18th International Conference on Information Fusion
Washington, DC - July 6-9, 2015

Online Playtime Prediction for Cognitive Video Streaming

D. Pasupuleti*, P. Mannaru*, B. Balasingam*, M. Baum*, K. Pattipati*, P. Willett*,
C. Lintz†, G. Commeau†, F. Dorigo†, and J. Fahrny†
*University of Connecticut, CT, USA
Email: {devaki, pujitha, bala, mbaum, krishna, willett}@engr.uconn.edu
†Comcast Corporation
Email: {christopher_lintz, gabriel_commeau, francesco_dorigo, jim_fahrny}@cable.comcast.com

***Abstract*—In this paper, we consider the problem of cognitive video streaming in video on demand (VoD) services. The focus lies on quantities that are indicative of the quality of experience (QoE) of the subscriber, such as playtime ratio, probability of return, probability of replay and startup time. Especially, in this paper, we develop and evaluate a playtime prediction tool. For this purpose, the applicability of different machine learning algorithms such as k-nearest neighbor, neural network regression, and survival models is investigated; then, we develop an approach to identify the most relevant factors that contributed to the prediction. The proposed approaches are tested by means of a data set provided by Comcast.**

Keywords*—Video quality of service (QoS), quality of experience (QoE), internet video, human factors, mean opinion score (MOS), video quality metrics, machine learning, neural networks, nearest neighbor classification, survival models.

I. Introduction

With individualized programming, there is tremendous strain on the video streaming infrastructure. However, there is also a tremendous opportunity to use it effectively and profitably: how should precious bandwidth be adaptively apportioned to each user? This would seem exactly the issue that the well-studied cognitive network faces, viz., to use feedback (of load, forecast, spectrum, other users' demands) for optimal allocation of resources. However, there is a fundamental difference: the information and entertainment media consumer defines acceptable quality of experience (QoE) in a subjective way. First and foremost, a viewer would like a video to start without delay, continue to play without any buffering events and have very good picture quality all the while; for a content provider, this would mean that all the quality of service (QoS) features of the streaming video are well maintained. Good quality streaming alone does not satisfy a viewer; a viewer expects the contents of the video to be likable, and suitable for the time of viewing – this would mean that the entire (video) resource has to be made readily available to the viewer's personal preferences. Considering that the viewers, numbered in tens of millions, are geographically scattered across the nation, it is logistically challenging to the content provider to facilitate the availability of *any video, anytime, anywhere* with guaranteed QoS for all the viewers.

Many approaches for adaptive video streaming have been published in the literature [21], [1], [12], [16], [13], [5], [9]. Most of the adaptive streaming strategies suggest to adapt the bitrate uisng metrics derived based on buffering events [15], [10]. Other than adaptive streaming, there are several works in the literature suggesting various approaches to enhance a specific aspect of QoE; in [2], an approach is suggested to enhance the accessibility in shared video forums; [3] suggests to exploit 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 [18]; in [20] the QoE is modeled based on a packet loss model; in [22], [23], the QoE is modeled in terms of the QoS factors such as loss, delay and jitter; and [6] talks about providing good quality video while being aware of the bandwidth quota of the user.

We envision a *cognitive video streaming strategy* [19] that is able to keep the entire swath of customers at maximal satisfaction by QoS standards; this is achieved by cognitive learning of user expectations, network traffic prediction, timely identification of anomalous behaviors and potential threats, content management and load forecasting by accounting for the effects due to content, viewer, QoS, season and other external factors.

It is observed that most of the opened videos are not completely watched by the viewers at once. The nature of completion of a particular video changes from viewer to viewer; some videos get 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 "temporarily" abandoned to be continued later. Once a viewer starts playing a video, knowing the *remaining playtime* of that video is a useful information to the content provider in order to ensure adequate QoE to the viewer; the knowledge of the remaining playtime can be used to allocate server bandwidth to the user; it can be used to device a more appropriate adaptive bitrate switching scheme; and the prior knowledge of a video getting abandoned by the viewer can be used to recommend better videos at the first place. In the network level, the predicted playtime of each viewing session is a useful information for predicting and managing network traffic.

The rest of the paper is organized as follows: The proposed playtime prediction tool is introduced in Section II. Section III describes the data sets used in the experimentations of this paper. Three approaches for playtime prediction are summarized in Section IV. Numerical results based on collected data is presented in Section V and the paper is concluded in Section VI.

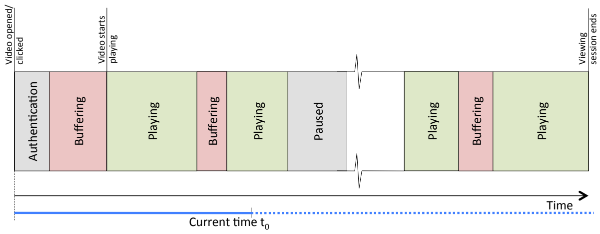

Fig. 1: **Typical video viewing session.** The purpose of the playtime prediction tool (PPT) is to estimate the remaining playtime at the current point in time t_0.

II. PLAYTIME PREDICTION TOOL (PPT)

Figure 1 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 process 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.

In this work, we aim at developing an online *playtime prediction tool (PPT)* that estimates a prediction of the remaining playtime in a viewing session, see Figure 1. Technically, the tool may run on either the client side or the server side. The most similar work [11] aims at developing methods for predicting the playtime of completed sessions.

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 1. 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 III.

We propose to exploit previously logged protocol data in order to "learn" a playtime predictor using machine learning algorithms. In general, a playtime predictor can be "set up" for specific users, particular VoD assets, or a group of users.

III. 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 at 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 playtime until t_0:* PLT(t_0)
- *Total pause time until t_0:* PAT(t_0)
- *Total buffering time until t_0:* BUT(t_0)
- *Average bitrate until t_0:* BR(t_0)
- *Number of paused events until t_0:* NRP(t_0)
- *Number of buffering events until t_0:* NRB(t_0)
- *Startup time:* STT
- *Average Frame Rate:* 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 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 for predicting the playtime.

IV. METHODS FOR PLAYTIME 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 paper.

A. Linear Least Squares Prediction

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

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

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 $\mathbf{k} = [k_1, k_2, \ldots, k_{N_x}]$ can be estimated by collecting the observation pairs $\{y_i, \mathbf{x}_i\}$ where $\mathbf{x}_i = [x_{i,1}, x_{i,1}, \ldots, x_{i,N_x}]^T$ for $i = 1, \ldots, M$, i.e.,

$$\hat{\mathbf{k}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y} \tag{2}$$

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

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

$$\hat{y}_j = \mathbf{x}_j^T \hat{\mathbf{k}} \tag{3}$$

B. 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

$$d_{i,j} = \mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) \tag{4}$$

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 playtime 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.

C. Survival Models

Survival modeling has found wide applications in a number of areas, including medicine [8] and equipment failure analysis [17]. Survival modeling was employed to derive a QoE metric in [7]. 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

$$P_\xi(t) \triangleq f(t) \tag{5}$$

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

The cumulative probability distribution function of ξ

$$F(t) \quad = \quad P(\xi \leq t) = \int_0^t f(u)du \tag{6}$$

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

$$R(t) = P(\xi > t) = 1 - F(t) \tag{7}$$

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 next moment, i.e.,

$$h(t) = f(t|\xi > t) = \frac{f(t)}{R(t)} \tag{8}$$

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 regardless of the shape of $f(t)$. Let us rewrite (8) as

$$h(t) = \frac{f(t)}{1 - F(t)} = \frac{F'(t)}{1 - F(t)} = -\frac{R'(t)}{R(t)} \tag{9}$$

Integrating both sides of (9)

$$-\int_0^t h(u)du = \ln R(t) \tag{10}$$

Hence,

$$R(t) = \exp\Big\{ - H(t)\Big\} \tag{11}$$

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

Using (7) and (11)

$$\begin{aligned} 1 - F(t) &= \exp\Big\{ - H(t)\Big\} \\ f(t) &= h(t)\exp\Big\{ - H(t)\Big\} \end{aligned} \tag{12}$$

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 [8], suggests to separate the time-dependent and feature-dependent hazards as follows:

$$h(t, \mathbf{x}) = \lambda(t)\exp\left\{\mathbf{b}^T\mathbf{x}\right\} \tag{13}$$

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, (11) and (12) are rewritten as

$$f(t) \quad = \quad \lambda(t)\exp\Big\{\mathbf{b}^T\mathbf{x} - \Lambda(t)e^{\mathbf{b}^T\mathbf{x}}\Big\} \tag{14}$$

$$R(t) \quad = \quad \exp\Big\{ - \Lambda(t)e^{\mathbf{b}^T\mathbf{x}}\Big\} \tag{15}$$

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 playtime can be computed as

$$\hat{y}_j = \frac{\int_u^\infty (t - u) f_j(t)dt}{R_j(u)} \tag{16}$$

where $f_j(t)$ and $R_j(u)$ are obtained by substituting $\mathbf{x}_j$ for $\mathbf{x}$ in (14) and (15), respectively, and u is the time elapsed.

An advantage of the survival model-based approaches described above is that the playtime prediction can be updated as the video progresses. In this paper, we assume $\lambda(t) = \lambda$.

D. Neural Networks

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

$$y_i = f(\mathbf{x}_i, \{w_{l,k}\}_{l=1,k=1}^{N_L,N_h}) \tag{17}$$

where $w_{l,k}$ 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 [14]. A trained neural network can be used to predict the playtime for a given feature set $\mathbf{x}_j$.

V. Simulation Studies

In this section, we evaluate the proposed approaches using data from 8808 viewing sessions. We focus on the first 8 minutes as we try to detect early quitters due to the low streaming quality. All the viewing sessions occurred on the same day. A portion of this data is randomly selected and denoted as the "learning" 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). 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.

A. *Performance Metrics*

In this section, we use the algorithms introduced in Section IV 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.

1) Normalized Mean-Squared Error (NMSE): This metric gives an insight about the error in playtime prediction of

$$\mathrm{NMSE} = \frac{1}{M}\sum_{i=1}^{M}\left(\frac{y_i - \hat{y}_i}{y_i}\right)^2 \tag{18}$$

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

$$R^2 = 1 - \frac{\sum_{i=1}^{M}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{M}(y_i - \bar{y})^2} \tag{19}$$

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

3) 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 hours. A prediction error of 1 min is significant if the actual playtime is 5 min; however, it is not so significant if the actual playtime is 2 hours. The NMSE captures this through normalization; however, the following metric captures this error in a different light.

$$\mathrm{RG}(r) \quad = \quad \frac{\#\left\{\frac{\hat{y}_i}{y_i} > r\right\}}{M} \tag{20}$$

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

4) Ratio of Predicted and True Less than $1/r$: Similar to $\mathrm{RG}(r)$, the following metric capture the ratio of instances when the prediction was significantly smaller than the true value of playtime.

$$\mathrm{RL}(1/r) \quad = \quad \frac{\#\left\{\frac{\hat{y}_i}{y_i} < \frac{1}{r}\right\}}{M} \tag{21}$$

B. *Feature Selection*

With N features there are $2^N - 1$ possible selections from them. 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 Figure 3, we show the performances with respect to the NMSE plotted against a binary representation of feature combinations, 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 is plotted in Figure 3. There are six subplots showing the results of different playtime prediction approaches: Survival modeling, k-nearest neighbor (mean), k-nearest neighbor (median), LS, neural networks and random. In "random" approach, we randomly select a playtime from the training data set. The above analysis can also be performed with the other performance metrics discussed in Section V-A, i.e., median R^2, median $\mathrm{RG}(r)$, and median $\mathrm{RL}(1/r)$.

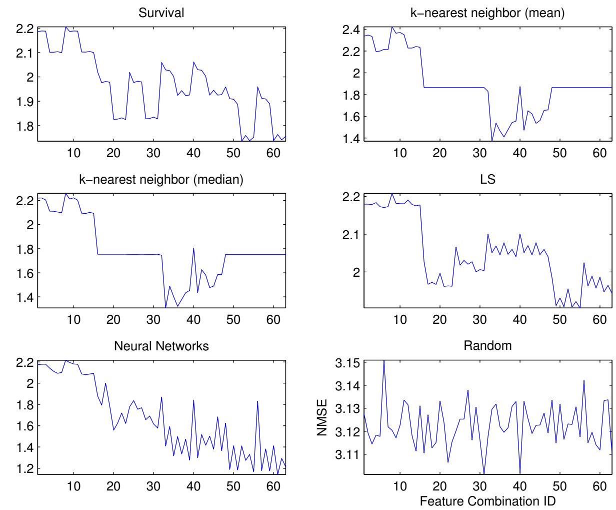

Fig. 3: **Median NMSE.**

Next, we focus on a single playtime prediction approach 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 gives the best result across all performance metrics defined in Section V-A.

Table I shows the first six feature sets 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 corresponding to the binary number 41, i.e., NRB, STT, and BUR, give the best performance according to RL(0.5).

We employ the *Borda count* [4] method in order to select the feature sets based on all four evaluation metrics. For each feature ID (binary number) in Table I, 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 II summarizes the Borda count procedure in selecting the best feature set. For this particular example, the feature ID 61 is ranked first. Hence, for neural network approach, the features NRB, IBT, STT, BR and BUR

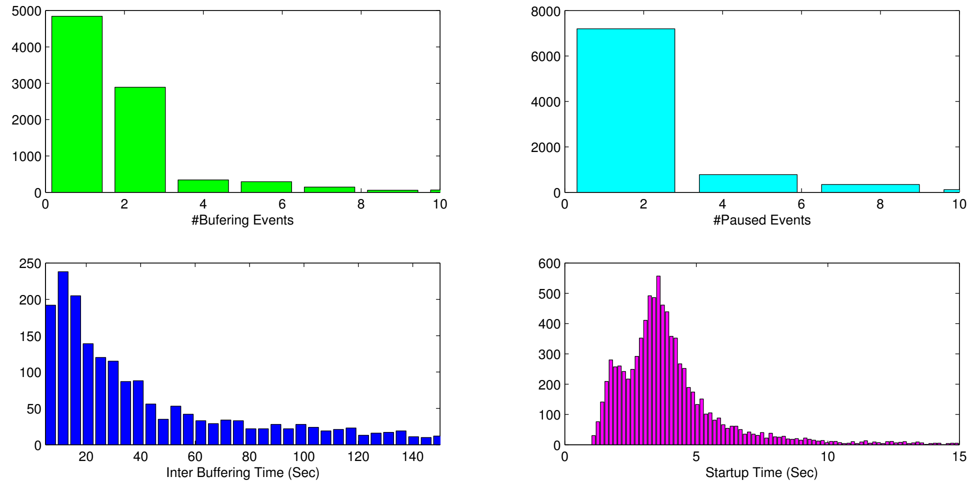

Fig. 2: **Histogram of features.**

TABLE I: Performance Metrics

Feature Combination ID	f1	f2	f3	f5	f6	f7	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
Total Features	23	10	14	14	18	24				

will be used for online playtime prediction. The features are selected in a similar fashion for the rest of the five playtime prediction methods.

C. 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 online playtime prediction results of each approach. Figure 4 shows a scatter plot of true vs. predicted playtime. 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 gradient 1; the "thickness" of the scatter as well as "concentrations" at off-diagonal places indicate the error in predictions. Figure 5 shows the prediction errors as a histogram; a "thin" histogram implies good prediction.

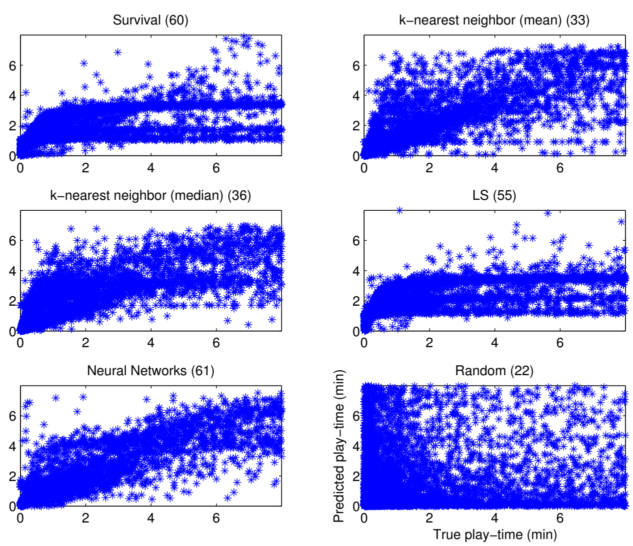

Fig. 4: **Scatter plot of true vs. predicted playtime.**

VI. Conclusions

In this paper, we argue that the predicted playtime is a strong indicator of the quality of experience (QoE) 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 playtime prediction using three novel approaches on Comcast's video on demand service - Xfinity. Furthermore, we developed an approach to detect significant features that contributed to the predicted playtime and to prioritize the response by the content provider.

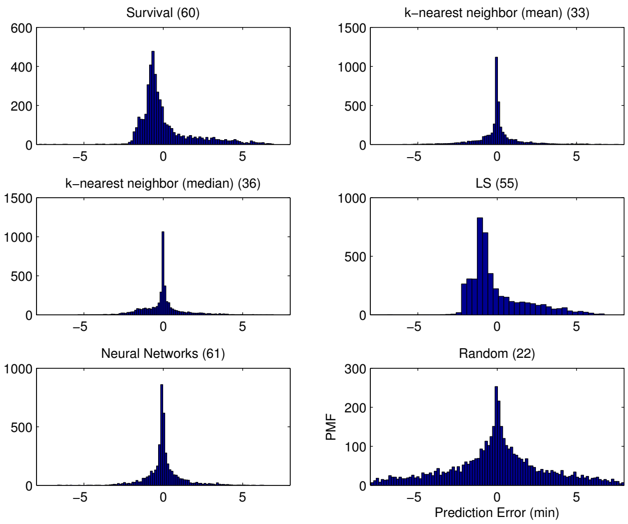

Fig. 5: **PMF of playtime-prediction error.**

TABLE II: 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

References

[1] S. Akhshabi, L. Anantakrishnan, C. Dovrolis, and A. C. Begen, "Server-based traffic shaping for stabilizing oscillating adaptive streaming players," in *Proceeding of the 23rd ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*. ACM, 2013, pp. 19–24.

[2] 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.

[3] 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.

[4] D. Black, "Partial justification of the borda count," *Public Choice*, vol. 28, no. 1, pp. 1–15, 1976.

[5] T.-Y. H. N. H. Brandon and H. N. M. R. Johari, "Confused, timid, and unstable: Picking a video streaming rate is hard," 2012.

[6] 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.

[7] 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.

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

[9] L. De Cicco and S. Mascolo, "An adaptive video streaming control system: Modeling, validation, and performance evaluation."

[10] T. De Pessemier, K. De Moor, W. Joseph, L. De Marez, and L. Martens, "Quantifying the influence of rebuffering interruptions on the user's quality of experience during mobile video watching," *IEEE Transactions on Broadcasting*, vol. 59, no. 1, pp. 47–61, March 2013.

[11] 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.

[12] B. Frank, I. Poese, G. Smaragdakis, S. Uhlig, and A. Feldmann, "Content-aware traffic engineering," vol. 40, no. 1. ACM, 2012.

[13] V. Gabale, P. Dutta, R. Kokku, and S. Kalyanaraman, "Insite: QoE-aware video delivery from cloud data centers," in *IEEE 20th International Workshop on Quality of Service*. IEEE, 2012, pp. 1–9.

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

[15] 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.

[16] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*. ACM, 2012, pp. 97–108.

[17] 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.

[18] 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.

[19] D. Pasupuleti, P. Mannaru, B. Balasingam, M. Baum, K. Pattipati, and P. Willett, "Cognitive video streaming," in *International Conference on EEECOS*, 2015.

[20] 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.

[21] 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.

[22] 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.

[23] ——, "Effects of internet path selection on video-QoE: Analysis and improvements," *IEEE/ACM Transactions on Networking*, vol. 22, no. 3, pp. 689–702, June 2014.