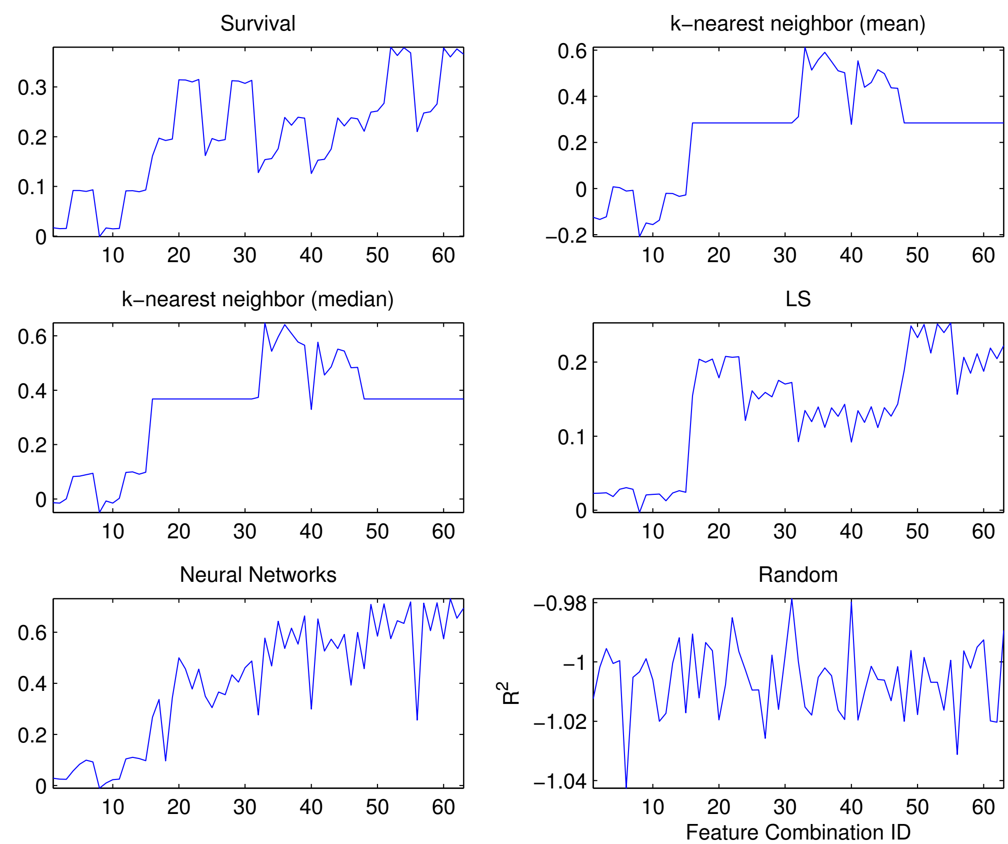


Fig. 3.5: Median R^2 Fit.

Next we select just one playtime prediction approach shown in Figures 3.5–3.8

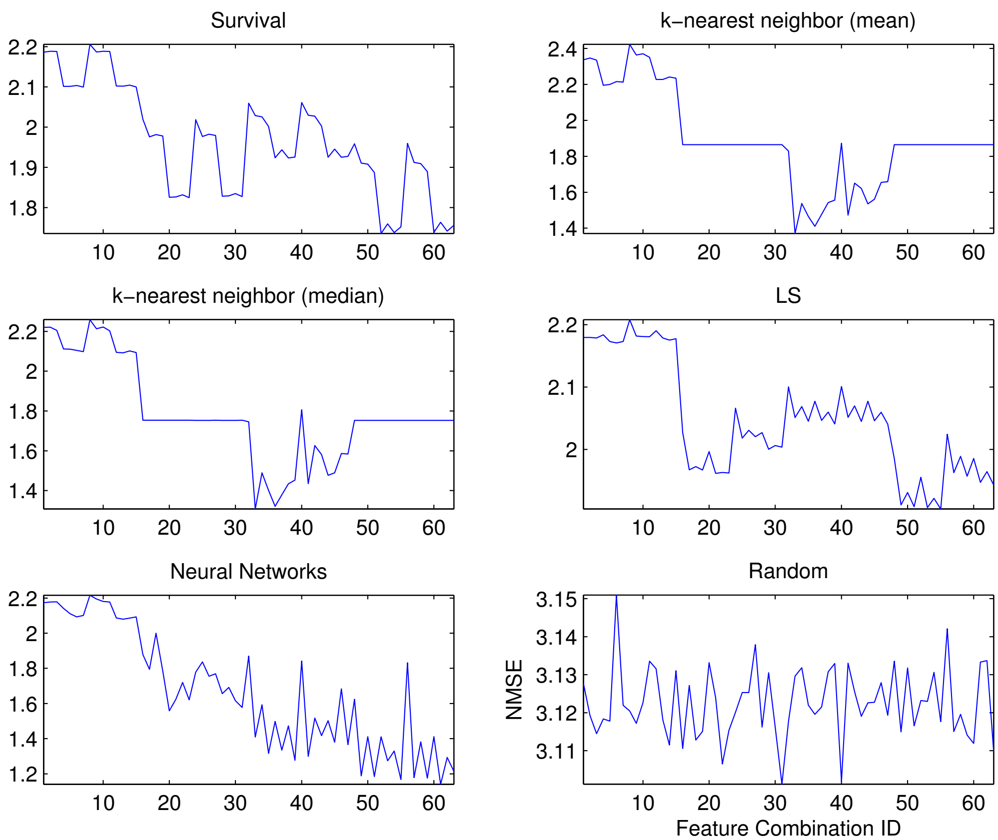


Fig. 3.6: Median NMSE.

and try to select the best feature set (out of $2^N - 1$) for online prediction. We select the neural networks approach for this evaluation. The objective of feature selection is to find the features that give the best prediction across all performance metrics defined in Section 3.5.2.

Table 3.1 shows the subset of six features ranked according to each of the performance metrics: R^2, NMSE, RG(2) and RL(0.5). For example, the features corresponding to the binary number 61, i.e, NRB, IBT, STT, BR and BUR, give the best performance according to the R^2, NMSE and RG(2), whereas the features correspond-

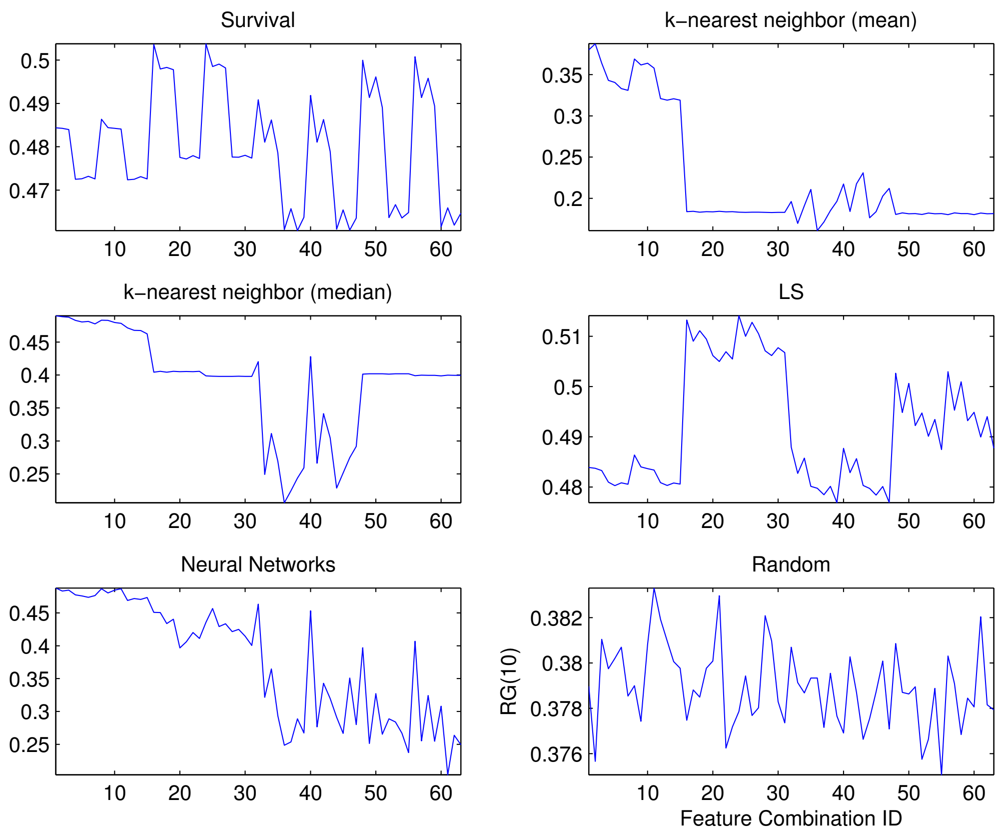


Fig. 3.7: Median RG(r).

ing to the binary number 41, i.e., i.e, NRB, STT, and BUR, give the best performance according to RL(0.5).

We employ a method known as *Borda count* [10]. in order to select the feature sets based on all four evaluation metrics. For each feature ID (binary number) in Table 3.1, the Borda count gives a point based on the ranking of that ID in each of the four evaluation metrics. Then, the feature ID having the most Borda points is selected as the best feature set in terms of all four evaluation metrics. Table 3.2 summarizes the Borda count procedure in selecting the best feature set. For this particular example, the

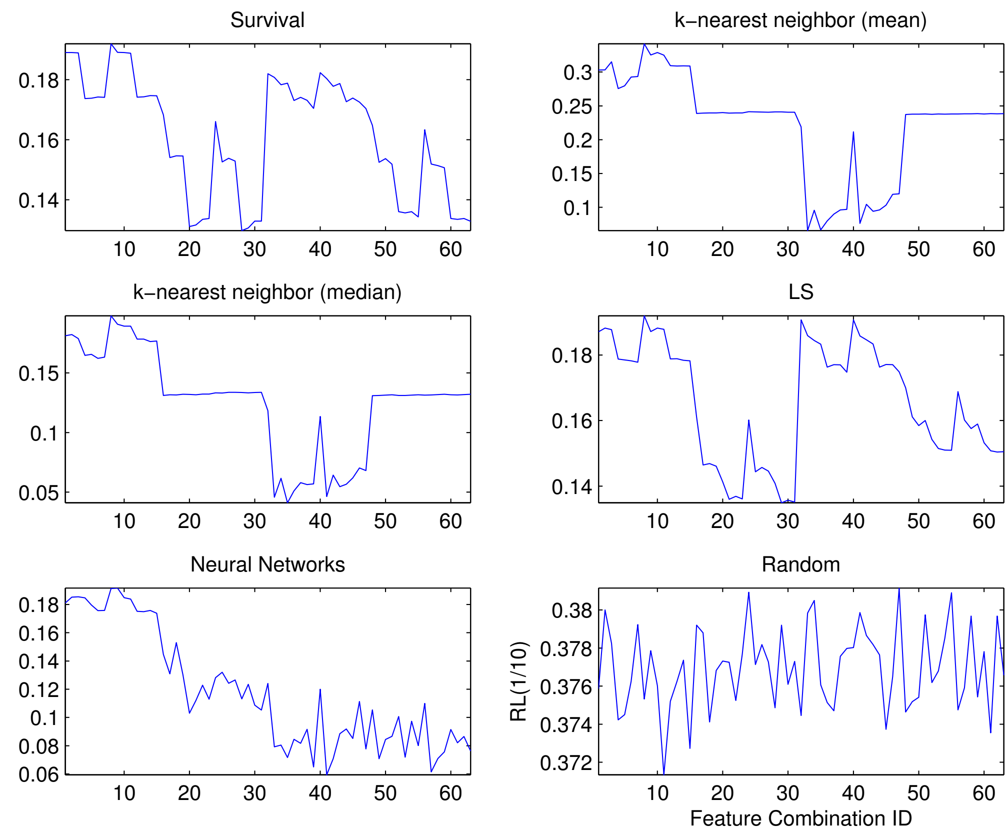


Fig. 3.8: Median RL($1/r$).

feature set ID 61 is ranked first. Hence, for the neural network approach, the features NRB, IBT, STT, BR and BUR will be used for online playtime prediction. The features are selected in a similar fashion for the rest of the five playtime prediction methods.

3.5.4 Playtime Prediction Results

Assuming that the best features are selected offline based on the approach described in the previous section, in this section, we show the accuracy of online playtime prediction results of each approach.

Table 3.1: Performance Metrics

Feature ID	NRB	NRP	IBT	STT	BR	BUR	R2	NMSE	RG(2)	RL(0.5)
61	x		x	x	x	x	0.74479	1.1028	0.19743	0.074251
55	x	x	x		x	x	0.73473	1.1427	0.22604	0.073569
63	x	x	x	x	x	x	0.71984	1.1738	0.24353	0.068233
57	x			x	x	x	0.71637	1.1788	0.25386	0.059378
59	x	x		x	x	x	0.7162	1.1766	0.25079	0.065054
49	x				x	x	0.71531	1.1683	0.2584	0.066417
61	x		x	x	x	x	0.74479	1.1028	0.19743	0.074251
55	x	x	x		x	x	0.73473	1.1427	0.22604	0.073569
49	x				x	x	0.71531	1.1683	0.2584	0.066417
63	x	x	x	x	x	x	0.71984	1.1738	0.24353	0.068233
59	x	x		x	x	x	0.7162	1.1766	0.25079	0.065054
57	x			x	x	x	0.71637	1.1788	0.25386	0.059378
61	x		x	x	x	x	0.74479	1.1028	0.19743	0.074251
37	x		x			x	0.69295	1.2289	0.2072	0.070391
36			x			x	0.54402	1.4849	0.21946	0.067666
55	x	x	x		x	x	0.73473	1.1427	0.22604	0.073569
45	x		x	x		x	0.65841	1.2883	0.23399	0.073683
63	x	x	x	x	x	x	0.71984	1.1738	0.24353	0.068233
41	x			x		x	0.65943	1.2865	0.28213	0.043824
53	x		x		x	x	0.71386	1.1796	0.271	0.057788
57	x			x	x	x	0.71637	1.1788	0.25386	0.059378
39	x	x	x			x	0.67346	1.2644	0.25318	0.060513
59	x	x		x	x	x	0.7162	1.1766	0.25079	0.065054
33	x					x	0.65471	1.2997	0.31903	0.065622
	23	10	14	14	18	24				

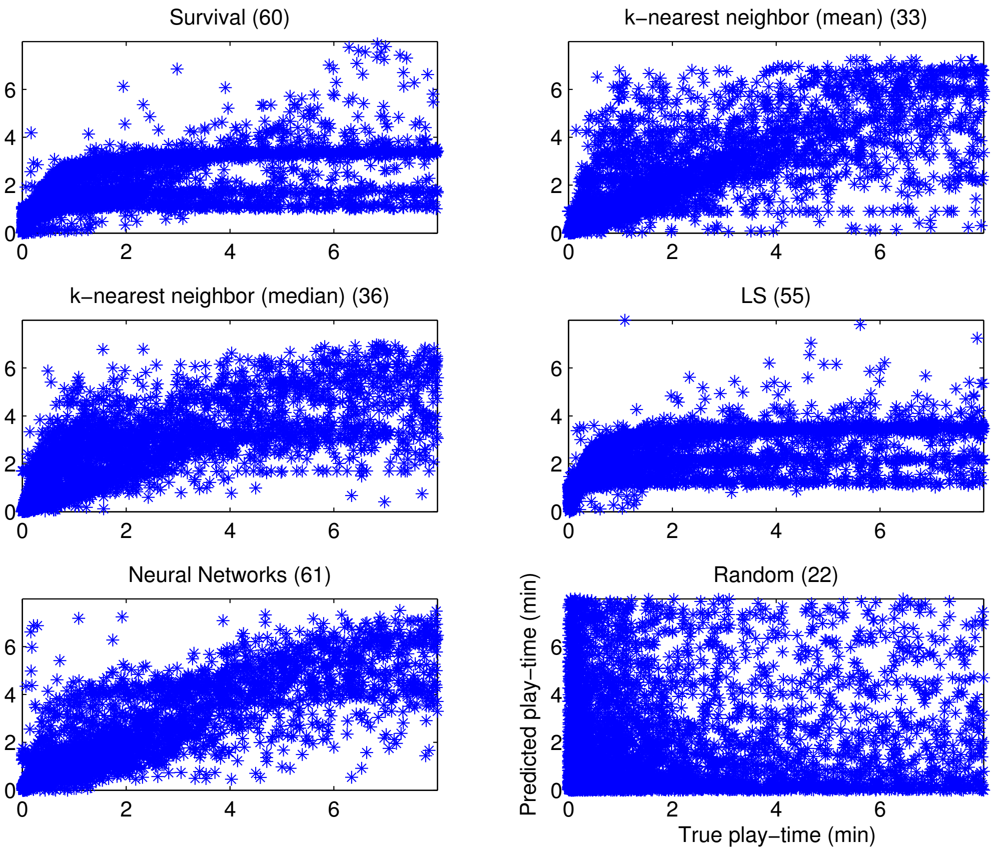


Fig. 3.9: Scatter plot of true vs. predicted playtime.

Figure 3.9 shows a scatter plot of true vs. predicted play time. Each subplot corresponds to the prediction approach mentioned in the title. For each approach, the feature ID corresponding to the top Borda count is displayed in parenthesis as well. Ideally, the scatter plot has to look like a line from the origin with a slope of 1; the "thickness" of the scatter as well as "concentrations" at off-diagonal places indicate the error in predictions.

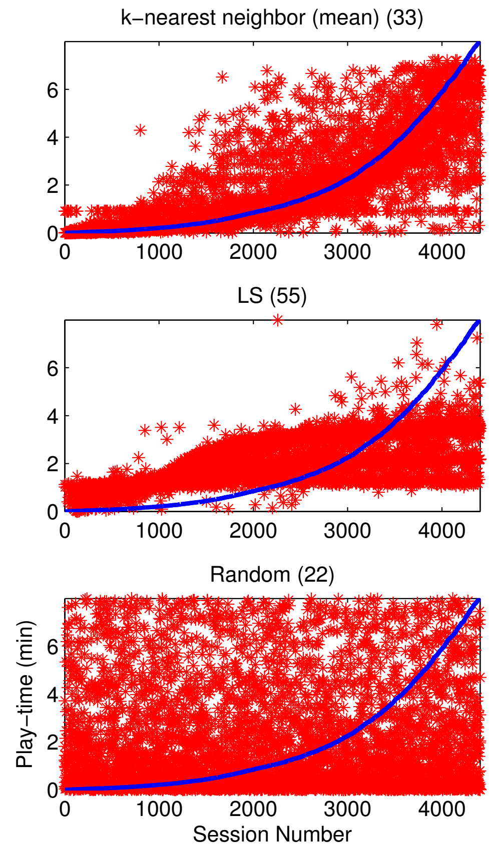


Fig. 3.10: Overlay plot of true vs. predicted playtime.

Figure 3.10 shows the predicted play time as an overlay plot of true and estimates.

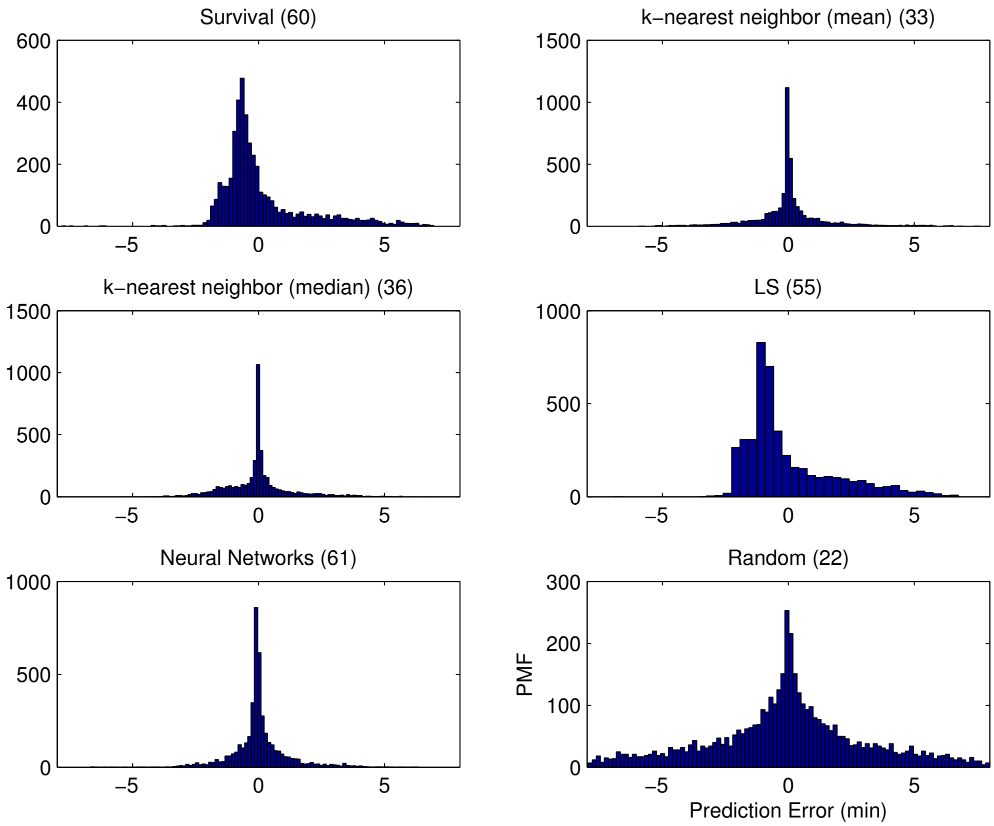


Fig. 3.11: PMF of playtime-prediction error.

Figure 3.11 shows the prediction errors as a histogram.

Table 3.2: Feature Ranking Based on Borda Count

Feature ID	Rank: R2	Rank: NMSE	Rank: RG(2)	Rank: RL(0.5)	Borda Count	Borda Rank
61	1	1	1	13	40	1
55	2	2	4	11	37	2
57	4	6	9	3	34	3
59	5	5	7	5	34	3
63	3	4	6	9	34	3
49	6	3	10	7	30	4
53	7	7	11	2	29	5
37	8	8	2	10	28	6
39	9	9	8	4	26	7
41	10	10	12	1	23	8
36	13	13	3	8	19	9
45	11	11	5	12	17	10
33	12	12	13	6	13	11

3.6 Conclusions

In this chapter, we proposed an approach to use the predicted play time as a measure of the quality of experience (QoE) of the user in video-on-demand (VoD) services. Online predicted playtime allows the content delivery service provider to allocate video server resources efficiently and to provide user-specific video recommendations, both of which contributes towards enhanced QoE of the viewer. We demonstrate play time prediction using three novel approaches on Comcast's video on demand service - Xfinity. We developed an approach that helps one to detect significant features that contributed to the predicted playtime and to prioritize the response by the content delivery service provider.

Chapter 4

Conclusions and Directions for Future Research

4.1 Conclusions

In this thesis, a novel *Cognitive Video Streaming (CVS)* architecture has been designed, in an attempt to provide interactive feedback system for improving the quality of experience (QoE) of Video on Demand (VOD) users. Our CVS architecture consists of three primary modules, **estimation module**, **prediction module** and **adaptation module**. Each module is made up of several subroutines; the proper functionality of each of these subroutine will improve the overall performance of the CVS. Prediction of the *play time ratio (PTR)* is an important subroutine in the prediction module. We developed various machine learning algorithms such as linear regression, k-nearest neighbor, survival model and neural networks for the prediction of PTR.

4.2 Future Research

Several other modules and subroutines of the proposed CVS architecture remain to be developed. Below, we briefly list some of the important future works.

- Enhancing the usability of the Cognitive Video Streaming (CVS) architecture by testing it on real time data from different Comcast customers.

- Developing algorithms for other subroutines of the prediction module, such as the probability of return (POR) prediction, probability of re-play (POP) prediction, and average length of scrubbing (LOS) prediction.

- Completing the subroutines of the estimation module, i.e., computing the remaining factors such as, content factors, viewer factors and seasonal factors in order to improve the functionalities of the prediction module.

- Implementing anomaly/threat detection capabilities in order to achieve a resilient CVS system.

- Developing a performance measuring schema with which the effectiveness of the CVS architecture can be objectively measured and further improved.

Bibliography

[1] "Cisco visual networking index: Forforecast methodology 2013-2018." Cisco, Tech. Rep., June 2014.

[2] *Transparent End-to-End Packet Switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming Over HTTP (3GP-DASH)*, 3GPP Std. TS 26.247 v10.1.0, June 2011.

[3] V. Aggarwal, E. Halepovic, J. Pang, S. Venkataraman, and H. Yan, "Prometheus: Toward quality-of-experience estimation for mobile apps from passive network measurements," *ACM HotMobile*, 2014.

[4] A. Anand, A. Balachandran, A. Akella, V. Sekar, and S. Seshan, "Enhancing video accessibility and availability using information-bound references," in *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*. ACM, 2013, pp. 345–356.

[5] A. Balachandran, V. Sekar, A. Akella, and S. Seshan, "Analyzing the potential benefits of CDN augmentation strategies for internet video workloads," in *Proceedings of the 2013 conference on Internet measurement conference*. ACM, 2013, pp. 43–56.

[6] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang, "A quest for an internet video quality-of-experience metric," in *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*. ACM, 2012, pp. 97–102.

[7] ——, "Developing a predictive model of quality of experience for internet video," in *Proceedings of the ACM conference on SIGCOMM*. ACM, 2013, pp. 339–350.

[8] B. Balasingam, M. Baum, D. Pasupuleti, P. Mannaru, K. R. Pattipati, P. Willett, C. Lintz, G. Commeau, F. Dorigo, and J. Fahrny, "Online playtime prediction in video streaming for better QoE," *IEEE/ACM Transactions on Networking*, 2014, (submitted, 2014).

[9] B. Balasingam, S. Sankavaram, K. Choi, D. M. Ayala, D. Sidoti, K. R. Pattipati, P. Willett, C. Lintz, G. Commeau, F. Dorigo, and J. Fahrny, "Online anomaly detection in big data," in *International Conference on Information Fusion*, 2014.

[10] D. Black, "Partial justification of the borda count," *Public Choice*, vol. 28, no. 1, pp. 1–15, 1976. [Online]. Available: http://dx.doi.org/10.1007/BF01718454

[11] P. Brooks and B. Hestnes, "User measures of quality of experience: why being objective and quantitative is important," *IEEE Network*, vol. 24, no. 2, pp. 8–13, March 2010.

[12] J. Chen, A. Ghosh, J. Magutt, and M. Chiang, "QAVA: quota aware video adaptation," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*. ACM, 2012, pp. 121–132.

[13] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei, "Quantifying Skype user satisfaction," in *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4. ACM, 2006, pp. 399–410.

[14] D. R. Cox and D. Oakes, *Analysis of survival data*. CRC Press, 1984, vol. 21.

[15] N. Damera-Venkata, T. D. Kite, W. S. Geisler, B. L. Evans, and A. C. Bovik, "Image quality assessment based on a degradation model," *IEEE Transactions on Image Processing*, vol. 9, no. 4, pp. 636–650, 2000.

[16] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. A. Joseph, A. Ganjam, J. Zhan, and H. Zhang, "Understanding the impact of video quality on user engagement," *SIGCOMM-Computer Communication Review*, vol. 41, no. 4, p. 362, 2011.

[17] X. Han, S. Mandal, K. Pattipati, D. Kleinman, and M. Mishra, "An optimization-based distributed planning algorithm: A blackboard-based collaborative framework," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 6, pp. 673–686, June 2014.

[18] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[19] T.-Y. Huang, R. Johari, and N. McKeown, "Downton abbey without the hiccups: Buffer-based rate adaptation for http video streaming," in *Proceedings of the 2013 ACM SIGCOMM workshop on Future human-centric multimedia networking*. ACM, 2013, pp. 9–14.

[20] *Objective perceptual video quality measurement techniques for standard definition digital broadcast television in the presence of a full reference*, International Telecommunication Union Std., ITU-R Recommendation BT.1683, 2004.

[21] *Methodology for the subjective assessment of the quality of television pictures*, International Telecommunication Union Std., ITU-R Recommendation BT.500-11, 2002.

[22] *Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference*, International Telecommunication Union Std., ITU-T Recommendation J.144, 2004.

[23] *Perceptual visual quality measurement techniques for multimedia services over digital cable television networks in the presence of a reduced bandwidth reference*, International Telecommunication Union Std., ITU-T Recommendation J.246, 2008.

[24] *Objective perceptual multimedia video quality measurement in the presence of a full reference*, International Telecommunication Union Std., ITU-T Recommendation J.247, 2008.

[25] *Subjective video quality assessment methods for multimedia applications*, International Telecommunication Union Std., ITU-T Recommendation P.910, 2008.

[26] *Information Technology – Dynamic Adaptive Streaming Over HTTP (DASH) – Part 1: Media Presentation Description and Segment Formats*, ISO Std. IEC DIS 23 009-1, Aug 2011.

[27] S. S. Krishnan and R. K. Sitaraman, "Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs," in *Proceedings of the 2012 ACM conference on Internet measurement conference*. ACM, 2012, pp. 211–224.

[28] S. Krishnan and R. Sitaraman, "Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs," *IEEE/ACM Transactions on Networking*, vol. 21, no. 6, pp. 2001–2014, Dec 2013.

[29] W. C. Levy, D. Mozaffarian, D. T. Linker, S. C. Sutradhar, S. D. Anker, A. B. Cropp, I. Anand, A. Maggioni, P. Burton, M. D. Sullivan *et al.*, "The seattle heart failure model prediction of survival in heart failure," *Circulation*, vol. 113, no. 11, pp. 1424–1433, 2006.

[30] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. 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, pp. 719–733, April 2014.

[31] C. Liu, R. K. Sitaraman, and D. Towsley, "Go-with-the-winner: Client-side server selection for content delivery," *arXiv preprint arXiv:1401.0209*, 2013.

[32] R. Lynch and P. Willett, "Bayesian classification and feature reduction using uniform Dirichlet priors," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 33, no. 3, pp. 448–464, June 2003.

[33] S. Marano, V. Matta, and P. Willett, "Nearest-neighbor distributed learning by ordered transmissions," *IEEE Transactions on Signal Processing*, vol. 61, no. 21, pp. 5217–5230, Nov 2013.

[34] R. K. Mok, E. W. Chan, and R. K. Chang, "Measuring the quality of experience of HTTP video streaming," in *IFIP/IEEE International Symposium on Integrated Network Management*. IEEE, 2011, pp. 485–492.

[35] C. Neukirchen, J. Rottland, D. Willett, and G. Rigoll, "A continuous density interpretation of discrete HMM systems and MMI-neural networks," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 4, pp. 367–377, May 2001.

[36] O. Oyman and S. Singh, "Quality of experience for HTTP adaptive streaming services," *IEEE Communications Magazine*, vol. 50, no. 4, pp. 20–27, 2012.

[37] M. H. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *IEEE Transactions on Broadcasting*, vol. 50, no. 3, pp. 312–322, 2004.

[38] F. Qiu and Y. Cui, "A quantitative study of user satisfaction in online video streaming," in *Consumer Communications and Networking Conference*. IEEE, 2011, pp. 410–414.

[39] J. Shaikh, M. Fiedler, T. Minhas, P. Arlos, and D. Collange, "Passive methods for the assessment of user-perceived quality of delivery," in *SNCNW*, 2011, p. 73.

[40] S.-H. Shen and A. Akella, "An information-aware QoE-centric mobile video cache," in *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM, 2013, pp. 401–412.

[41] C. J. Van den Branden Lambrecht and O. Verscheure, "Perceptual quality measure using a spatiotemporal model of the human visual system," in *Electronic Imaging: Science & Technology*. International Society for Optics and Photonics, 1996, pp. 450–461.

[42] M. Venkataraman and M. Chatterjee, "Quantifying video-QoE degradations of internet links," *IEEE/ACM Transactions on Networking*, vol. 20, no. 2, pp. 396–407, April 2012.

[43] Z. Wang and A. C. Bovik, "Mean squared error: love it or leave it? a new look at signal fidelity measures," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, 2009.

[44] Z. Wang, L. Lu, and A. C. Bovik, "Video quality assessment based on structural distortion measurement," *Signal processing: Image communication*, vol. 19, no. 2, pp. 121–132, 2004.