Take-away TV: Recharging Work Commutes with Predictive Preloading of Catch-up TV Content

Dmytro Karamshuk[1¶], Nishanth Sastry[1¶◇], Mustafa al-Bassam[1†], Andrew Secker[2], Jigna Chandaria[2]
[1]King's College London, Strand, London, WC2R 2LS [2]BBC R&D, 56 Wood Lane, London, W12 7SB
[1]{firstname.lastname}@kcl.ac.uk [2]{firstname.lastname}@rd.bbc.co.uk

✦

Abstract—Mobile data offloading can greatly decrease the load on and usage of current and future cellular data networks by exploiting opportunistic and frequent access to Wi-Fi connectivity. Unfortunately, Wi-Fi access from mobile devices can be difficult during typical work commutes, e.g., via trains or cars on highways. In this paper, we propose a new approach: to *preload* the mobile device with content that a user might be interested in, and thereby avoid the need for cellular data access. We demonstrate the feasibility of this approach by developing a supervised machine learning model that learns from user preferences for different types of content, and propensity to be guided by the UI of the player, and predictively preload entire TV shows. Testing on a dataset of nearly 3.9 million sessions from all over the UK to BBC TV shows, we find that predictive preloading can save over 71% of the mobile data for an average user.

Keywords—video streaming, predictive preloading, content delivery, mobile prefetching, supervised learning, catch-up TV

1 INTRODUCTION

Internet video services are increasingly going mobile. Conveniences offered by high bandwidth mobile networks and the availability of dedicated mobile video apps have raised the volume of per-user mobile video traffic by an incredible 262% in recent years [7]. Cisco predicts that mobile video will increase 14-fold between 2013 and 2018, accounting for 69 percent of total mobile data traffic by the end of the forecast period [6]. Mobile video and TV have reached an inflexion point: In October 2014, tablets overtook PCs in number of accesses to BBC iPlayer, an over-the-top TV streaming service used widely in the UK for accessing BBC TV shows [4].

This increase is accompanied by a behavioural shift among mobile users who now not only watch more, but are increasingly watching on the move, during their daily commutes [7, 17]. From the operators' perspective, as more and more users start watching videos during commutes, it is expected that there will be more capacity problems, a challenge that has not gone unnoticed by regulators [26]. From the users' perspective, the high bandwidth requirements of video content limit usage

because of data caps, which are very common in mobile plans [19].

A common solution to these problems lies in augmenting mobile networks with Wi-Fi connectivity[1]. Indeed, Cisco estimated that the common policy of preferentially using Wi-Fi where available, rather than cellular networks, allowed users to offload a remarkable 45 percent of total mobile data traffic in 2013 [6]. For cases when Wi-Fi may not be available at the time of request, researchers have also proposed *mobile data offloading* techniques, where users' requests for content are either processed with a delay [20, 3], or their mobility pattern is predicted and content pre-fetched to a Wi-Fi Access Point (AP) that the user may encounter in the near future [30, 29, 23, 28].

However, these "traditional" techniques for mobile offloading exploit opportunistic access to Wi-Fi connectivity, and are inadequate to support continous streaming of long-duration content such as TV shows during commutes: Wi-Fi APs appear in bursts, and are highly unlikely to be seen during typical commutes on highways, etc [3]. Further, measurement studies have shown that if the back up option of cellular connectivity is used, throughput diminishes when accessing from fast moving trains and cars [14]. Connectivity can also be patchy: in a test conducted in June 2014 on ten of the most popular commuting routes in to and out of London, 23.2% of 3G data packets and 37.2% of 4G data packets did not make it to their intended destinations [13]. The situation can be worse in metro trains which may go underground. A recent study of 48 metro systems from 28 countries suggests that the lack of good Internet connectivity underground is a common problem for many developed cities across the globe [31].

To address these difficulties in finding opportunistic Wi-Fi during commutes, we propose *predictive preloading*, a new approach to mobile data offloading for current and future generation 5G networks: In contrast to *predicting mobility patterns*, we propose to *predict the content* that a

---

1. Throughout this paper, we use Wi-Fi to denote access through a fixed-line broadband connection, potentially via a Wi-Fi access point. Similarly cellular networks is used to refer to data access over 3G, 4G or 5G networks.

user is likely to watch during the commute and *preload* that content on her mobile device in advance, when she might have access to reliable Wi-Fi connectivity, with sufficient spare bandwidth, e.g., at her home. The main challenge with preloading is that long-duration videos such as TV shows can take up a large amount of storage. Given the limited storage available on mobile devices, predictions for preloading need to be highly accurate in order to be useful. The amount of spare bandwidth available for preloading could also limit the amount of savings.

Our contributions are twofold: First, we analyze a trace of nearly 3.9M sessions from mobile devices accessing BBC's TV shows online during July 2014 to understand users' preferences in content types and propensity to be guided by the UI of the player. Second, we develop a predictive offloading mechanism for 3G/4G and future generation cellular networks which allows to save nearly 71% of mobile data for an average user.

From our data trace we find that users have their favourite channels and genres, which all capture a large proportion of their accesses, i.e., 75% users' accesses are made for content items from only 3 out of 11 categories and for 4 out of 11 channels. More remarkably, we notice a similar concentration of user preferences towards the top 4 or 5 content genres and shows despite a significantly higher degree of choice available (there are 172 genres and more than a thousand different shows available on iPlayer). We also find that a vast majority of users (around 75%) are influenced by the User Interface of the video player, and tend to access items which are featured by the BBC content editors on the iPlayer homepage, or access items on "most popular" lists (25% of accesses by average users and more than 80% of accesses by the top 10% of users are for such content items, which appear prominently on the user interface).

Based on these results, we develop a supervised learning model that predicts whether a user would watch a content item, and preload the most-likely-to-be-watched items at a scheduled time point on a daily basis, e.g., after midnight. We compare the results of the predictive preloading with a naïve baseline model which greedily preloads remaining parts of the last unfinished item when accessing over Wi-Fi/broadband or at a scheduled time. Our results suggest that predictive preloading allows to offload up to 71% of mobile data usage for an average user (over 95% for top 10% of users) and significantly outperforms naïve greedy techniques (which can only save $\approx$ 22% of per-user mobile data on average).

2 BACKGROUND

2.1 Related Work

The idea of augmenting cellular networks with opportunistic accesses to WiFi networks has attracted a lot of attention in the recent literature. Lee *et al.* [20] have shown that about 65% of outdoor *urban* mobile traffic can be saved by offloading cellular traffic to WiFi networks

without using any delayed transmission and extra 29% can be achieved by allowing long delays (over 1 hour) in delay-tolerant settings, whereas as little as 2-3% of savings can be achieved for short deadlines (i.e., less than 100 secs). Balasubramanian *et al.* [3] proposed an approach to the more difficult problem of offloading during *commutes* (rather than urban settings) by elaborating a simple method to predict future WiFi throughput and allowing delayed transmissions only if cellular savings are expected within an acceptable time window. Whereas this works well for delay tolerant applications, delay sensitive applications are still affected by the bursty and infrequent availability of Wi-Fi APs on highways and other commuter routes [3]. Although prefetching content to APs that are expected to be encountered in the near future can help to some extent, bursty availability of APs can still cause buffering and stalls for continuous streaming applications, and this is known to be deleterious for user engagement [8]. Other works in this direction have focused on *exploring predictability of human mobility* using a cross-layer implementation [30, 29, 23, 28]. In contrast to predicting mobility we focus on *predicting to-be-watched content* to assist pre-fetching decisions.

Other attempts at data offloading have started to use social context to prefetch content that a user might be predisposed to access for social reasons [11, 33, 15]. None of these attempt to completely offload content directly to the user's device, and it is unclear whether social information by itself can provide accurate-enough predictions when space is limited to less than 5–10 items. However, social and other context information, where available, can enhance preloading predictions made solely by using user preferences, as in our approach.

In contrast to using specific context, globally popular content can be prefetched for all users. Hoque *et al.* [12] focus on the most popular content and exploit crowd-sourced popularity statistics of other users to decide which chunks (of an item currently being watched) a user is more likely to watch next and speculatively pre-fetch more of the popular chunks from a WiFi network. The potential of utilising broadcast channels to pro-actively push bundles of the most popular content has been explored before [9], but only a mere 20% of mobile traffic savings has been reported. This is consistent with the savings we see for prefetching "Top" items (the baseline used in Fig. 4 (left)). By drawing on a much larger scope of signals specific for catch-up TV systems (i.e., user preferences, featured content lists, periodically released serial content, etc.) we obtain significantly larger savings, i.e., up to 71% of per-user mobile traffic.

It worth noting that in the context of BBC iPlayer itself, we have previously suggested a simpler prediction algorithm, where only local information about individual user preferences in TV programmes was exploited to predictively record broadcasted shows on set-top boxes [25]. While this simplified approach worked well for set-top boxes with significantly larger storage capacities (many current DVRs may have a 500 GB or 1 TB hard disk),

it proved to be inefficient for predictive preloading on mobile devices which have storage capacities to store only a dozen content items at maximum and, so, require significantly higher prediction accuracies. Similarly we have shown that peer-assisted approaches can deliver significant savings for accesses over fixed-line broadband [16]. However, support for P2P has not yet become prevalent in cellular networks, although device-to-device sharing has been proposed recently [10]. Because the approach in this paper caches directly on user devices, it has the advantage of not requiring additional spectrum for device to device (P2P) communications at the edge.

More broadly, predictive analytics in Internet TV systems have been primarily applied for channel zapping in IPTV [22, 1, 18], off-loading internet traffic for set-top box users [24] and programme recommendations [21, 34]. This paper elaborates on this line of work and extends it to mobile content offloading.

2.2 Motivation

A recent marketing research of mobile users conducted on a monthly trace of all mobile activities for 470 volunteers[2] suggests that 30% of smartphone users and 40% of table users watch videos on a daily basis with time of accesses peaking in the morning and afternoon hours. Indeed, in agreement with the above study, we previously found [17] that the majority of mobile accesses to BBC iPlayer happen during commute times, i.e., around 7-10AM in the morning and around 5-7PM in the afternoon (In contrast, accesses from broadband ISPs peak during evenings [17].).

The current work is motivated by two factors: On the one hand, the high bandwidth requirements of video content limit usage because of data caps, which are very common in mobile plans [19]. On the other hand, watching catch-up TV during work commutes in the UK is still an extremely challenging affair: neither train companies [13] nor London underground system [31] provide adequate Internet access (whether WiFi or cellular) suitable for streaming high resolution video content. Research studies in other environments have also found that high speed Internet (Wi-Fi or cellular) is difficult in typical commute trajectories [3, 14].

Although the general scarcity of Wi-Fi APs enroute makes the use of "traditional" mobile data offloading techniques difficult for delay-sensitive applications like continuous streaming of a TV show, it has been observed that most mobile devices have plenty of opportunities for high bandwidth Wi-Fi access [20]. Further, even though *real-time* streaming of TV content is delay sensitive, the content could potentially be *pre-staged* on the mobile device[3], removing the dependence on time.

Thus the core proposal of the current paper is to take advantage of high-bandwidth Wi-Fi connectivity when available, *to predict* what a user is going to watch next and download content *much before access* by the user. The total savings with predictive preloading is limited by the amount of bandwidth/storage available – perfect savings could be achieved by preloading the entire content catalogue, but this is clearly unrealistic. Therefore, to prioritise and make the best use of limited storage, we develop a machine learning model that takes user preferences, and current UI/featured content as signals, and predicts per-user likelihood of watching for each content item. The most-likely-to-be-watched items are then saved on the user's device.

3 UNDERSTANDING WATCHING PREFERENCES OF MOBILE CATCH-UP TV USERS

In this section we analyze watching preferences of catch-up TV users using a dataset of access logs from nearly 3.9 million sessions from mobile devices to BBC iPlayer – a widely used service for accessing BBC's TV and Radio shows over the Internet. Our goal is to find patterns in user accesses which can drive predictive preloading.

3.1 Dataset description

BBC iPlayer is an "over-the-top" video streaming service which provides free access to TV and radio content from a number of local and national BBC channels in the UK. Content items are typically published on iPlayer for "catch-up" viewing soon after broadcast and is made available for up to 30 days depending on licensing terms and other policies. iPlayer additionally provides live streaming access to content currently being broadcast, but on-demand access constitutes the vast majority ($\approx$90%) of TV sessions.

In this paper we consider a month-long snapshot of access logs for video content on BBC iPlayer in July, 2014. Each record in the dataset contains information about a user's session in the following format:

<*network type, user id, start time, duration, content id*>

The anonymized *user-id* is based on long-term cookies (with a four year expiration date), that uniquely identifies each device/user agent separately. *Network type* is obtained by resolving users' IP addresses to Autonomous System IDs using the RIPE[4] dataset and further manual classification of network ids into two classes: *mobile* (cellular connection) and *Wi-Fi* (Fixed-line Internet). A single user might have more than one *user-id* if they use more than one device, or even if they use more than one browser to access iPlayer. Users might also get multiple IDs if their cookies expire. However, in general, the two IDs (user and network) allow us to identify the different providers of each user.

Session duration shows the number of unique seconds of a content item that a user has watched during a

---

2. 2014 Mobile Behavior Report, http://goo.gl/uatYhj

3. Content can be protected using Digital Rights Management. Many applications including BBC iPlayer do this effectively already.
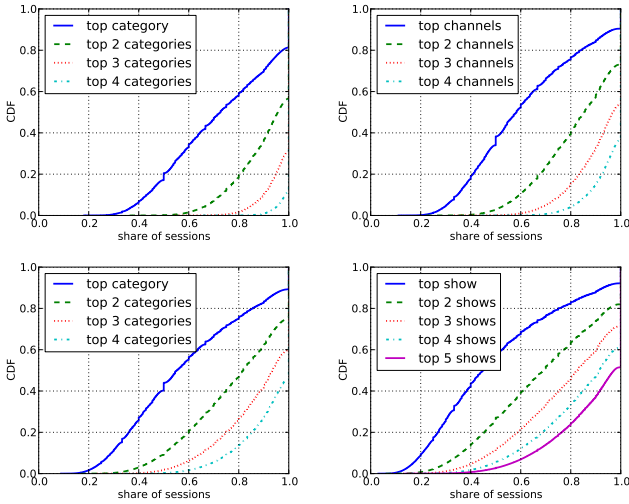
4. http://www.ripe.net/data-tools/db

Fig. 1: **Users' preferences in content types.** Distribution of per-user accesses for the content items in a user's Top-N out of 11 categories (top-left), out of 11 channels (top-right), out of 172 genres (bottom-left) and out of more than thousand serial programmes (bottom-right).

session. It is worth noting that iPlayer's video player automatically records watching position when a user interrupts a session and, if a user re-accesses the same content later, starts streaming from the recorded position. Therefore, we assume that users complete watching consecutive parts of content items during repeated accesses. We focus on regular users of iPlayer, defined as users with at least 10 sessions overall (i.e., over Wi-Fi or mobile), and at least 5 mobile sessions. This results in a subset of *3,863,031* sessions from *113,731* mobile users.

3.2 Users Preferences for Content Types

Equipped with a dataset of iPlayer accesses we firstly analyse user preferences for different types of content. All content items in iPlayer are annotated by BBC editors with one (or several) of 11 content categories (e.g., drama, comedy) and one (or several) of 172 content genres (e.g., sitcoms, crime, soaps). We also consider on which of 11 BBC channels content items were broadcast. Finally, many content items are part of multi-episode TV series, which are typically serialised into weekly broadcasts. We consider which (if any) of the more than thousand serial programmes in the content corpus an episode belongs to.

Thus, content items can be classified along four different *content type* axes: categories, genres, channels and shows. We calculate the share of per-user accesses that fall in the users' Top-N classes according to each content type axis, and measure user preferences or affinity towards particular types of items. We note that for all users in our dataset, 75% of their accesses are made for content items from only 3 out of 11 categories and for 4 out of 11 channels (Fig. 1, top row), suggesting high user affinity towards categories and channels. More remarkably, we notice a similar concentration of user preferences, even
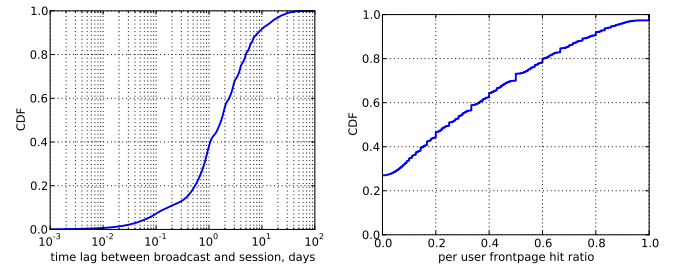

Fig. 2: **Users prefer recently released and featured shows.** Distribution of time interval between users' accesses and the broadcast time (left) and per-user shares of accesses for the content items featured on the front page (right).

when we move from 11 categories to a more fine-grained subdivision into 172 content-genres, and to extremely specific TV serials or shows (Fig. 1, bottom row).

Further, we analyse the age of the content items watched by users, by measuring the Cumulative Distribution Function (CDF) of the time elapsed between the time of an access and time when the requested content item was broadcast (Figure 2 (left)). We note that the majority of accesses are for items with ages less than 1.5 days, suggesting a preference for recent broadcasts.

3.3 UI Guidance

Next, we consider the extent to which users may be influenced by items featured at any given time on iPlayer. The iPlayer's user interface (UI) provides several means of navigating across the content corpus: via the list of featured and popular content items on the front page; via the featured lists in each content category and each of BBC channels; and via textual search over content titles. To study the effect of the UI on users' choice of content, we periodically crawled the 25 main BBC iPlayer UI elements (including the front page, channel and category pages). Complete snapshots were collected every half hour from 1-31 July 2014[5]. It is worth noting that iPlayer UI is not personalised (i.e., the front and other pages remain same across different users, browsers and device types; it also flows similarly whether the browser or a specialised mobile app is used for access). Therefore, it was sufficient to crawl the UI from a single machine, with default HTTP parameters. Although the pages are adjusted based on screen size, the relative positions of different content items on each page does not change across device types.

The front page of the BBC iPlayer user interface typically displays a list of 16-20 episodes featured by BBC editors and mixed with several groups of serial content items and a group of top-5 most popular shows. We use the timestamped snapshots of UI pages collected by repeatedly crawling BBC iPlayer and assess the probability of users accessing featured content.

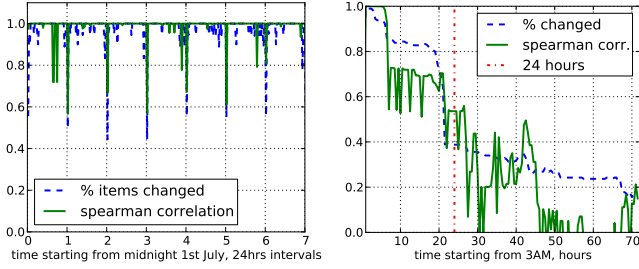5. Network failures resulted in a small loss of $< 1\%$ of collected data.

Fig. 3: **Featured pages are static on a daily basis.** Percentage of content items featured on the front page which remain unchanged from time t_i to t_{i+1} (left) and from time $t_0 = 3AM$ to time t_i (right).

In Fig. 2 (right) we plot the ratio of per-user accesses for content items which were featured on the front page at time of accesses. We note that on average, users have 25% of their accesses for content featured on the front page. However, the share may significantly vary across different users: for around 25% of users, none of their accesses are for featured content, whereas for $\approx 10\%$ of users, the vast majority (i.e., more than 80%) of their accesses are for featured content.

However, one obstacle to using such correlations between user accesses and whether or not the accessed item was featured on the UI for predictions is that the set of items featured changes regularly. Thus, predictions made based on the set of items featured at the *time of prediction* may not be accurate at the *time of access* – predictions need to be reasonably close to the time of access. To understand how far ahead the set of featured items can serve as predictive signals, we analyze how frequently featured pages are updated using our repeated crawls of featured items, performed every 30 minutes. We measure the proportion $\rho_F = \frac{F_{t+30} \cap F_t}{F_t}$ of items listed on a featured page F at time t which continue to remain on F at time $t + 30$ minutes. For the content items $F_{t+30} \cap F_t$ which remain on a featured page from time t to $t + 30$, we also measure and report Spearman's correlation of their relative position on that featured page. Fig. 3 (left) shows that featured pages remain relatively stable, with periodic changes just after midnight. To quantify this, we measure cumulative changes to the set of items featured of the front page from a fixed time $t = 3AM$ up to time $t + n$ in Figure 3 (right). We note that the vast majority of the content items remain on the featured pages for almost 24 hours, featuring a slight change in the morning hours around $10 - 11AM$, and a major change after midnight next day. This observation will justify our design decision of performing predictive preloading at 3AM in the morning (§4).

4 PREDICTING TO-BE-WATCHED CONTENT

In this section we develop a machine learning algorithm which models user preferences observed in §3 and predicts what a user is going to watch next.

4.1 Predictive Features

First, based on the observations in §3, we devise a set of features to predict whether a user U will watch an episode E. For each pair (U, E), we exploit the state of user U's profile at the time of prediction T; meta information about item E; its position in the popularity ranking and featured lists at time T; and the user U's affinity towards item E calculated from information collected for U by time T. Table 1 summarises all 449 features computed.

Our first class of features attempts to match the content type of E to the importance of that content type for U: Because a content item may simultaneously be classified in multiple top-level categories or content-genres, we use a 11-long (respectively 172-long) 0-1 vector to describe the content category (genre). We use another 11-long (172-long) vector, to describe *user affinity* to particular content categories (genres). The affinities are calculated as a share of the particular category (genre) among all items watched by the user. Similarly, we compute user affinity to particular TV series/shows and channels. However, since each item belongs to one serial and is broadcast on one channel, these affinities can be directly computed as a fraction of the user's previous history which can be attributed to a show or channel. We also compute the time since the show was broadcast as a scalar feature capturing user preference for recent content.

Next, we calculate a number of features relating to the UI that can boost or hinder its probability of being watched. To compute these features, we first rank items according to popularity, and according to the position at which a content item is featured on each of the 25 main pages of iPlayer UI at the time of preloading. In addition, to account for the exposure of the user to the item as featured on the UI, we also compute the average position of the item in each list from all previous accesses in a user's history. The user's probability of watching items from each featured page is used to personalise the weight to be placed for each page.

Our final set of features computes the propensity of the user to complete watching an unfinished program or rewatch it. As we will discuss later, some users rewatch episodes, making it important to cache even after it has been completely been watched. Similarly, if a user has completed a reasonable fraction of the show (e.g., at least 10-15 mins), it may be a good indication of intention to complete watching it, whereas if only a very small part of the show (e.g., first 2-3 mins) has been watched, it may indicate that the user did not find the item interesting and abandoned watching it. We capture such information by computing whether the item has been previously watched, what fraction of the show was watched and whether the user has re-watched other shows.

4.2 Prediction Model

For each session (U, E, T), where U denotes a user, E a content item and T the time of access, we compute the

Feature	Dimensions	Type	Description	Hypothesis	Importance
Users' Preferences in Content Types (Total importance: 0.555)					
Content category	11	E	Category to which a content item belongs.	Users prefer shows from a small number of categories, genres, channels and programmes (§3.2, Fig. 1)	0.038
Content genre	172	E	Genres to which a content item belongs		0.063
Category affinity	11	U	Share of content items a user has watched from each category.		0.042
Genre affinity	172	U	Share of content items a user has watched from each genre.		0.103
Show affinity	1	U, E	$\frac{n_{s(E)}(U)}{N_U}$, where $n_{s(E)}(U)$ is the number of episodes of show $s(E)$ that user U has watched and $N(U)$ is the total number of content items the user has watched.		0.179
Channel affinity	1	U, E	$\frac{n_{c(E)}(U)}{N(U)}$ where $n_{c(E)}(U)$ - number of episodes from channel $c(E)$ that user U has watched, $N(U)$ - total number of content items the user has watched		0.043
Content age	1	E	Time elapsed since the show was broadcast.	Users prefer recently released content items (Fig. 2 (left)).	0.087
UI Guidance (Total importance: 0.292)					
Featured content	25	E	The position of the content item on each of the featured pages at time of preloading (-1 if not featured).	Users watch content featured on the front page, channel, genre and popular pages. (§3.3)	0.061
Featured position	25	U	Average position of content items watched by a user, on each of the featured pages. The average is calculated from all previous accesses in a user's history		0.061
Content popularity rank	1	E	Position of the content item in the global popularity ranking at time of preloading.		0.071
Popularity position	1	U	Average position of content items in the popularity ranking at the time of access, calculated from all previous accesses in a user's history.		0.008
Featured probability	25	U	Empirical probability of a user watching something from each featured page.	Some users have a higher probability of accessing featured content than others. (Fig. 2.b)	0.091
Repeatedly Watched Content (Total importance: 0.154)					
Previously watched	1	U, E	Whether user U has already watched content item E before.	Users continue watching content items across networks or watch the same items again. (§5.3)	0.066
Completion ratio	1	U, E	Share of the episode length which the user has already watched before.		0.081
Probability of re-watching	1	U, E	Probability of a user watching content items more than once.	Watching content items more than once is a user-specific behaviour. (§5.3, Fig. 4 (middle)	0.007

TABLE 1: **Predictive features and their importance for mobile content preloading.** Type column indicates whether a feature has been constructed from a user's information (U), an episode's information (E) or both (U, E). Note that the importance of individual features is rounded to three decimal places here for readability. The unrounded sum of importances equals one.

set of predictive features from §4.1, which describe the history of the user and the state of the content item at time $T^{overnight}$, the night before access happens at time T [6]. For training, we only consider mobile sessions of the users which happened when users have already collected significant history, i.e., at least 5 mobile sessions. We also generate a set of negative samples for each session in the training set by randomly sampling 4 other content items available on iPlayer at T with probability proportional to the items' popularity at T (i.e., more popular content is more likely to be selected to achieve a balance in popularity of negative and positive samples). We assume that user U had a chance to watch those items but decided not to and train a model to understand what factors matter.

Our model uses a Random Forest classifier (with 500 tree-predictors), which is known for a good prediction performance with high-dimensional datasets [2]. Random Forest is an ensemble classifier which operates by constructing a multitude of decision trees using a combination of *tree-bagging* and *random subspace projections*

6. we experimented with different time points for $T^{overnight}$, and found that predictions conducted at 3AM achieve the best prediction performance.

– two different techniques which reduce the variance and the bias of individual tree-predictors, correspondingly [5]. Following standard methodology, we use Gini impurity values as the criterion for constructing decision trees and train the model over a random sample of 2K users, resulting in 85K training samples (17K positive and 67K negative samples).

4.3 Validation methodology

Firstly, we measure the performance of the algorithm in terms of its ability to classify between the preferred and not preferred content for a given user (Table 2). We note that the algorithm is able to achieve considerable classification performance (precision of 0.91 with recall of 0.80) with 10-fold cross validation over the dataset of 2K randomly selected users described above. We also note that the prediction performance does not improve significantly if we further increase the size of the dataset.

In practice, we are interested in how the algorithm is able to pick a few most likely to-be-watched episodes for a user among the variety of shows available on iPlayer and, so, we further measure the *ranking accuracy* of the algorithm as follows. We process each user session

Performance Metric	Value
Precision	0.91
Recall	0.80
Accuracy	0.94
F1-score	0.85
AUC	0.94

TABLE 2: **Performance of the prediction algorithm** as measured by mean values of accuracy, precision, recall, area under precision-recall curve (AUC) and F1-score with 10-fold cross-validation.

(E, U, T) individually and for a set of content items $E_{T^{overnight}}$ available[7] on iPlayer at time $T^{overnight}$, we use the trained model to predict the likelihood of user U watching the item, and rank them according to the predicted likelihoods. In practice, we rank more than a hundred of content items in each individual experiment. For the cases when we have very short users' histories by the time of prediction (i.e., less than 3 mobile sessions) we empirically find that users' histories are not sufficient to properly capture users preferences yet and we use popularity-based predictions instead. Overall, we conduct the testing on a total of $16.2K$ mobile sessions, from a set of $1K$ users chosen randomly such that the training and testing sets do not overlap. As discussed later in §5.1, we assume that the mobile device has space for 5–10 items. Thus, the prediction is accurate if it ranks the items watched in the top 5–10. Therefore, we measure the performance of the model by calculating the per-user accuracy of successful prediction in the top-N of the predicted list (also known as Accuracy@N).

4.4 Results

In Fig. 4 (left) we compare the performance of our personalised model against a baseline (denoted TOP acc@N) of predicting the globally most popular Top-N items with the popularity also measured at $T^{overnight}$. This shows significant improvements in accuracy with our machine learning model (denoted ML acc@N). We also note a diminishing returns for Accuracy@N with the corresponding growth of the prediction list length N: the median per-user accuracy increases considerably (by around 140%) between Top-1 (ML acc@1) and Top-5 (ML acc@5) predictions, but, only by 12% from Top-5 (ML acc@5) to Top-10 (ML acc@10) predictions. This suggests that, if successfully predicted, content items are consistently ranked high (i.e., in the Top-5) in the prediction list.

Further, we compute the extent to which individual signals contribute to the overall prediction accuracy of the proposed model. In the last column of Table 1 we report the *importance* of individual features calculated as the expected fraction of the training samples they contribute to. We note that user affinity towards shows

7. Note that the content catalogue on iPlayer changes constantly – new items are made available shortly after being broadcast on TV, and old items are regularly removed, e.g., after one week. Thus there may be content items which even a perfect predictive model cannot preload, as they may not be available as part of the content catalogue at the time of preloading.

and genres are the strongest predictors of users' future accesses – a result which is an agreement with the finding in Figure 1 that users specialize in a handful of content types.

More generally, User Preference features gained the maximum importance (i.e., 0.555) among the three different groups of features, followed by UI Guidance with total importance of 0.292. Interestingly, individual users' preferences for featured content, as measured by the featured probability, have proved to be an important feature (i.e., importance of 0.091), whereas a measure of a user's attraction towards popular content, i.e., popularity position, is an order of magnitude less important for predicting to-be-watched content (i.e., importance of 0.008). Intuitively, most users have a shared preference for popular content (and hence the content item is popular), and therefore this feature is not as discriminatory as preference for featured content, as different users prefer different featured content. This finding is also in agreement with the result in Fig. 2 (right) that featured content has strong affinity among some of the users, and anti-affinity among others. Finally, the Previously Watched features jointly account for an overall importance of 0.154 and are less important than the User Preferences and UI Guidance features.

5 PREDICTIVE PRELOADING

Equipped with the prediction algorithm from the previous section we finally introduce *predictive preloading* – a new mobile data offloading approach for catch-up TV.

5.1 Mobile Preloading

To start off, we define a naïve baseline preloading strategy based on making use of good broadband connectivity when it is available to "greedily" complete the downloading of partially watched items, and "greedily" caching them on device-local storage, therefore offloading from future sessions that may require cellular data access. This baseline approach is supported by the finding in our previous work [17, Fig. 8b] that in many cases, it may take users multiple sessions across both fixed-line and cellular ISPs to complete watching a TV show[8].

A crucial question for mobile preloading is deciding when the content should be downloaded. From the perspective of trace-based evaluation, it is not a priori clear when Wi-Fi access is available. Hence we consider two conservative assumptions: The most conservative assumption is that Wi-Fi connectivity is available only when some content is being streamed. Even this limited connectivity is sufficient to greedily prefetch content in advance of being watched, given that average broadband speed in the UK is 18.7 Mbps [27], whereas typical video bitrates, in e.g., BBC iPlayer, could be 1.5 Mbps or even lower. Thus, it is not unreasonable to expect that content can be downloaded at up to 10 times the playback speed.

8. This may happen for various reasons. Users may, for instance, stop watching when their train reaches the destination, and continue at home.
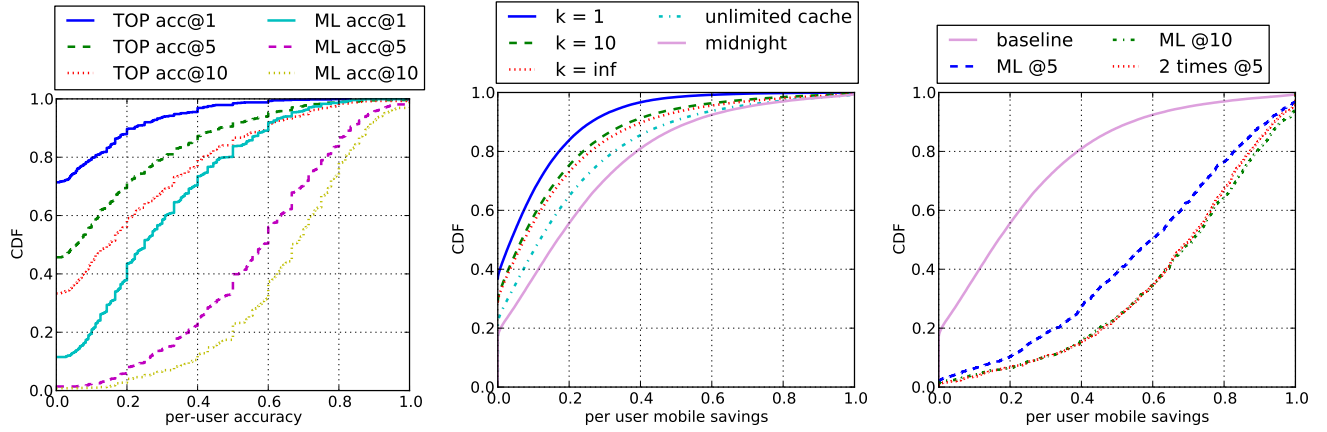
Fig. 4: **Performance of the Preloading Algorithms. Left:** Per-user accuracy@N of ML predictions (ML acc@N) for different lengths N of prediction list benchmarked over popularity-based predictions (TOP acc@N). **Middle:** Cumulative Distribution Function (CDF) of per-user mobile savings (θ) with greedy preloading of currently (and previously) watched content items. Simulations conducted for different settings of spare bandwidth factor k, with a single content item in cache (top solid, dash, dot), with unlimited cache (dash-dot), and after adding a scheduled preloading session at midnight (bottom solid). **Right:** Mobile savings achieved from predictive preloading with different storage sizes of 5 items (ML@5) or 10 items (ML@10); and with a more frequently scheduled preloading of twice-daily schedule and with a storage size of 5 items (2 times @5), all benchmarked against the best greedy preloading method (baseline), which is also shown as 'midnight' in the middle plot.

A less conservative, but still reasonable, assumption is that Wi-Fi connectivity and spare bandwidth is available at some scheduled time during the day. For instance, network usage decreases drastically after midnight, and a user may be expected to be at home, where her device can obtain Wi-Fi connectivity with ample spare bandwidth. Note that although the evaluation below assumes a specific time point for clarity, we only require that spare Wi-Fi bandwidth is available at some time daily.

We expect that more than one content item can be preloaded during one or multiple Wi-Fi sessions. However, in reality, the savings may be limited by the storage available on the phone. As a back-of-envelope calculation, a 60 minute TV show encoded at 1.5 Mbps could take up ≈ 675 MB of storage. An older 8GB iPhone may have ≈ 4.9GB available for *all* user data[9]. Thus, it is reasonable to assume storage for preloading about 5 shows. More recent phones or tablets could have 16GB storage for example, thus storing ≈ 10 shows is still reasonable.

Predictive preloading expands beyond partially watched content items and allows to load unwatched content that a model predicts are likely to be accessed. In the following, we explore to what extent predictive preloading based on the machine learning algorithm from §4 can outperform simple greedy techniques.

5.2 Simulation Settings

We develop an event-based simulator where for a given Wi-Fi session $S_{Wi-Fi} = (U, E, D)$ of a user U watching

content E for D seconds, we can preload and cache on the user's device content corresponding to kD seconds, where $k = \frac{\alpha}{\beta}$ is a *spare bandwidth factor*, defined as the proportion between a user's download bandwidth α and content bitrate β. The higher k is, the more the spare bandwidth available.

Thus, at the end of Session S_{Wi-Fi}, the user's device has preloaded content equal to $D_{pre} = min(P(U, E) + kD, L(E))$ seconds of playback, where $L(E)$ denotes the length of the episode, and $P(U, E)$ denotes the part of the content item preloaded during previous accesses of U for content E. The $min(\cdot)$ here captures the fact that it is impossible to preload more than the length of the episode $L(E)$ even if the spare bandwidth factor and the duration of a session D are large. It is also worth noting that $k = 1$ corresponds to the case when only as much content is cached on a user's device as the user has actually watched.

Caching: To derive any benefit from preloading, the preloaded content needs to be cached. We considered two variants. The first, basic, assumption is that exactly one item is cached. Thus, preloading a new content item would *replace* the previously preloaded item. We follow this up by considering an infinite cache that can save all previously preloaded items. Intermediate cache sizes are not reported, because we find (see Fig. 4 (middle)) that increasing storage does not yield much improvement.

Calculating savings: For each mobile session $S_{Mobile} = (U, E, D)$ of user U to episode E which lasts for a duration of $D = D_{mobile}(S)$ seconds, we check whether a user

9. http://goo.gl/5dxDKi

has already preloaded the part of the content item being accessed, and if yes, measure how many seconds of the user's mobile traffic would have been saved by watching D from the user's cache rather than streaming over a cellular connection. Note that if the part being watched has not been preloaded fully beforehand, we assume that only the preloaded part (say, D_{pre} seconds) would be watched from the user's cache, whereas the rest of the session ($D_{mobile} - D_{pre}$ seconds) would be streamed from a mobile network. We measure performance of the proposed preloading mechanism in terms of the *per-user mobile savings* which we formally define as follows[10]:

$$\theta(U) = \frac{\sum_{S \in S_{Mobile}(U)} D_{pre}(S)}{\sum_{S \in S_{Mobile}(U)} D_{mobile}(S)} \quad (1)$$

5.3 Naive Baseline: Greedy Preloading

Active preloading is not required for savings: We start off with a scenario when *no* preloading happens, i.e., only the content that has been already played back to the user is cached (i.e., $k = 1$). We also set the cache size to one item, i.e., only the last item played back is cached. Surprisingly, even this *passive caching* of the last item streamed, without downloading in advance, can achieve non-trivial savings (Fig. 4 (middle)) of $\approx 11\%$ on average, albeit for a fraction of the user population ($\approx 63\%$). The savings arise because those users also watch the same content more than once. However, as many users do not rewatch shows, the average savings for the whole population is only $\approx 7\%$.

Increased bandwidth and storage help, but benefits are limited: Next, we consider scenarios where $k > 1$, to study the benefits of increased bandwidth. Fig. 4 (middle) shows that there is additional savings to be had when k increases to 10, but almost no additional savings beyond this can be achieved, even when k becomes infinite and content can be preloaded instantaneously, as soon as a Wi-Fi session starts. To understand why this is the case, notice that with a spare bandwidth factor of $k = 10$ (which we recall is reasonable assumption given the average download bandwidth of 18.7Mbps in the UK and a typical iPlayer streaming bitrate of 1.5Mbps), a typical 60 minute-long TV show can be preloaded in just 6 minutes. Thus, no additional benefit is to be had by increasing k when the Wi-Fi session lasts longer than 6 minutes. We then study the importance of cache size by simulating an unlimited cache, in addition to $k = \infty$. This increases the mobile savings for 70% of the users and saves $\approx 13\%$ on average, suggesting that some users

10. Note that we measure savings in terms of mobile minutes rather than directly in terms of bytes saved because bitrate information was not available for many mobile sessions in our dataset. However, given that bitrate variation in iPlayer sessions is typically quite small [16], we envisage that savings in terms of minutes translate approximately to bytes.

are likely to switch back and continue watching (or re-watching) a content item even after they have already started watching something new. On the other hand, we note that even with all these proposed unrealistic adjustments, the median per-user savings of this *just-in-advance preloading* only reaches $\approx 9\%$ when averaged over the whole population.

Scheduled preloading helps by catching mobile-only split sessions: With an infinitely fast preloader ($k = \infty$) and infinite storage, all sessions over cellular network that are preceded by a Wi-Fi session to the same content item should be offloaded. Thus, any remaining split sessions are due to mobile-only access. For instance, a user may begin watching the first part of a TV show over cellular network during her morning commute and finish also over cellular network during her evening commute, or on the next day's commute. To catch such sessions, we need a separate special session that accesses iPlayer solely to preload content rather than piggybacking the preloading on top of an existing Wi-Fi session. To study the effect of such a dedicated preloading session, we explore a *scheduled preloading* approach and model a batch job on each user's device which wakes up each midnight and preloads all partially watched content items from the users' previous sessions. Fig. 4 (middle) shows that with this strategy, up to 83% of users can save and each obtains an average savings of $\approx 22\%$.

5.4 Predictive Preloading

Next, we ask how the machine learning model introduced in §4 can improve mobile savings.

Predictive approach improves over greedy scheduled preloading: Fig. 4 (right) shows that predictive preloading achieves significantly higher savings in comparison to the greedy scheduled overnight preloading: greedy scheduled preloading (shown as baseline in Fig. 4 (right)) assumes infinite cache size and still performs worse than ML@5 and ML@10 with cache sizes of 5 and 10 items. This suggests that machine learning based selection of to-be-watched content items can capture a significantly wider range of indicators for future preferences of the users than just the history of previously watched episodes, as used in the greedy method. This result is in agreement with the fact that information about whether (and for how long) a content item has been watched contributes only 0.154 (out of 1) to overall importance of prediction features in Table 1.

Increased storage offers diminishing returns: Fig. 4 (right) shows that doubling storage from 5 to 10 items yields only a slight (18%) increase in mobile savings. This is in agreement with the accuracy results discussed in §4 which suggests a high concentration of successful predictions in the Top-5 of the prediction list (Figure Fig. 4 (left)). Thus, storing a larger list of items does not translate directly to mobile savings.

Increased frequency of preloading helps a bit: Finally, we analyse potential gains from increasing the frequency of preloading opportunities throughout a day. We motivate this step with the illustrative example of a commuter who may access iPlayer during the morning commute once and again during the evening commute. Recomputing predicted items after the morning commute can help because the model can learn from the new addition to the user's history, and also because a more updated list of featured items, as well as an updated list of most popular items can be used. Examining our dataset, we find that a majority ($> 60\%$) of the users, if they access iPlayer from a mobile device during a day, do so two or more times during that day. Indeed, we observe a 18% improvement in mobile savings by incorporating this additional opportunity for predictive preloading during the day (dash dot line in Figure 4 (right)). However, this approach yields diminishing returns and we do not observe significant improvements from further increasing the number of preloading opportunities during the day. Interestingly, we notice that increasing storage size and adding preloading opportunities yield similar gains in mobile savings, a fact that can be exploited to trade-off between the two strategies depending on whether storage capacity or Wi-Fi access availability is a constraint.

6 DISCUSSION AND CONCLUSIONS

In this paper we proposed a novel approach for data offloading that operates by predicting which items users are likely to watch while on the move, and preloads such items from a reliable Wi-Fi connection beforehand. We evaluated this technique using as case study a large "catch-up" TV system used widely in the UK for streaming BBC shows. However, similar techniques could potentially be applied in other situations as well, with the prediction models adapted to the use case at hand. We believe that such predictive preloading techniques could be a valuable addition to the array of offloading techniques proposed for future cellular systems [32]. Below we conclude with a summary of the main results in the context of our specific study, and discuss caveats and limitations:

Using a trace of nearly 3.9 million sessions to iPlayer from mobile devices we first looked for signals which can predict future user access. We found that individual users' preferences in content items are typically concentrated around a handful of content genres, channels and shows. We also found that in general users are influenced by the UI guidance, such as content featured on different iPlayer pages but the extent of this influence varies significantly across different users: For around 25% of users, none of their accesses are for featured content, whereas for 10% of users the vast majority (i.e., more than 80%) of their accesses correlate with the content featured in the iPlayer UI.

We exploited these insights to develop a supervised learning algorithm which can predict more than 66% of per-user accesses in the Top-10 of the prediction list for the majority of users and over 88% for the top 10% of users. Based on the machine learning model, we proposed a predictive preloading mechanism to offload mobile traffic of individual users during work commutes and compared it with a number of naïve approaches. Our result suggests that predictive preloading significantly outperforms all naïve strategies and allows up to 71% of mobile traffic savings for an average user and over 95% for the top 10% of users.

Although we evaluated using a large and real dataset, full-scale deployments may need to consider additional factors. For instance, our evaluation relied on a month-long trace, whereas models used in a real deployment may need to be tuned to perform well over larger time periods. On the one hand, a larger time period can benefit from a longer user history. On the other hand, user interests may change over time, and the content catalogue on BBC iPlayer itself changes regularly (new shows are typically added in the first hours after broadcast and are removed after 30 days). Therefore, the machine learning model will need to adapt and use an appropriate amount of recent user history to maintain good performance. Similarly, our approach relies on capturing a wide range of signals, some of which may be specific to catch-up TV systems such as BBC iPlayer. Additional tuning may be required if it needs to be applied to other platforms such as Netflix which may have slightly different content availability characteristics (e.g., larger content catalogue, or a catalogue of items which does not change as often as iPlayer).

Further, storage strategies in full-scale deployments can incorporate simple improvements that can potentially improve the performance beyond what we report here. One approach would be to optimise the length of the preloaded episode based on a typical duration of a user's commute: Instead of preloading the full length of a long episode, the algorithm may preload small fractions of multiple episodes for short commutes, thereby increasing the chance that the content watched by the user will be in the preloaded list. Another approach relies on using the storage of the device more effectively by taking the bitrate preferences of the user or device capabilities into account: For example, a smartphone with small screen resolution and, consequently, a lower bitrate requirement, can accommodate a larger number of episodes given the same storage constraints.

In summary, we envisage that with the right predictive model, preloading can be a very fruitful strategy for many kinds of applications. However, a number of use-case specific adaptations will need to be made, as in our case study. Full-scale deployments will need to bear other considerations in mind, as discussed above.

REFERENCES

[1] M. Z. Ahmad, J. Qadir, N. Rehman, A. Baig, and H. Majeed. Prediction-based channel zapping latency reduction techniques for iptv systems - a survey. In *Proceedings of ICET*, pages 466–470. IEEE, 2009.

[2] D. R. Amancio, C. H. Comin, D. Casanova, G. Travieso, O. M. Bruno, F. A. Rodrigues, and L. da Fontoura Costa. A systematic comparison of supervised classifiers. *PloS one*, 9(4):e94137, 2014.

[3] A. Balasubramanian, R. Mahajan, and A. Venkataramani. Augmenting mobile 3g using wifi. In *Proceedings of MobiSys*, pages 209–222. ACM, 2010.

[4] BBC. Bbc iplayer performance pack. http://goo.gl/DZhHU3, Oct 2014.

[5] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[6] Cisco. Cisco visual networking index: Global mobile data traffic forecast update, 2013-2018. Available from http://goo.gl/c0DRr5, February 2014.

[7] comScore. UK digital future in focus, Feb 2013.

[8] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang. Understanding the impact of video quality on user engagement. *ACM SIGCOMM CCR*, 41(4):362–373, 2011.

[9] A. Finamore, M. Mellia, Z. Gilani, K. Papagiannaki, V. Erramilli, and Y. Grunenberger. Is there a case for mobile phone content pre-staging? In *Proceedings of Co-NEXT*, pages 321–326. ACM, 2013.

[10] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire. Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution. *Communications Magazine, IEEE*, 51(4):142–149, 2013.

[11] B. Han, P. Hui, V. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan. Mobile data offloading through opportunistic communications and social participation. *Mobile Computing, IEEE Transactions on*, 11(5):821–834, 2012.

[12] M. A. Hoque, M. Siekkinen, and J. K. Nurminen. Using crowd-sourced viewing statistics to save energy in wireless video streaming. In *Proceedings of the 19th MobiCom*, pages 377–388. ACM, 2013.

[13] ISPReview. Shock as study finds naff mobile broadband coverage on UK trains. Available from http://goo.gl/7IJ19V, October 2014.

[14] K. Jang, M. Han, S. Cho, H.-K. Ryu, J. Lee, Y. Lee, and S. B. Moon. 3g and 3.5 g wireless network performance measured from moving cars and high-speed trains. In *Proceedings of MICNET*, pages 19–24. ACM, 2009.

[15] P. Kamaraju, P. Lungaro, and Z. Segall. A novel paradigm for context-aware content pre-fetching in mobile networks. In *WCNC, 2013 IEEE*, pages 4534–4539. IEEE, 2013.

[16] D. Karamshuk, N. Sastry, A. Secker, and J. Chandaria. Isp-friendly peer-assisted on-demand streaming of long duration content in bbc iplayer. In *Proceedings of INFOCOM*. IEEE, 2015.

[17] D. Karamshuk, N. Sastry, A. Secker, and J. Chandaria. On factors affecting the usage and adoption of a nation-wide tv streaming service. In *Proceedings of INFOCOM*. IEEE, 2015.

[18] Y. Kim, J. K. Park, H. J. Choi, S. Lee, H. Park, J. Kim, Z. Lee, and K. Ko. Reducing iptv channel zapping time based on viewer's surfing behavior and preference. In *Broadband Multimedia Systems and Broadcasting, 2008 IEEE International Symposium on*, pages 1–6. IEEE, 2008.

[19] KnowYourMobile. All you can eat data: Which UK network offers the most. Available from http://goo.gl/noj2GG, June 2014.

[20] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong. Mobile data offloading: how much can wifi deliver? In *Proceedings of Co-NEXT*, page 26. ACM, 2010.

[21] S. Lee, D. Lee, and S. Lee. Personalized dtv program recommendation system under a cloud computing environment. *Consumer Electronics, IEEE Transactions on*, 56(2):1034–1042, 2010.

[22] S. Lee, H. Moon, H. Bahn, T. Kim, and I.-S. Kim. Popularity and adjacency based prefetching for efficient iptv channel navigation. *Consumer Electronics, IEEE Transactions on*, 57(3):1135–1140, 2011.

[23] Z. Lu and G. de Veciana. Optimizing stored video delivery for mobile networks: The value of knowing the future. In *INFOCOM, 2013 Proceedings IEEE*, pages 2706–2714. IEEE, 2013.

[24] G. Nencioni, N. Sastry, J. Chandaria, and J. Crowcroft. Understanding and decreasing the network footprint of catch-up tv. In *Proceedings of WWW*, pages 965–976. ACM, 2013.

[25] G. Nencioni, N. Sastry, G. Tyson, V. Badrinarayanan, D. Karamshuk, J. Chandaria, and J. Crowcroft. Score: Exploiting global broadcasts to create offline personal channels for on-demand access. *Transactions on Networking*, 2015.

[26] Ofcom. 4g and 3g mobile broadband speeds research. Available from http://goo.gl/cTIHX1, November 2014.

[27] Ofcom. UK fixed-line broadband performance, may 2014. Available from http://goo.gl/5xq5Q6, Oct 2014.

[28] V. Singh, J. Ott, and I. D. Curcio. Predictive buffering for streaming video in 3g networks. In *Proceedings of WoWMoM, 2012*, pages 1–10. IEEE, 2012.

[29] V. A. Siris and M. Anagnostopoulou. Performance and energy efficiency of mobile data offloading with mobility prediction and prefetching. In *Proceedings of WoWMoM, 2013*, pages 1–6. IEEE, 2013.

[30] V. A. Siris and D. Kalyvas. Enhancing mobile data offloading with mobility prediction and prefetching. *ACM SIGMOBILE Mobile Computing and Communications Review*, 17(1):22–29, 2013.

[31] UITP. Internet connectivity in underground rail systems. Available from http://goo.gl/y0im3T, May 2014.

[32] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. Leung. Cache in the air: exploiting content caching and delivery techniques for 5g systems. *Communications Magazine, IEEE*, 52(2):131–139, 2014.

[33] X. Wang, T. T. Kwon, Y. Choi, H. Wang, and J. Liu. Cloud-assisted adaptive video streaming and social-aware video prefetching for mobile users. *Wireless Communications, IEEE*, 20(3), 2013.

[34] M. Xu, S. Berkovsky, S. Ardon, S. Triukose, A. Mahanti, and I. Koprinska. Catch-up tv recommendations: show old favourites and find new ones. In *Proceedings of the 7th ACM RecSys*, pages 285–294. ACM, 2013.
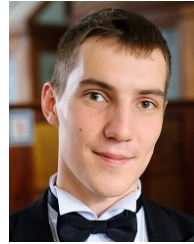
Dmytro Karamshuk Dmytro Karamshuk is a postdoctoral researcher at King's College London. His research interest is in machine learning approaches to modelling behavior of online users. He has previously worked on studying watching patterns of TV-streaming users (e.g., BBC iPlayer), mining geo-location social networks (e.g., Foursquare) and predicting user behaviour in image-based social platforms (e.g., Pinterest). He is an active contributor to the computer networks (Infocom, Com-Mag, etc.) and data mining communities (KDD, WWW, etc.).

Nishanth Sastry Nishanth Sastry is a Senior Lecturer at King's College London. He holds a PhD from the University of Cambridge, UK, a Master's degree from The University of Texas at Austin, and a Bachelor's degree from Bangalore University, India, all in Computer Science. He has over six years of experience in the Industry (Cisco Systems, India and IBM Software Group, USA) and Industrial Research Labs (IBM TJ Watson Research Center). His work has ranged over several layers of the network stack and he is currently involved in building better networked systems using data-driven approaches.
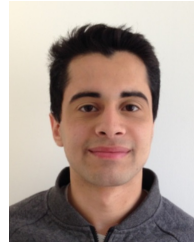
Mustafa al-Bassam Mustafa Al-Bassam is an undergraduate in Computer Science at King's College London, and was a research fellow in the Informatics department at King's College London working on optimizing content delivery through the analysis of users' access patterns.

Andrew Secker Andrew Secker is a Senior Research Engineer for BBC R&D. After gaining a Ph.D. from the University of Kent, Andrew worked as a Research Associate applying machine learning algorithms to protein data sets. He currently specialises in the use of the BBC's programme metadata on streaming, broadcast and hybrid platforms. Andrew represents the BBC technical interests across a number of industry bodies including the Digital Television Group and Digital UK.

Jigna Chandaria Jigna Chandaria is a Lead Research Engineer at BBC R&D. Her research focuses on the environmental impact of broadcasting and media technology. She is interested in how we assess environmental impact, sustainable design and using sustainability to drive innovation. She chairs the EBU Strategic Programme on Sustainable Technology in Broadcasting.

Prior to starting up this research area, she developed new technology for TV production, in particular real-time 3d graphics for sports and virtual studios. Her areas of research included real time image processing, camera tracking, augmented reality and 3d modelling. She worked closely with BBC production teams; worked on several UK and European collaborative projects and contributed to the Piero sports graphic system which was awarded The Queen's Award for Enterprise.

She has a MEng in Electrical and Information Sciences from the University of Cambridge and an MSc in Innovation and Design for Sustainability at Cranfield University.